INFORMATICS INSTITUTE OF TECHNOLOGY

INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

6COSC019C: Cyber Security

Coursework

Name: Muhammad Areeb Niyas

20200129 | w1809939

Module Leader: Mr. Saman Hettiarachchi

Submitted in partial fulfilment of the requirements for the BSc(Hons) in Computer Science degree at the University of Westminster.

Date: May 2024

Table of Contents

List of Tables

List of Figures

SCENARIO

Recently, my company was hired to carry out a penetration test for a **medical clinic** containing **60 employees (medium sized)** with many branches across Sri Lanka. The clinic offers a variety of online services including checking online lab reports, scheduling appointments and checking availability of doctors.

The clinic **web application** allows patients to view their reports online and schedule appointments, so it handles **sensitive information like patient data, history and transactions**. Also, the staff attendance monitoring and employee data is also contained in the application due to payroll and HR management. It is critical that the clinic complies with mandatory healthcare rules and regulations like HIPAA due to the personal and sensitive information it contains.

The clinic contains **2 types of users: staff and patients**. Staff members use the clinic's application to monitor appointments and publish patient/lab reports. Patients can access the web application to schedule appointments, view reports and check the availability of doctors.

Machine and Corresponding IPs

Kali Linux (Attacker)	OWASP (Vulnerable Server/ Victim)	Windows (Victim)
192.168.56.100	192.168.56.101	192.168.56.102

Table 1: Virtual machine IPs

PART A – INFORMATION GATHERING

A.1 OSINT Activities

A.1.1 OSINT Activity 1 – WHOIS

The author utilized WHOIS which is a protocol for querying and retrieving information about domain names and IP addresses. Below illustrations shows the command used and the results obtained by using this utility.

Command Used – whois cwscenario.site

```
┌──(w1809939㉿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 o
utput for information on how to contact the Registrant, Admin, or Tech contact of the queried d
omain name.
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 outp
ut for information on how to contact the Registrant, Admin, or Tech contact of the queried doma
in 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: 2024-04-30T15:53:03.0Z <<<
```

Figure 1: WHOIS results

The results obtained from WHOIS for the cwscenario.site reveals private information such as domain status, registration expiry and other name servers. This information can be used to identify

potential vulnerabilities in the infrastructure and plan attack strategies. Some of the sensitive information gathered are listed below:

- Registrar URL (https://ionos.com)
- Creation date – 2021-01-07, Registry expiry date – 2025-01-07
- Domain status
- Registrant Country – GB
- Registrant Province – GLS
- Name servers
- Registrar emails, phone numbers

They can provide valuable insights to attackers which can be used to plan and execute various types of malicious activities such as: phishing, social engineering, DNS manipulation and cybercrimes.

A.1.2 OSINT Activity 2 – TheHarvester

The author utilized TheHarvester which is a tool for querying and retrieving passive information mainly about domain names, hosts and IPs. Below illustration shows the command used and the results obtained by using this utility.

Command Used - theHarvester -d cwscenario.site -b all

```
[*] 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

┌──(w1809939㉿kali)-[~]
```

Figure 2: TheHarvester results

The results obtained from TheHarvester for the cwscenario.site reveals private information such as ASNS, URLs, IPs and hosts. This information can be used for various harmful purposes. Some of the sensitive information gathered are listed below:

- ASNS

- 3 IP addresses
- 12 Hosts information

The information gathered like the ASNS can be used to identify the internet service provider and the IP addresses obtained could be specifically targeted for cybercrimes and host information could be used to identify entry points for attacks. Attackers can use this to plan and execute various types of malicious activities such as: network attacks, data breaches, and service disruptions.

A.1.3 OSINT Activity 3 – dnsenum

Command Used - dnsenum --enum cwscenario.site

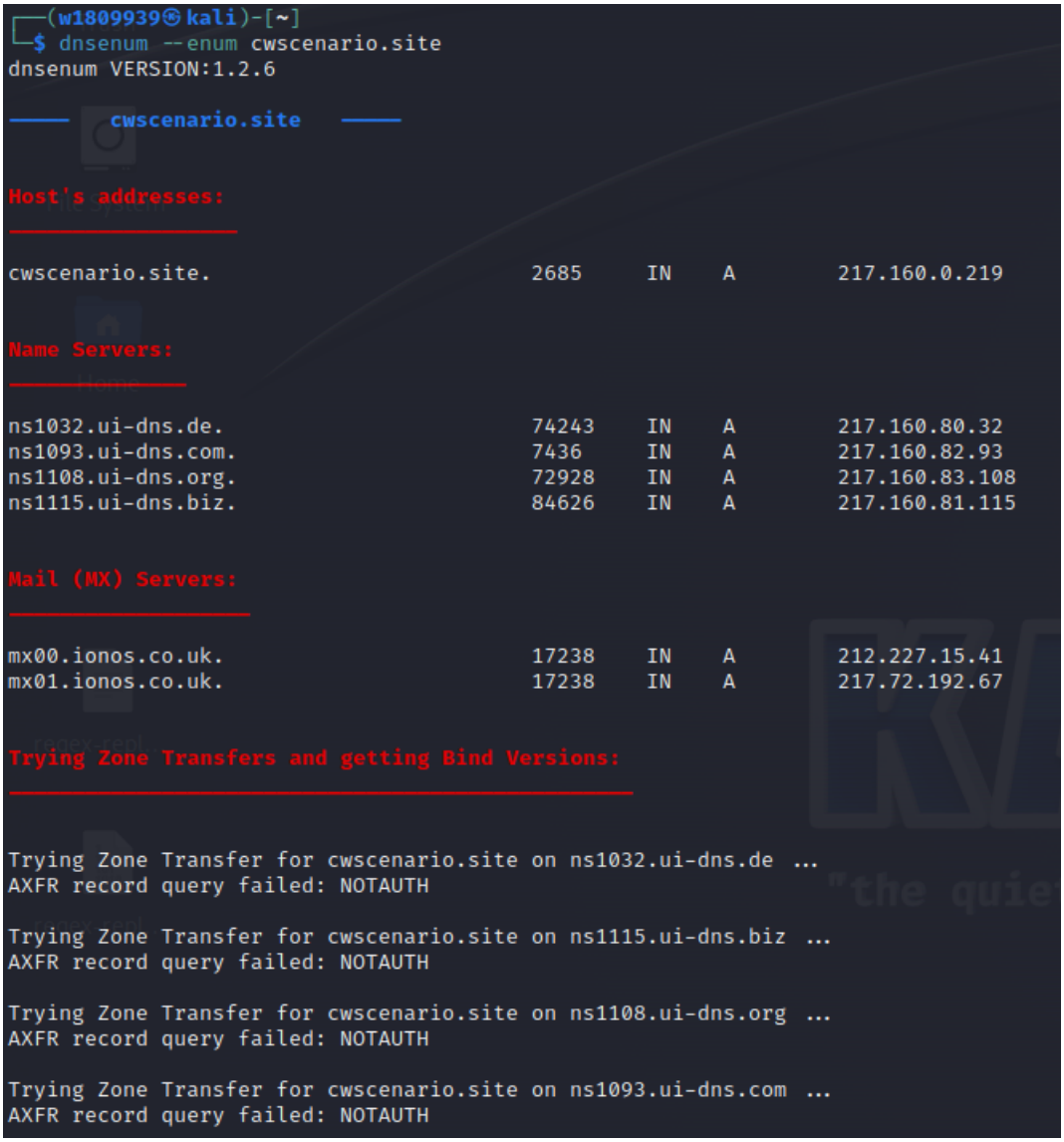

Figure 3: The results for dnsenum (1/2)

Figure 4: The results for dnsenum (2/2)

The results obtained from dnsenum for the cwscenario.site reveals private information such as Host addresses, Name servers and IPs. This information can be used for various harmful purposes. Some of the sensitive information gathered are listed below:

- 4 name servers
- 2 mail servers
- Host information

- IPs

The information gathered via DNS Enumeration revealed sensitive information such as IPs, name servers, mail servers and host information. However, performing reverse lookup was denied which suggests the implementation of a security measure which denied a zone transfer.

A.1.4 How OSINT Can be Effective and Why it is the First Activity Carried Out

OSINT stands for Open-Source Intelligence which is used to collect information from public sources. It involves acquiring data and gathering information from sources available publicly like websites, social media, public databases and etc. The data gathered can be used for multiple purposes such as market research, data analysis and investigating competition. On the other hand, it is a critical tool used for cybercrimes to plan and execute targeted attacks on organizations (Titterington, 2023).

Some of the ways in which OSINT activities can be **effective** are listed below:

1. Target Identification
2. Attack Surface Enumeration
3. Vulnerability Assessment
4. Phishing

It is often the **first step carried out by penetration testers** because by the effectiveness mentioned above, it can be used to understand the target environment and gather information about the infrastructure of an organization. Also, helps identify potential attack entry points during the assessment such as exposed systems and network configurations. So these activities help penetration testers identify weaknesses and identify breaches in industry regulations and mitigate risks which will contribute to a comprehensive testing process (Pritchard, 2020).

A.1.5 Scenario Assessment

Information Obtained	How Dangerous it can be
Registrar URL	Knowing the registrar URL, an attacker could impersonate communications from the

	registrar and deceive clinic staff into providing sensitive information.
Creation date and registry expiry date	This could be used to plan attacks in the long-term by strategically planning based on the duration of the domain of the clinic
Domain status	Attackers could exploit the domain status by impersonating and hijacking the domain with the exposed domain information.
Registrant country and province	Attackers could use this for social engineering attacks and phishing purposes by exploiting regional vulnerabilities near the clinic area and location.
Registrar emails and phone numbers	An attacker could impersonate as an employee from the medical clinic and try to gain unauthorized access.
Name servers	Knowledge of the name servers could lead to targeting DNS infrastructure for hijacking or DDoS attacks which could disrupt clinic operations.
IPs and Host information	This exposure could be used to find potential entry points for attack for the clinic which could lead to the loss of personal patient information.

Table 2: Scenario assessment (OSINT)

A.2 Reconnaissance

A.2.1 Reconnaissance Activity 1 - DirBuster

The author utilized DirBuster to extract information by brute force about searching for a list of files and directory in a web server by giving a text file which contains the target list of words. Below illustration shows the commands used and the results obtained by using this utility.

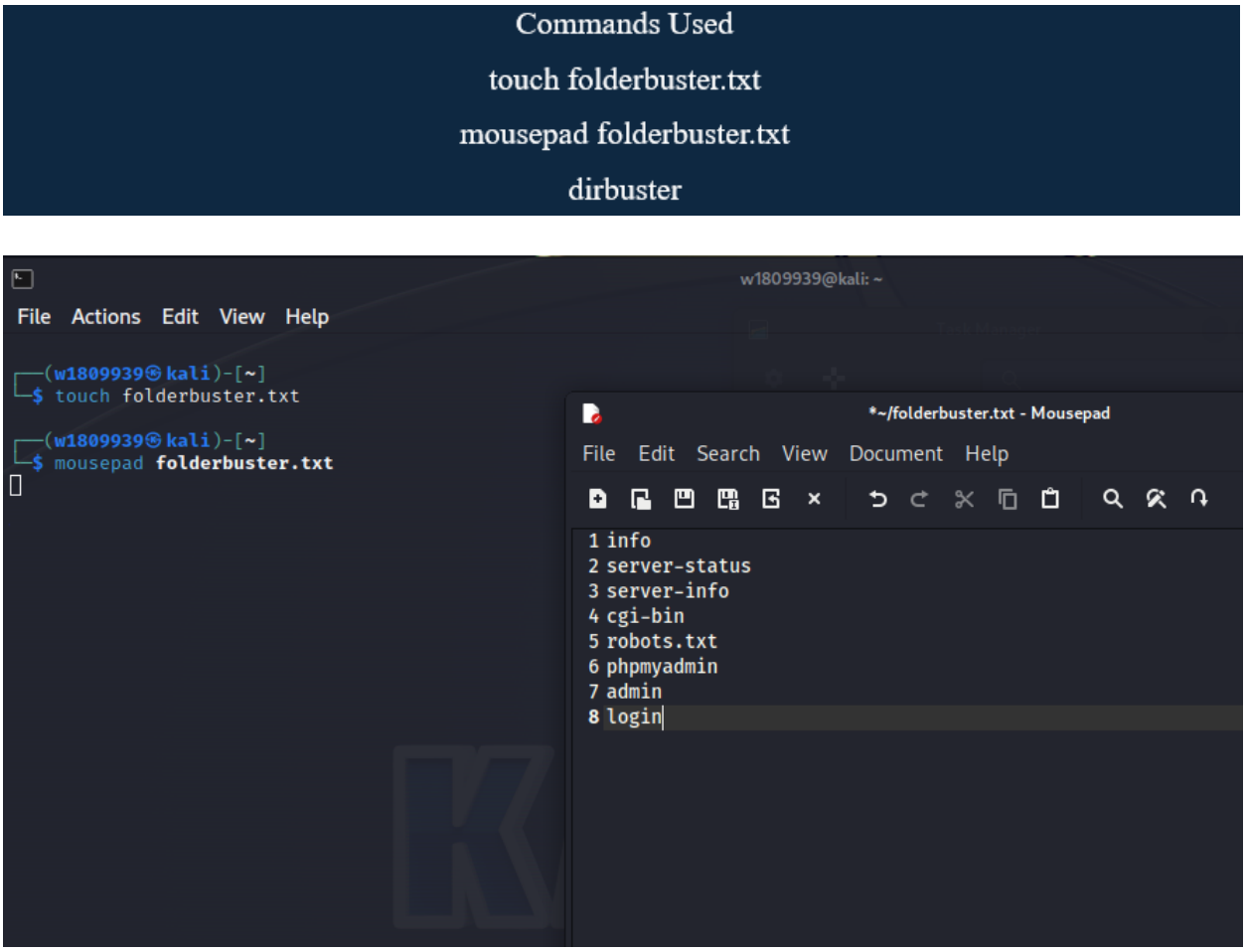

Figure 5: DirBuster setup (1/2)

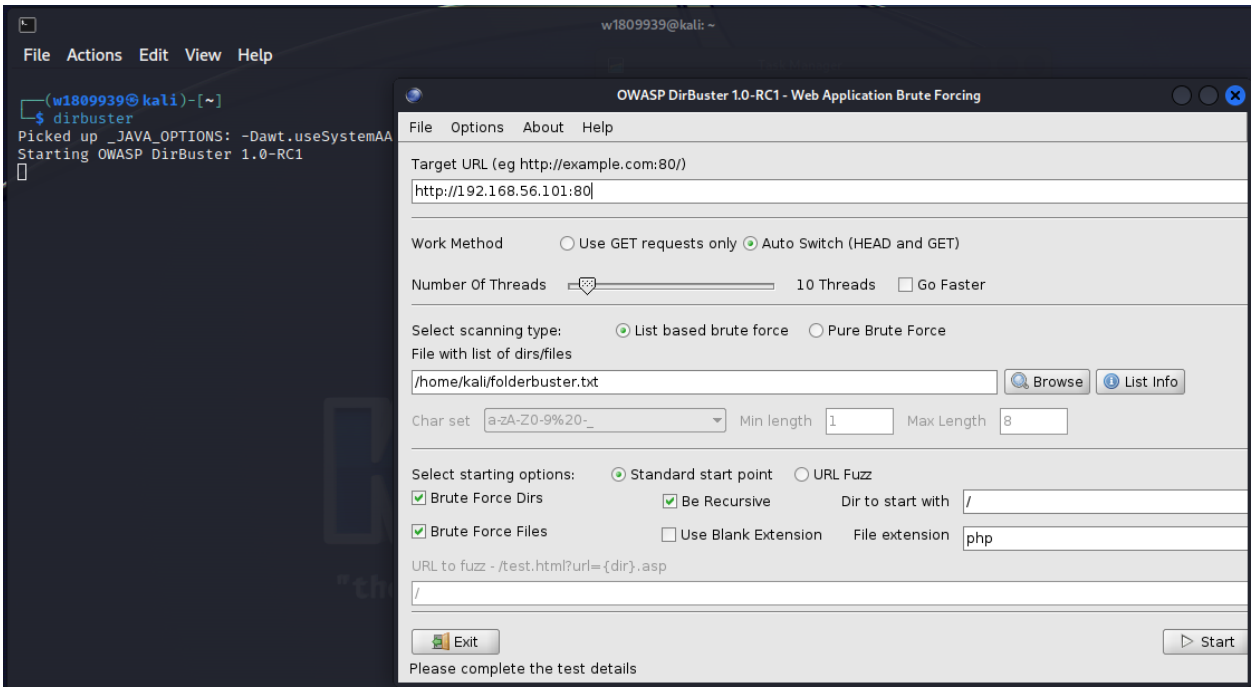

Figure 6: DirBuster setup (2/2)

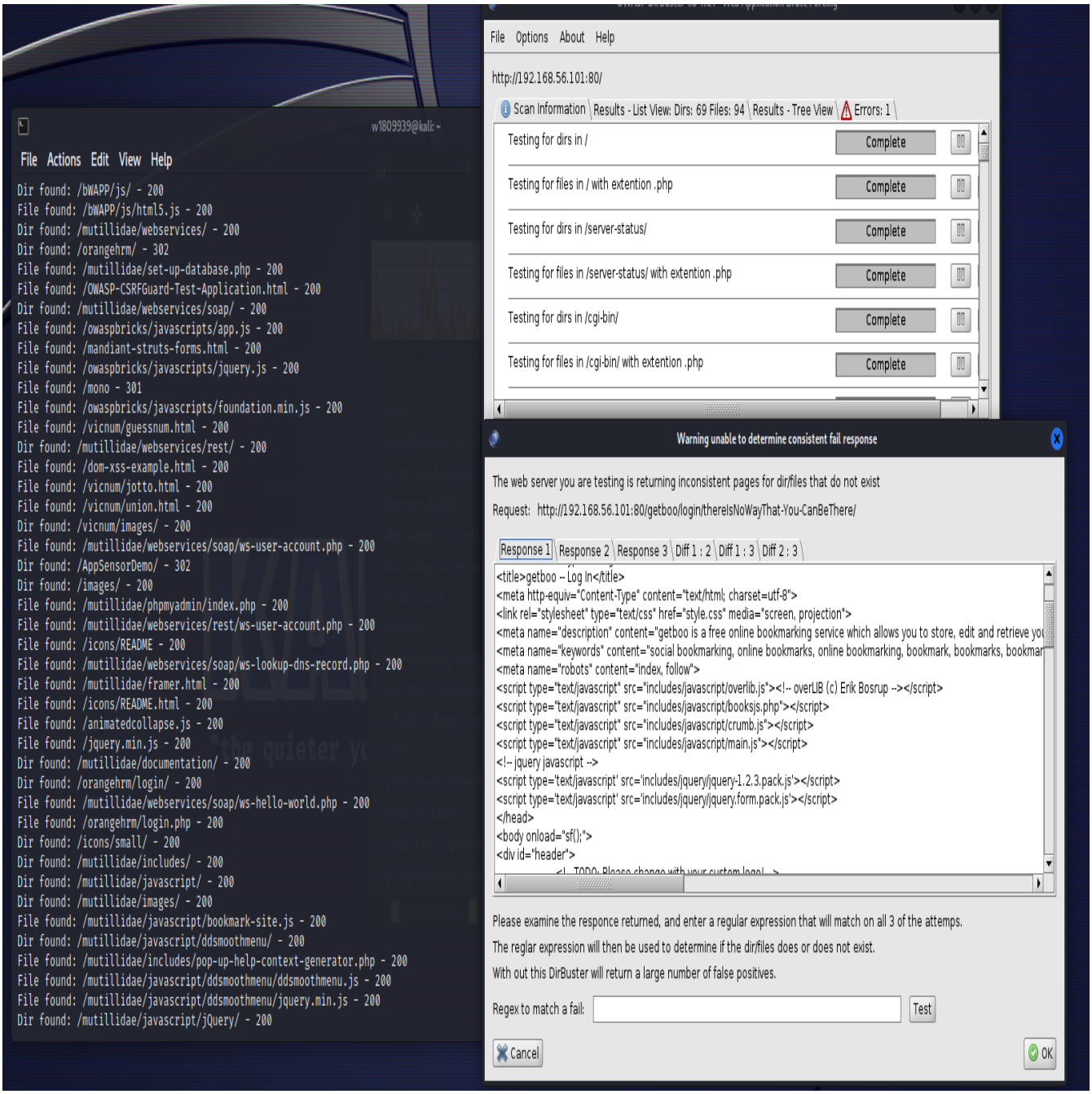

Figure 7: DirBuster results

OWASP bricks, jquery files, images and multiple JavaScript files were found using brute force on the server URL. Various directories were identified as well discovering hidden and different directories and file names. By identifying vulnerable and unprotected directories, attackers can use this to identify weak points allowing access to directories without authorization. Also, discovering hidden files could be used to gather intelligence and target attacks while stealing sensitive data.

Then, configurations could be exploited from the misconfigured directories using the information gathered.

A.2.2 Reconnaissance Activity 2 - Active Reconnaissance

Sometimes by simply looking at the code of the page source, some information could be exposed. The author utilized active reconnaissance by 'View page source'. Below illustration shows the results obtained by using this utility

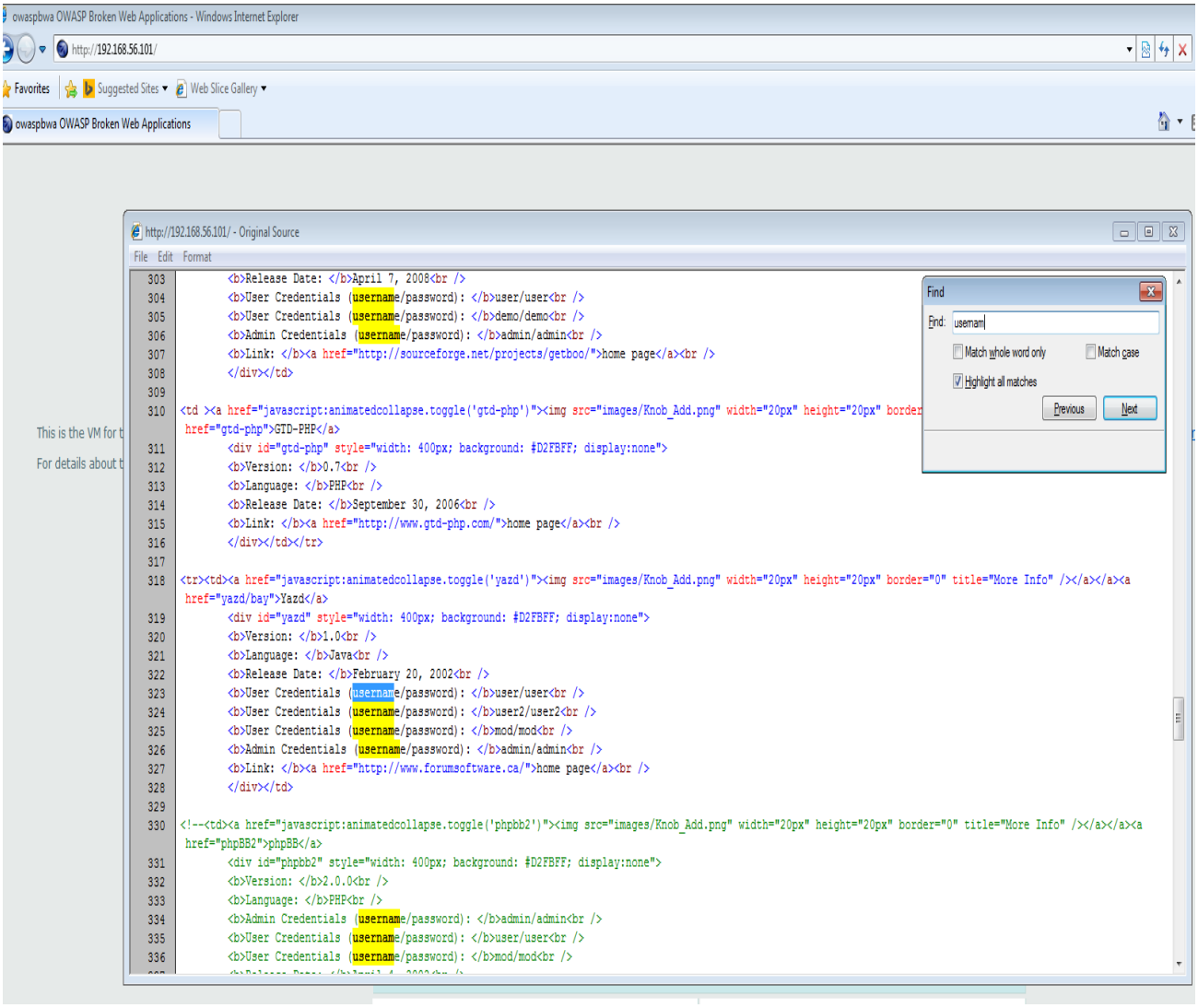

Figure 8: View page source code

Here you can see two crucial information visible by just inspecting the page source:

- Usernames: user, demo, admin

- Corresponding passwords: user, demo admin

This information could be easily used by attackers to gain unauthorized access to any system and word lists and workaround can be used to update these usernames and passwords to whichever credential the attacker wants.

A.2.3 Reconnaissance Activity 3 - robots.txt

The author utilized vicnum which contains web apps based on games. The author utilized this method by trying to access to robots.txt file of this site to find files and directory that may not be directly linked to the main application.

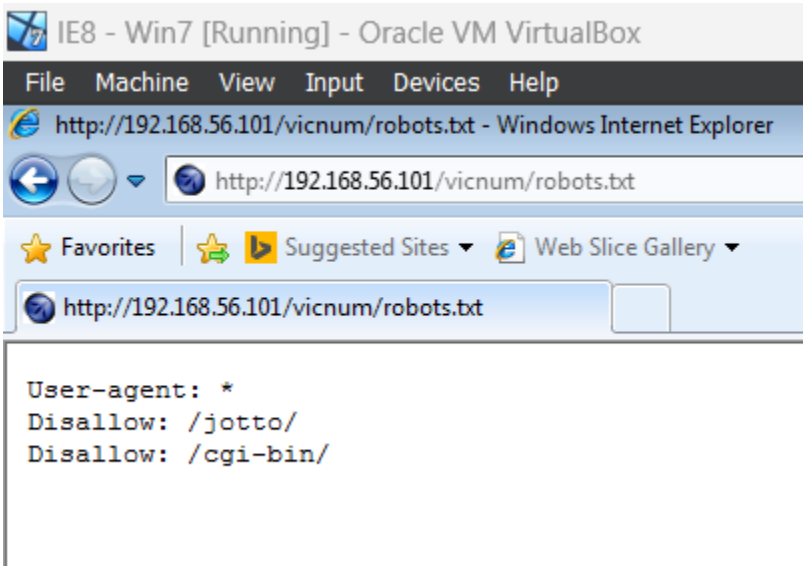

Figure 9: Access robots.txt

This shows that the /jotto/ and /cgi-bin/ is disallowed for all user agents. However, may be vulnerable to exploitation and can be checked by trying to access that directory.

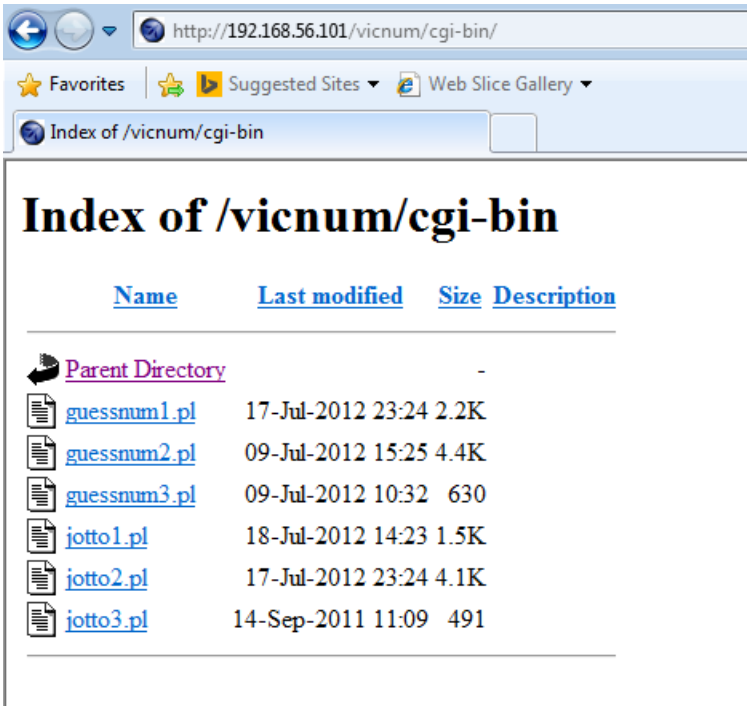

Figure 10: Accessing cgi-bin

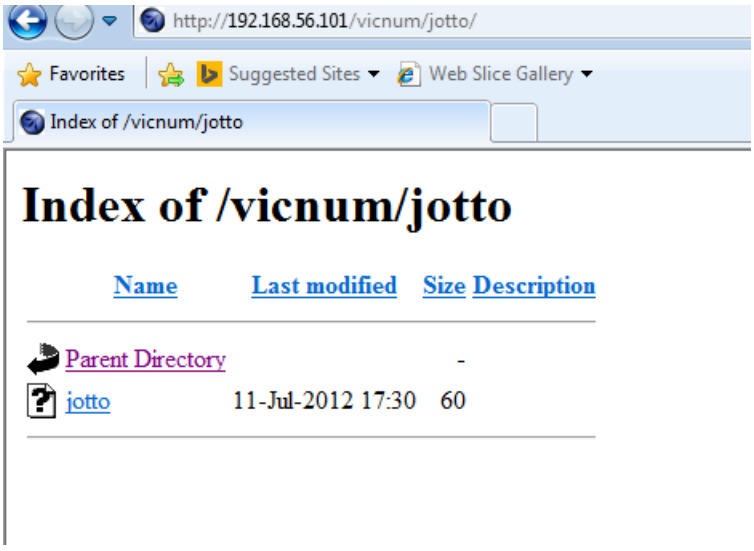

Figure 11: Accessing jotto

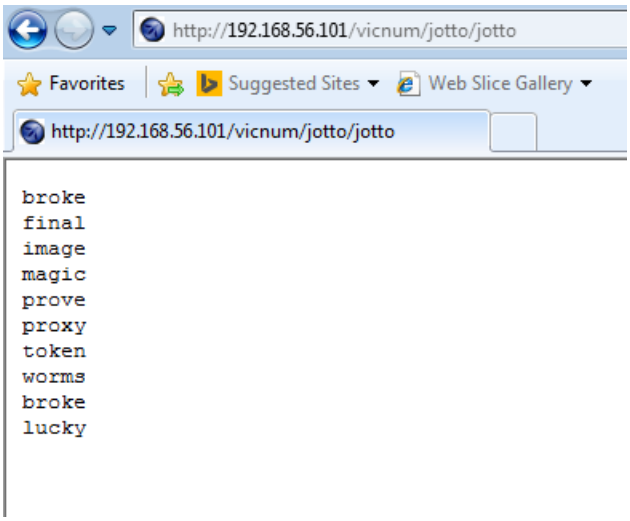

Figure 12: Exploited jotto answers

By seeing the robots.txt we can see the important files for the web application. By indexing and accessing jotto and cgi-bin directories, unauthorized access was gained and the answers were obtained using this approach.

A.2.4 Scenario Assessment

The DirBuster revealed information such as OWASP bricks, image directories, multiple JS and jQuery files. For a clinic it could lead to identifying the infrastructure of directories and could be used to exploit potential vulnerabilities. Also, if image directories are exposed that would compromise sensitive patient data and lead to reputation damage as it violates compliance regulations. Discovery of OWASP bricks could also lead to security flaws and attacks of the clinic's web services.

Looking at the code directly and having exposed credentials like the examples could lead to an attacker stealing those credentials and using it for unauthorized access. So, the medical services of a clinic could be disrupted due to a system breach and affect the overall reputation and trust of the clinic. Leaked credentials could also lead to: identity theft, financial losses and reputational damage.

Accessing the robots.txt file revealed a way to access directories which cannot normally be browsed through a web application directly. So, if the medical clinic has an exposed or unprotected robots.txt file it could be used by attackers to exploit the web services of the clinic in several ways such as: directory enumeration and access, discovering vulnerabilities, gathering information and

identifying entry points visible through the robots.txt like the /cgi-bin/ and /jotto/ directories. So, accessing directories in an unauthorized way will compromise patient data and leak sensitive directories and files of the clinic.

A.3 Port Scanning and Enumeration

A.3.1 Ports Identified

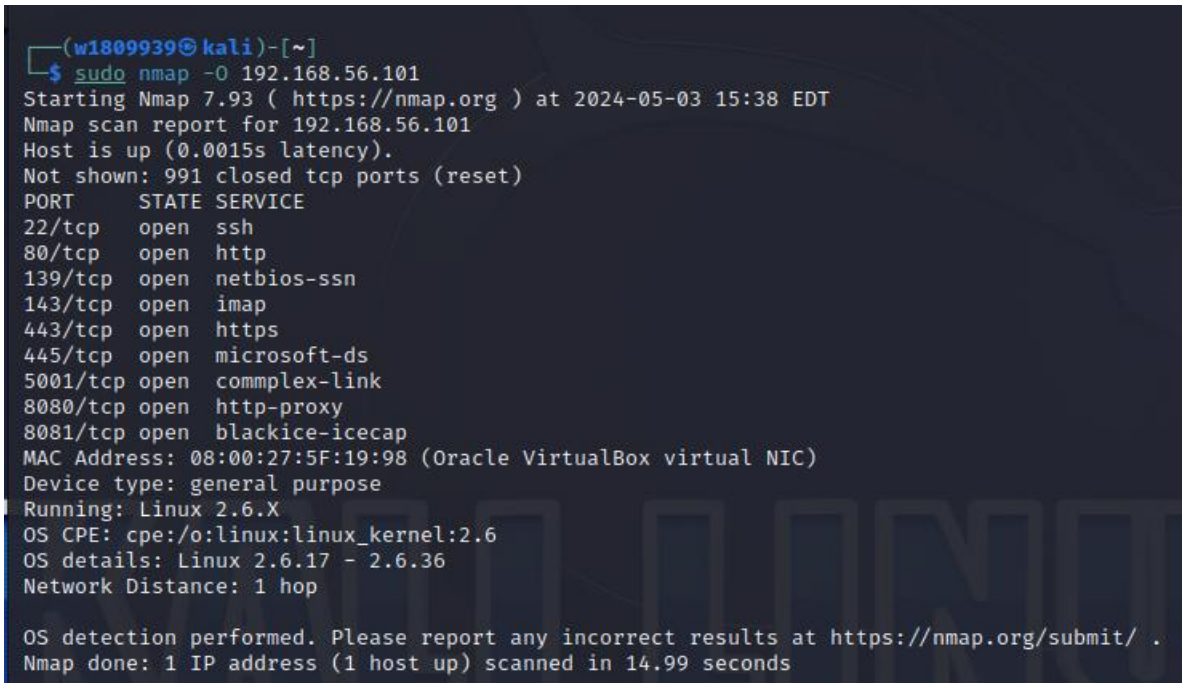

Figure 13: Investigating open ports and OS

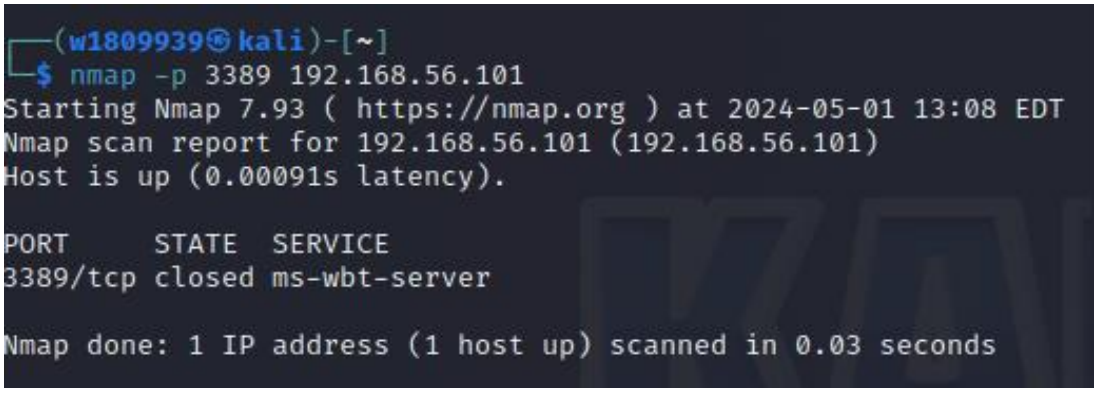

Figure 14: Investigating remote access port

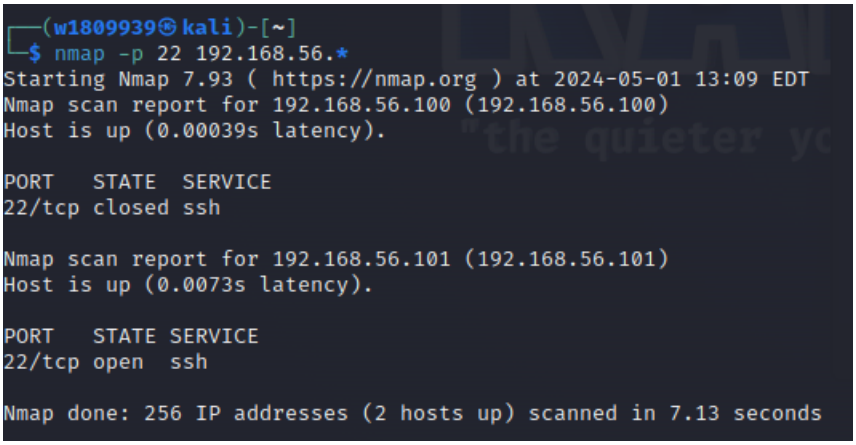

Figure 15: Identifying all machines on the network

Nmap or 'Network Mapper' is a tool used to provide insights about the network architecture and was utilized by the author. It helps in port scanning, service enumeration, OS detection and network mapping among many. The ports identified through this tool which are used by the server are listed below.

1. **Port 22/tcp (SSH)** – If an attacker finds this port to be open, they will try to gain unauthorized access through weak credentials and steal SSH keys and try to breach the vulnerability.
2. **Port 80/tcp (HTTP)** – An open HTTP port could lead to brute force attacks and DDoS attacks for running services on this open port.
3. **Port 143/tcp (IMAP) –** An attacker may intercept emails and try to gain unauthorized access from weak authentication through open services on this port.

Some of the other open ports identified include 139/tcp (NetBIOS-SSN), 445/tcp (Microsoft-DS), 8080/tcp (HTTP-proxy), 443/tcp (HTTPS), 5001 and 8081 ports. So having such open ports increases risk of unauthorized access and its best practice for any business or company with an online presence to minimize their open port to reduce the risks associated with open ports.

A.3.2 Open Ports and Threats Research

By definition, an open port refers to TCP or UDP port number that is configured to accept packets. Meanwhile, a port that declines connections or processes all packets yet drops them is a closed one (Tunggal, 2024). Having a port open is similar to leaving an open door to enter and exit a system that allows network communication flow to come in and out. This raises the possibility of potential

dangers that comes with a port being open as it could be used as an entry point by an attacker (Schrader, 2022).

Some threats that can be caused by having an open **Port 22/tcp (SSH)** include gaining unauthorized access from weak credentials or attempts to steal SSH keys that may compromise the system and lose control (Einorytė, 2024).

Additionally, open **HTTP ports such as 80/tcp** could lead to brute force attacks and DDoS attacks by using the port as a launching point. This is where attackers can overwhelm the application listening on open ports with large amounts of traffic and disrupt the system (Chappell, 2022).

So, attackers scan for open ports to find potential vulnerabilities and exploits. By accessing the services and versions of a machine, attackers carry out mapping and find vulnerabilities in a system. So for that purpose, attackers rely on open ports which are publicly accessible to analyze the infrastructure of the services on a machine (Tunggal, 2024).

A.3.3 Scenario Assessment

Port Identified	Applicability to Scenario
Port 22/tcp (SSH)	Gaining unauthorized access via this port or through stolen SSH keys could lead to the attacker gaining unauthorized access. This would compromise the system security and lead to a breach of patient and employee information which is personal.
Port 80/tcp (HTTP)	A brute force or DDoS attack carried out on this open port could lead to overwhelming traffic for the medical clinic application. This could lead to patients or potential customers not being able to access the application which threatens the availability.
Port 143/tcp (IMAP)	An attacker may intercept emails sent to the medical clinic via this port. This would compromise sensitive information shared between the clinic and its patients. It could also disrupt the communication service if exploited through weak authentication and credentials.

Table 3: Ports applicability to scenario

Overall, it is crucial for medical clinics to minimize the number of open ports it has in their web application and integrate strong security measures in place. Also constantly, updating the software and web application with patches and recent security updates would help mitigate risks associated with compromising patient data and operations within a clinic.

PART B – SERVER-SIDE EXPLOTS

B.1 Data Tampering

B.1.1 OWASP Mantra

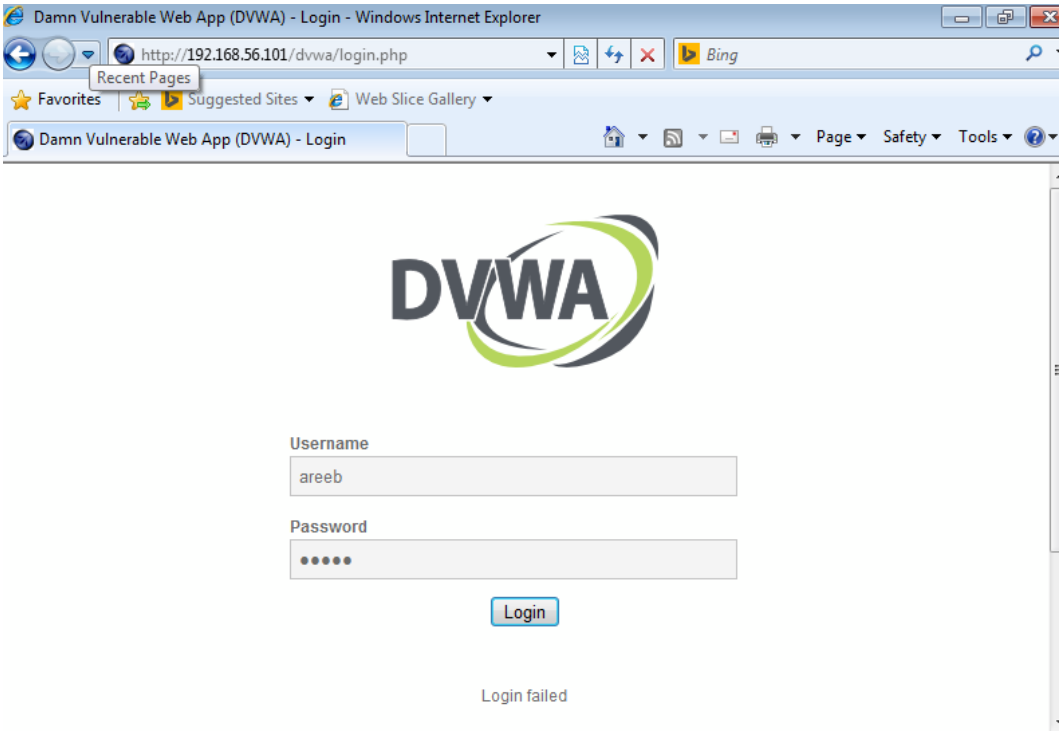

Figure 16: Application landing page

The login page of the site which data tampering is attempted is shown above. A tool called OWASP mantra will be used for this purpose to see if the data can be intercepted and exploited if possible.

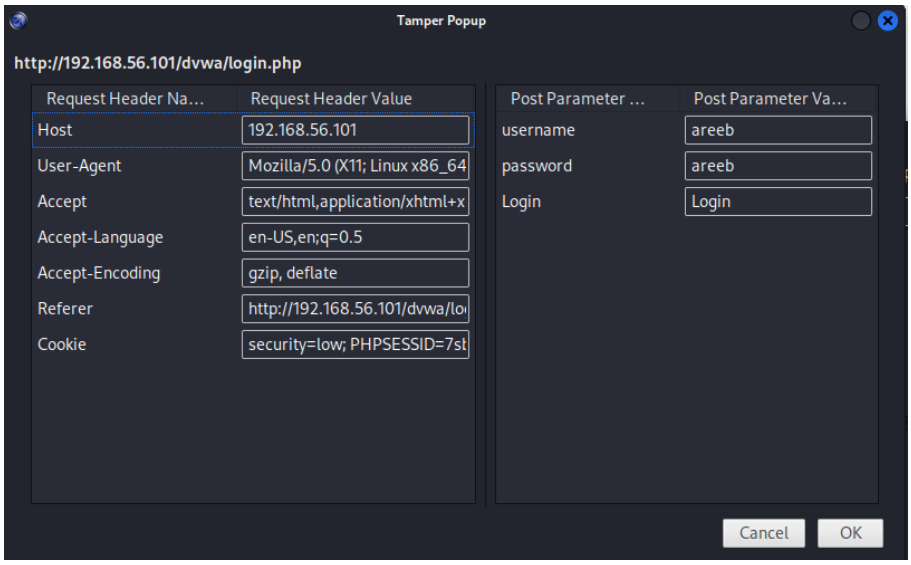

Figure 17: Intercepted request

Above illustration shows the intercepted request before submission where the data can be manipulated such as the username and password making it vulnerable. Therefore, this could be used to bypass invalid credentials with valid ones.

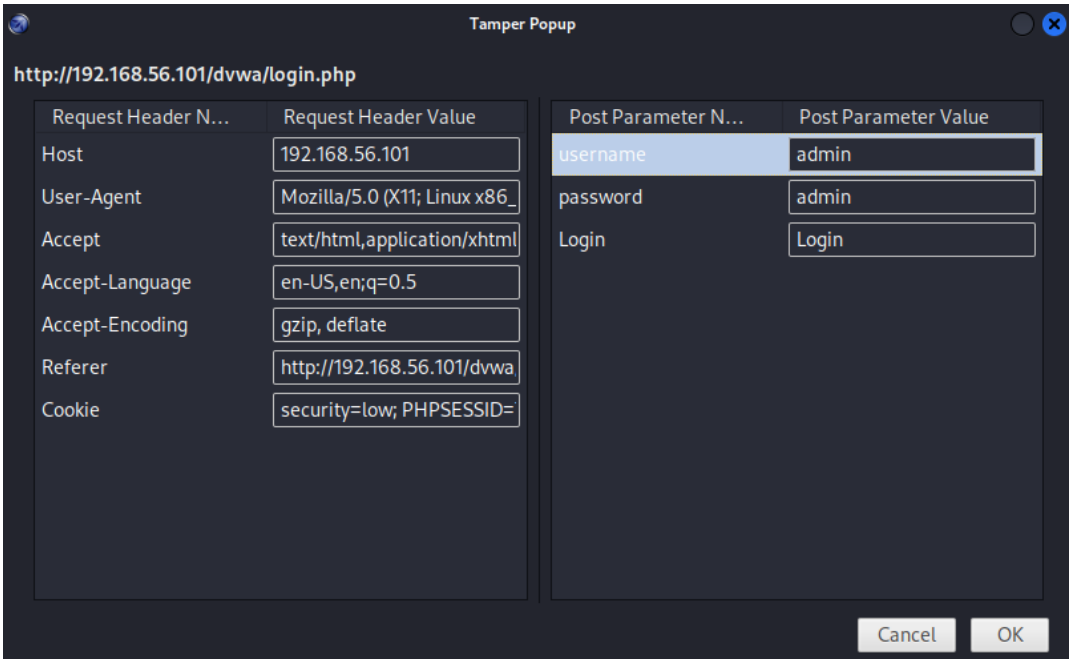

Figure 18: Manipulated request

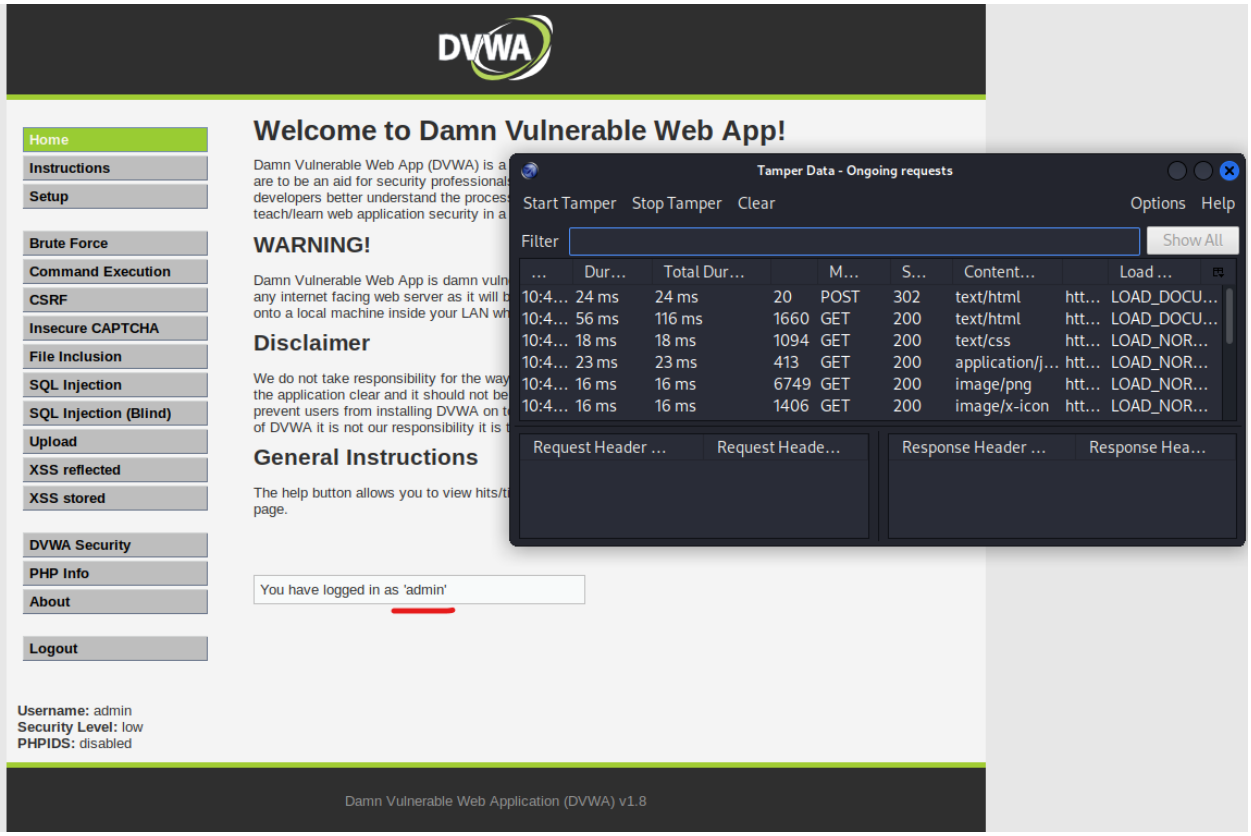

Figure 19: Successful login via manipulation

This proves that there is a major vulnerability in the sites login where unauthorized access can be gained by manipulating the POST request sent for a user to login to the site. By intercepting and changing the POST request values, data tampering and exploitation can be done.

B.1.2 Data Tampering Vulnerability Research

Data tampering vulnerability is a major weakness which allows the modification of data through unauthorized access (Kapsamer, 2022). Some of the key impacts it leads to include:

1. Data compromise and financial loss
2. Leakage of sensitive information
3. Unidentified attacker or hard to trace

Therefore, these key impacts could lead to overall reputational damage of an organization and should be minimized by applying strict security measures (Awati, 2022).

This breaches the cyber security tenet of **integrity** of the CIA triad because it modifies the accuracy of data initially set or sent through unauthorized entities.

B.1.3 Scenario Assessment

In context of our medical clinic, tampered data could be used to gain unauthorized access to the clinic's web application. This breach could lead an attacker to access personal patient data through tampered credentials.

Attackers also may modify the details of appointments which were scheduled. This could lead to a risk of the patient's health as an appointment may be urgent for a patient. Also, by gaining unauthorized access, the hospital data and lab reports will be compromised affecting the overall reputation of the clinic and violating the compliance regulations.

Since the medical clinic relies on the web application to schedule appointments and publish lab reports, data tampering could introduce incorrect doctor details and reports violating the integrity of the available data through the application. This could mislead patients and impact their appointment scheduling and payment.

B.2 SQL Injection

B.2.1 SQLi Implementation

When submitting the user ID field with only the number of '1', successful information was retrieved. Afterwards, the User ID could be given unexpected values like "1'" to see if the system is prone to vulnerabilities and is demonstrated below.

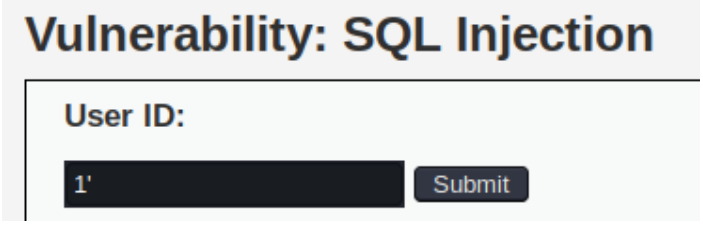

Figure 20: Unexpected input field

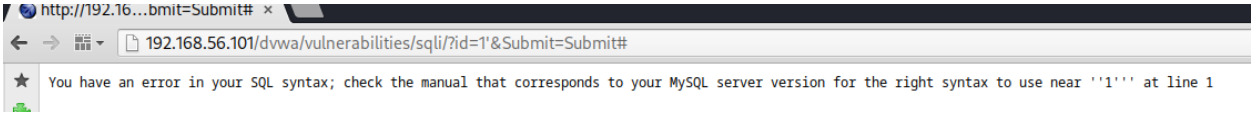

Figure 21: Response from unexpected input

An error message was shows stating that we have changed the well-formed query but it does not mean there can be a SQL injection here so a further step was conducted by inputting '1"' with two apostrophes as shown below.

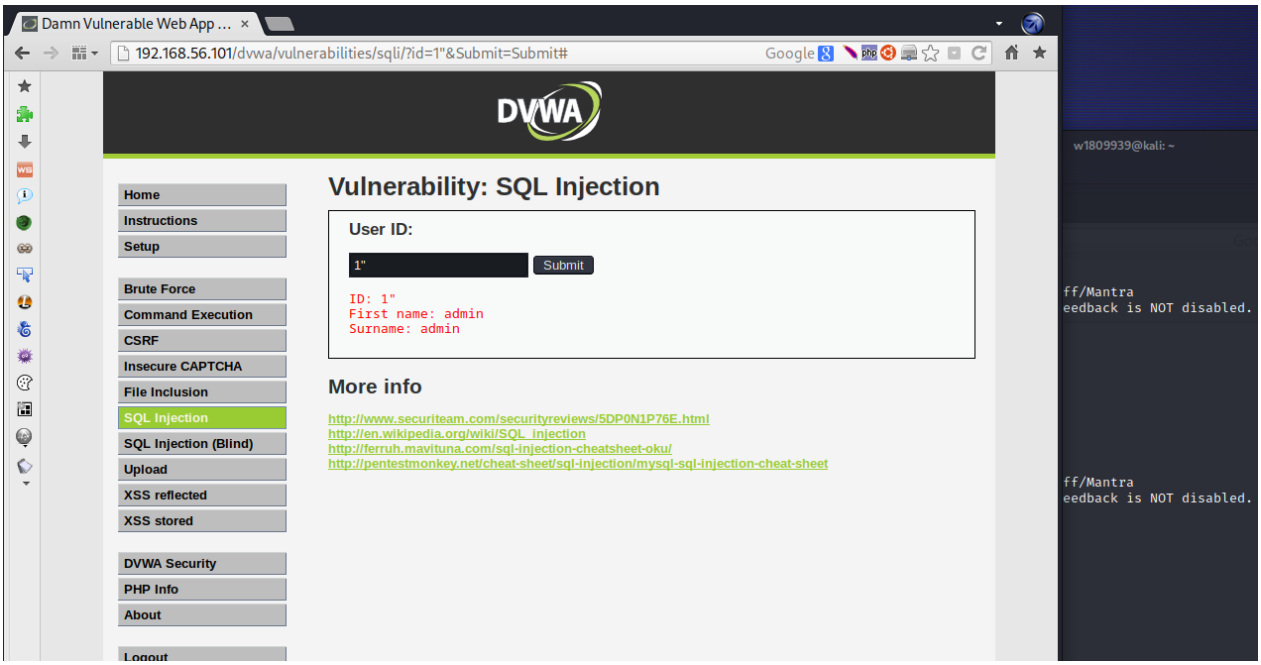

Figure 22: Two apostrophe invalid input

It worked successfully and returned results despite not having the correct exact value which means the application is surely vulnerable to attacks through SQL injection. Now attacks with advanced payloads could be attempted due to the confirmation of the vulnerability and is demonstrated below.

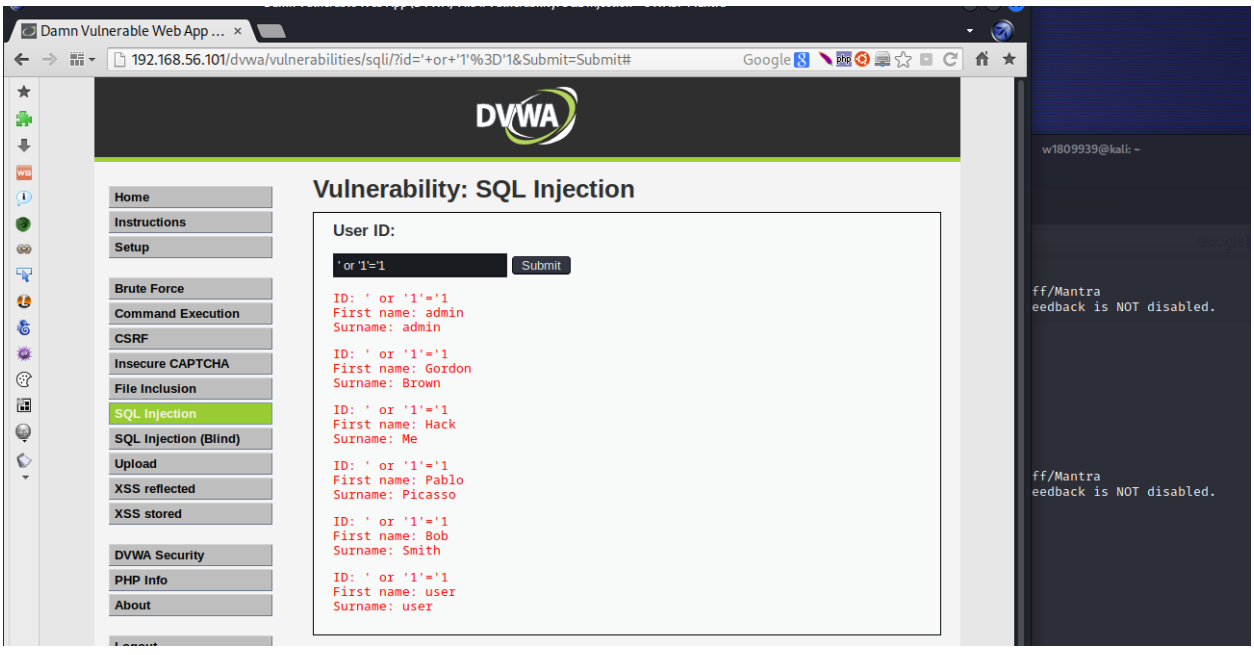

Figure 23: SQLi Exploitation

This manipulated query returned all the users available in that database by using: ‘ or ‘1’=’1. The outcome proved the successful manipulation of the User ID input field query and shows the SQL injection attack.

B.2.2 SQL Injection Vulnerability Research

SQL injection vulnerability is a weakness in web security systems that allows hackers to change a database by entering dangerous SQL commands into the input box of a website (Yasar, 2023).

This kind of vulnerability facilitates the theft, destruction or alteration of valuable data through unauthorized access to databases (Choudhary, Verma and Meena, 2021). The attack employs methods where unauthorized users can enter, modify or delete information accordingly to surpass securities measures and obtain sensitive materials or even perform administrative roles within and organization’s DBMS (Rua et al., 2017).

It violates the tenet of **confidentiality** because through this attack, unauthorized users gain access and view sensitive information. For more than 20 years now it has remained a hazardous form of cyber-attacks and developers need to use parameterized queries along with stored procedures that do not allow for such kind of attacks to happen again (Sadotra and Sharma, 2017).

B.2.3 Scenario Assessment

For a medium sized medical clinic, attackers could gain various types of sensitive information via SQLi including:

- Patient Medical Records
- Personal Identifiable Information (PII)
- Financial and Transactional Data (billing)
- Appointment Schedules

The database connected to the clinic’s web application could be exploited by injecting malicious commands by the attacker gaining unauthorized access to sensitive information of the clinic. Since the web application handles patient data and clinic’s daily operations it would lead privacy violations, identify theft and legal repercussions.

By modifying and manipulating SQL queries and carrying out SQL injection attacks it would pose significant threats and risks to patient privacy and the clinic’s daily operations and developers need

to use parameterized queries along with stored procedures that do not allow for such kind of attacks to happen again.

B.3 XSS Scripting

B.3.1 XSS Scripting Exploitation

An input field value could be entered to see if there is a vulnerability to XSS scripting. Below illustration demonstrates a sample input.

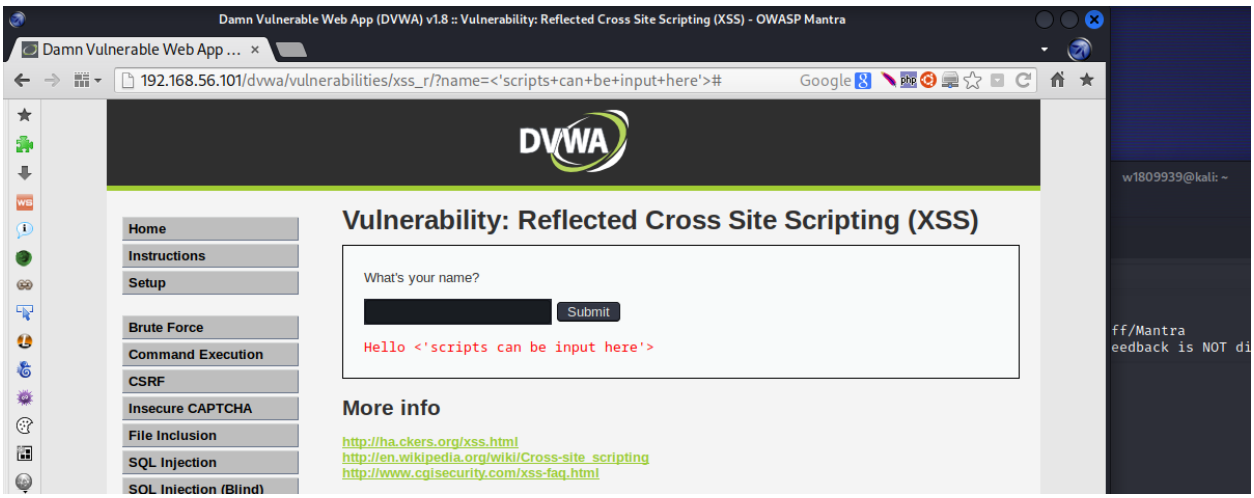

Figure 24: Testing XSS scripting vulnerability

Here it can be seen that the output results given is also returning the angle brackets (<>), which are used to write scripts. Therefore, this proves it is prone to XSS scripting attacks and we can inspect it further in the page source to see if it is getting read from server side as well.

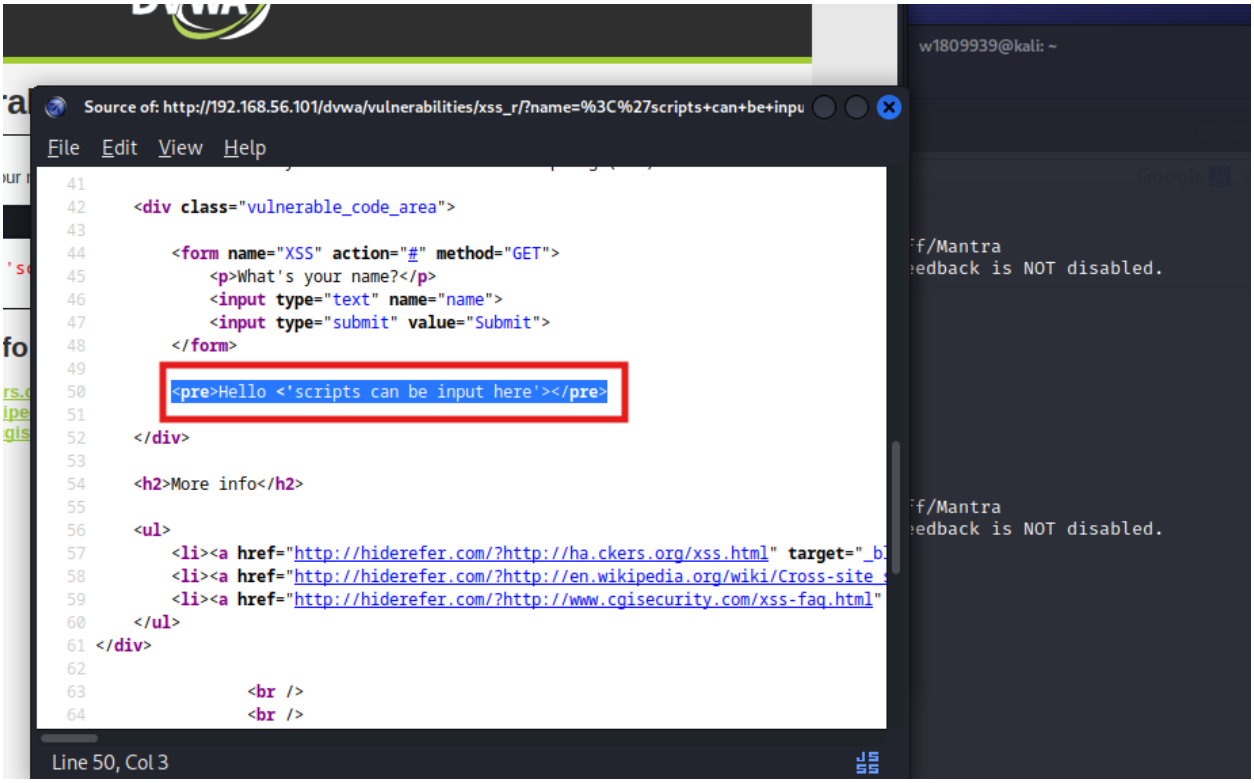

Figure 25: Viewing page source (XSS)

Since it is visible here, it is further proved to be prone to XSS scripting attacks and can be attempted to be exploited using an advanced script such as returning the cookie details as shown below.

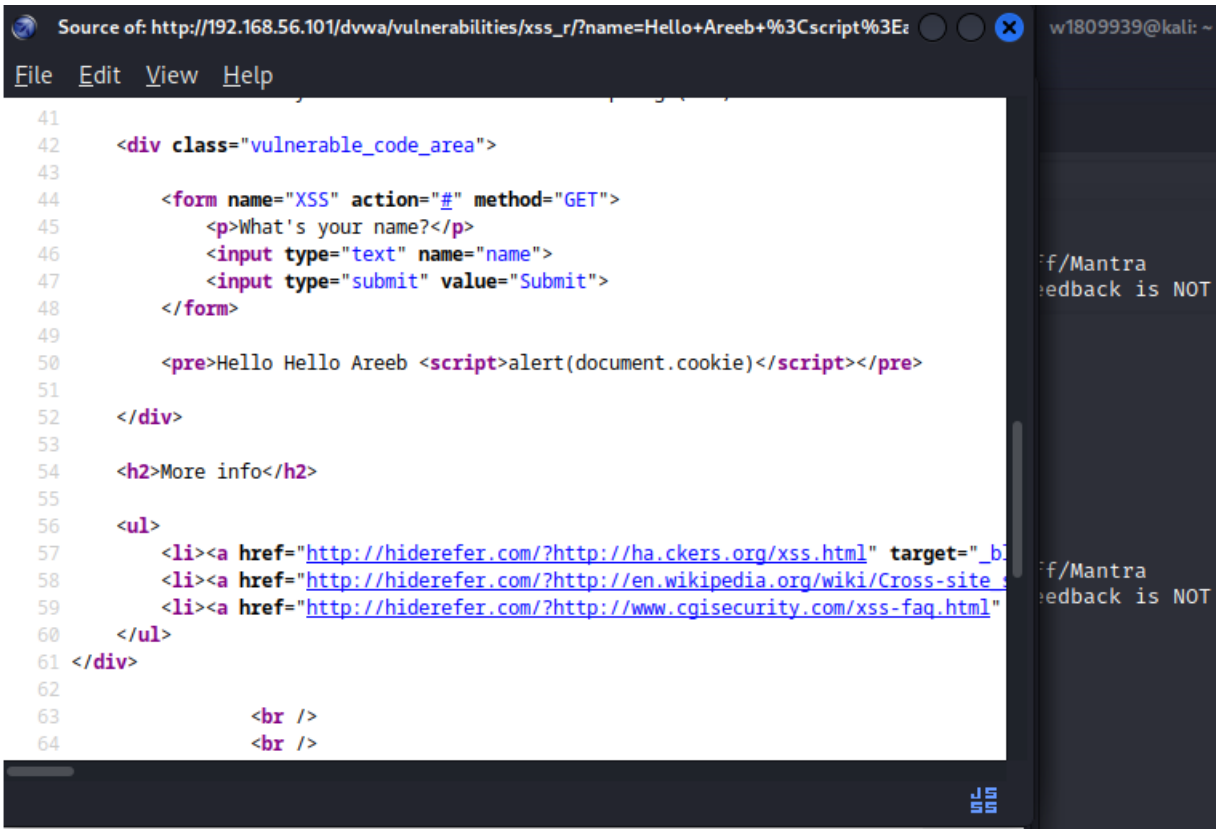

Figure 26: Cookie alert script (XSS)

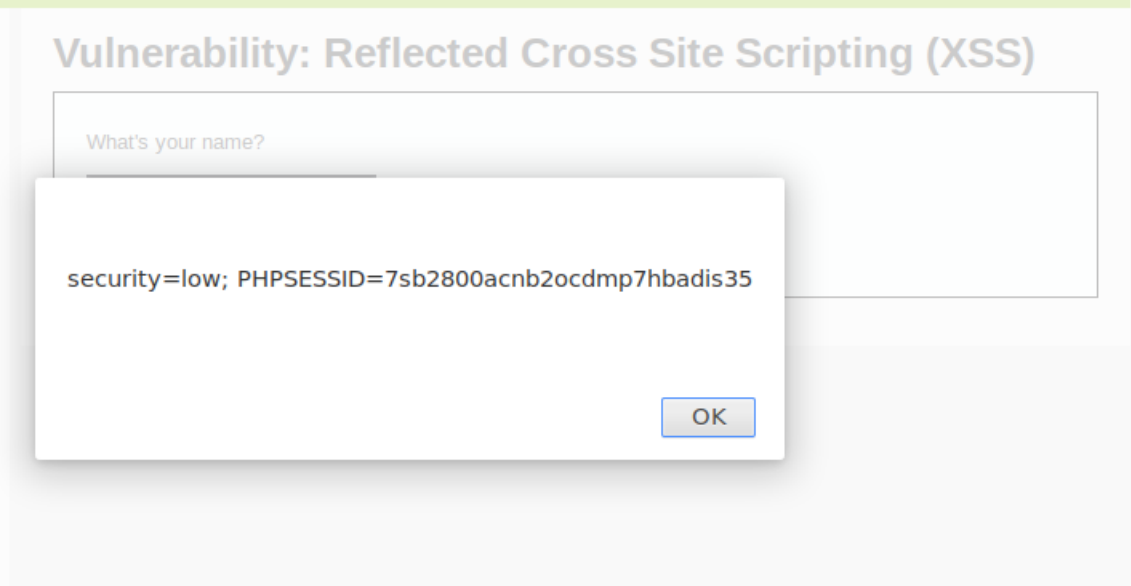

Figure 27: Returned cookie details

This successfully returned the document cookie details via exploitation and shows a significant risk in the application as an attacker could input harmful scripts to disrupt the application.

B.3.2 XSS Scripting Vulnerability Research

XSS or Cross Site Scripting is a vulnerability that allows the attacker to insert harmful scripts into the website pages that users view. The 3 main types of CSS attacks include: Reflected XSS, Stored XSS and DOM-based XSS (Walkowski, 2020). This can be used to steal passwords, session tokens, or change a webpage's data.

The XSS problem is caused when validation of user input is not done properly and securely resulting in the execution of the attacker's code in other user's browsers (Gupta and Gupta, 2017).

It may cause different types of attacks, e.g. session hacking, site unavailability. For XSS be avoided, developers need to practice input validation, output encoding and implement script security policies to prevent scripts from executing unauthorizedly (Mitropoulos et al., 2016) The scripts used can violate all tenet principles of the CIA triad such as: **confidentiality, integrity and availability** due to the nature of its attacks.

B.3.3 Scenario Assessment

An attacker could inject malicious scripts into the clinic's web application if the vulnerability ceases to exist. This could be used to modify patient data and steal sensitive information by exporting the data via scripts.

Furthermore, the clinic web services can be disrupted via scripts that can be used to crash the web servers and the application. It could also lead to phishing attacks and spread malware as script can be used to redirect a patient or customer to a malicious site or prompt them to download malware unknowingly.

Overall, this would lead to a data leak, an alteration of records, an acquiring of the sensitive information, a disrupting of services, and the expansion of malware. This vulnerability raises concerns over personal healthcare, the clinic's good name and image, and the financial integrity. To overcome these, security measures such as input validation and thorough security testing are vital to stop such attacks from being carried out successfully.

B.4 Other Vulnerabilities

B.4.1 Buffer Overflow

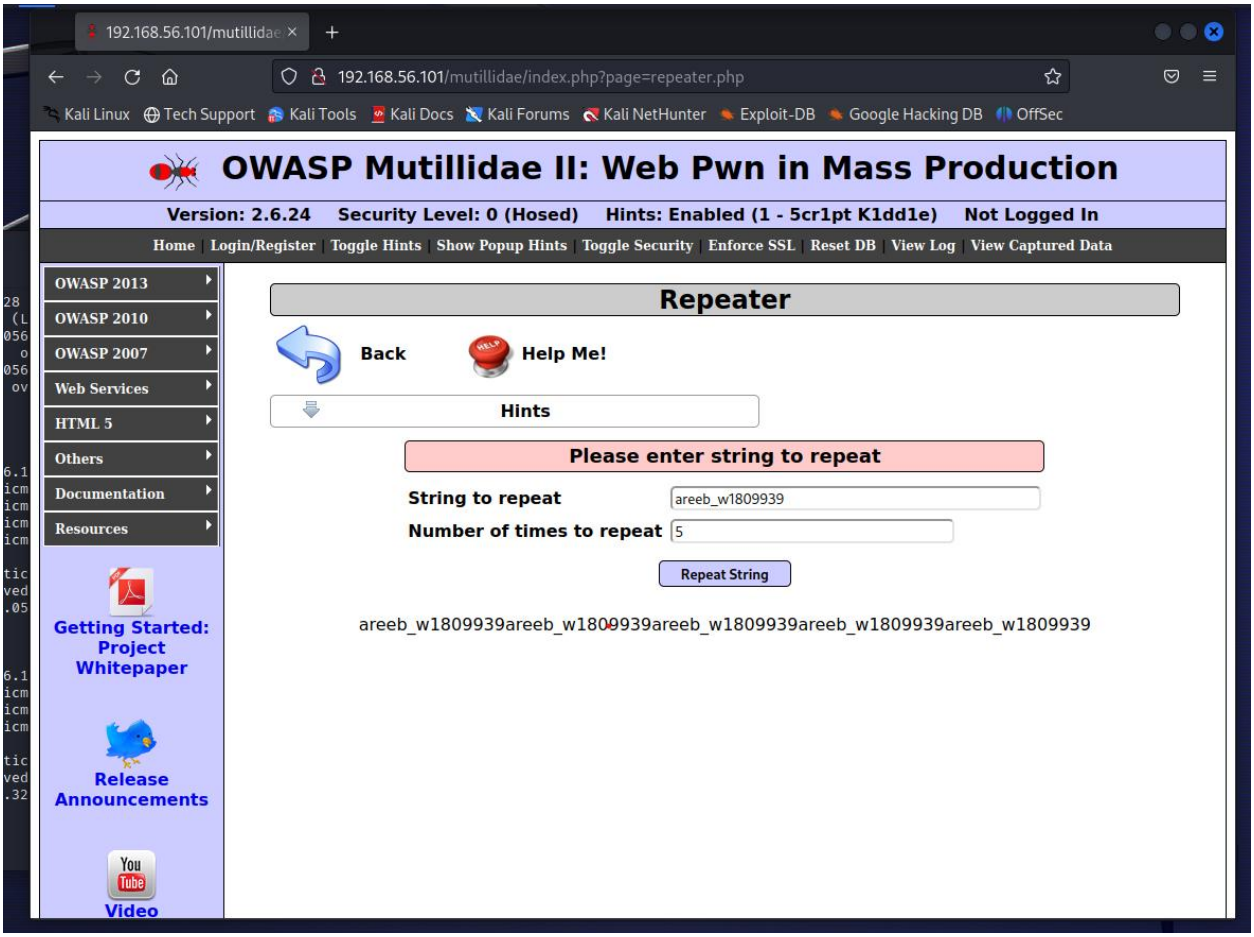

Figure 28: Buffer overflow demonstration

Buffer overflow is when more data is attempted to be written for a memory buffer with a fixed size of memory (Cobb, 2021). Attackers exploit this by sending excess data which may lead to a system crash amongst various other consequences.

B.4.2 Command Injections

Figure 29: Command injections demonstration

For a vulnerable application, sometimes you can execute OS commands as input data to try and exploit the system. Above screenshot shows output may be vulnerable to OS commands which is attempted next.

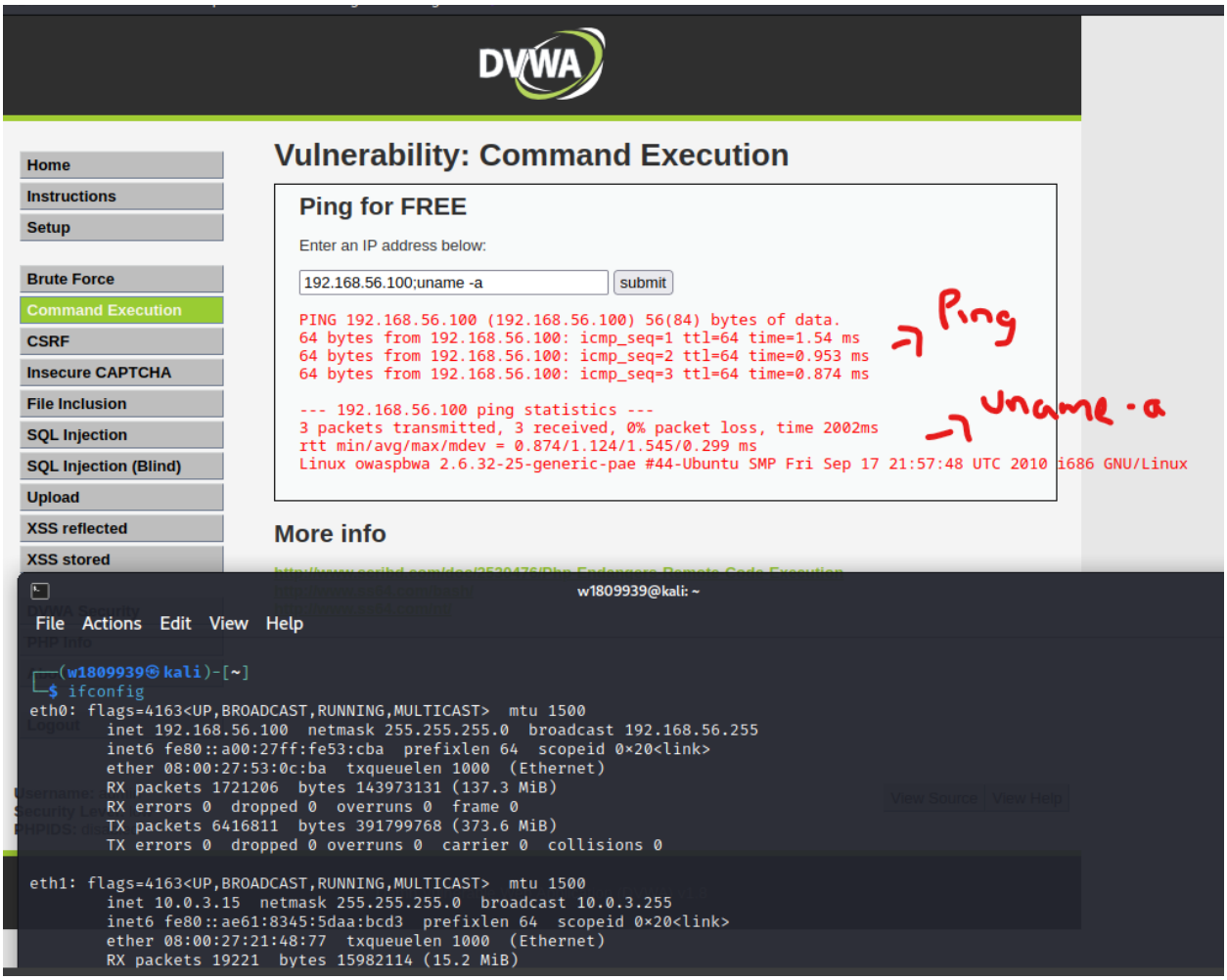

Figure 30: Injecting OS command

This proves there is a command injection vulnerability as ping and uname -a results were directly given.

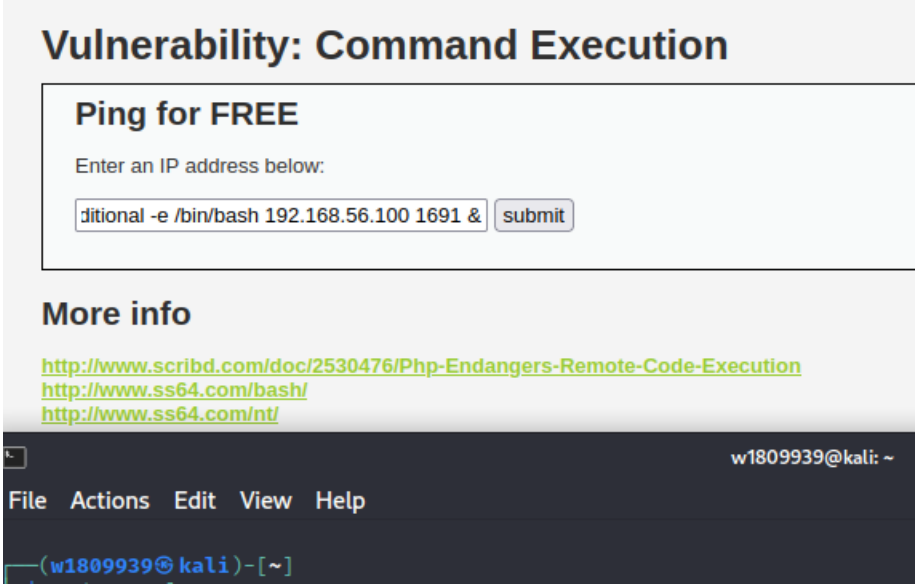

Figure 31: Exploiting the vulnerability

```
┌──(w1809939㉿kali)-[~]
└─$ sudo nc -lp 1691 -v
[sudo] password for w1809939:
listening on [any] 1691 ...
192.168.56.101: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.100] from (UNKNOWN) [192.168.56.101] 37602
uname -a
Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010 i686 GNU/Linux
cd ~
ls
ClientAccessPolicy.xml
MCIR
OWASP-CSRFGuard-Test-Application.html
WackoPicko
animatedcollapse.js
assets
bWAPP
```

Figure 32: Results of OS command injection

Here information regarding the OS and access to directories was gained which shows a major vulnerability in the system.

B.4.3 Scenario Assessment

Buffer Overflow attacks on the clinic where an attacker could input excess data and exploit it by sending overwhelming input could lead to affecting the memory capacity of the application. This could lead to a possible system crash and data loss and manipulation disrupting the services of the clinic. This leads to a direct violation of the cybersecurity tenets: availability, integrity and confidentiality.

Command injection attacks on the clinic's web application where an attacker may inject commands into input fields may lead to exposure of the directory files and OS details of the clinic's web application. This would lead to a data breach and the attacker could even manipulate the data and gain complete access to the root. This leads to a direct violation of: **confidentiality, availability and integrity** principles.

B.5 Cryptanalysis Attack

B.5.1 Cryptanalysis

We can demonstrate a cryptanalysis attack from this challenge using the security shepherd tool.

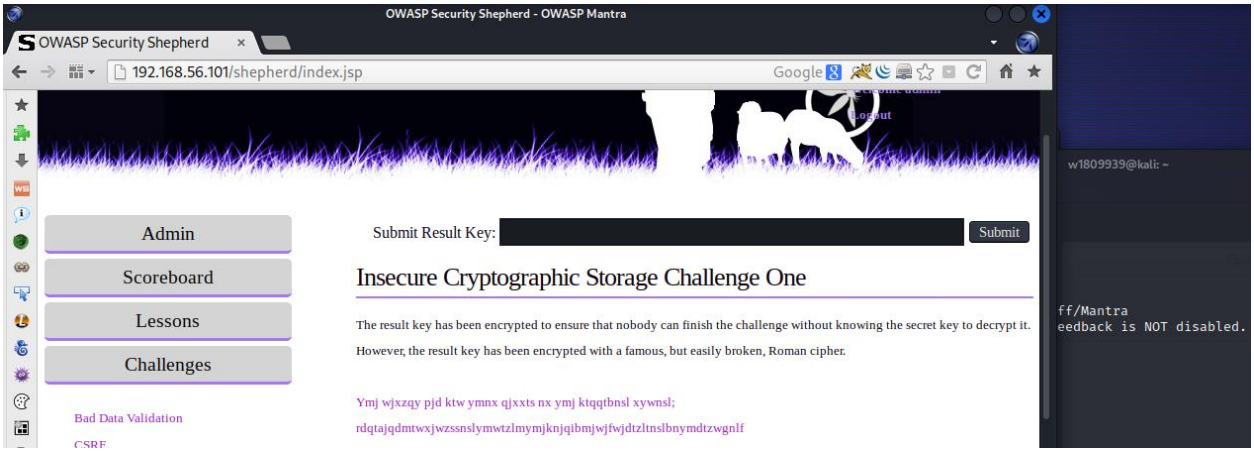

Figure 33: Cryptanalysis challenge

From the given ciphered text, we can use external tools such as: https://raw.org/tool/caesar-cipher/ to decrypt the ciphered text as shown below.

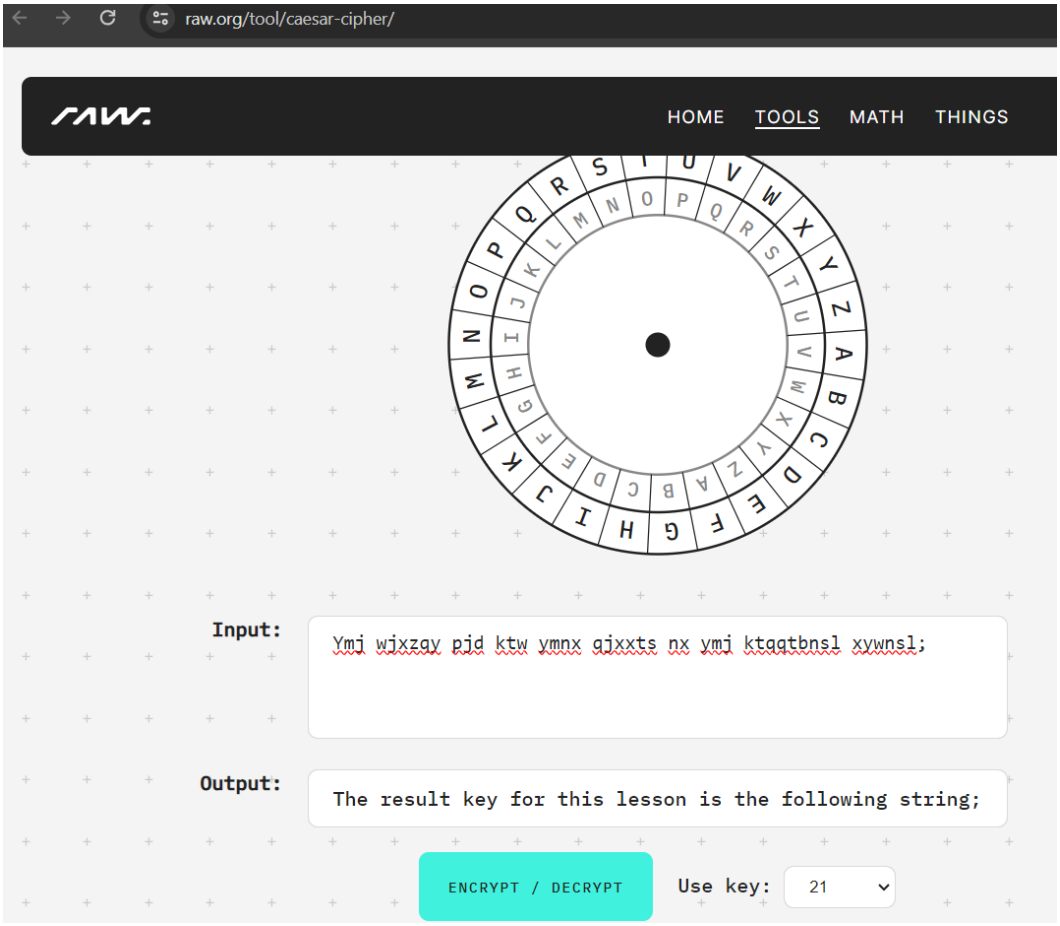

Figure 34: External tool for decryption

By selecting different keys, we can arrive with the decrypted message. In this by choosing the key as 21, it said that the result key is given in the next line so then the next line could be attempted to decrypt as shown below.

By selecting ‘guess’, it seemed to give a meaningful string which could be used in the next step to see if it is the result key.

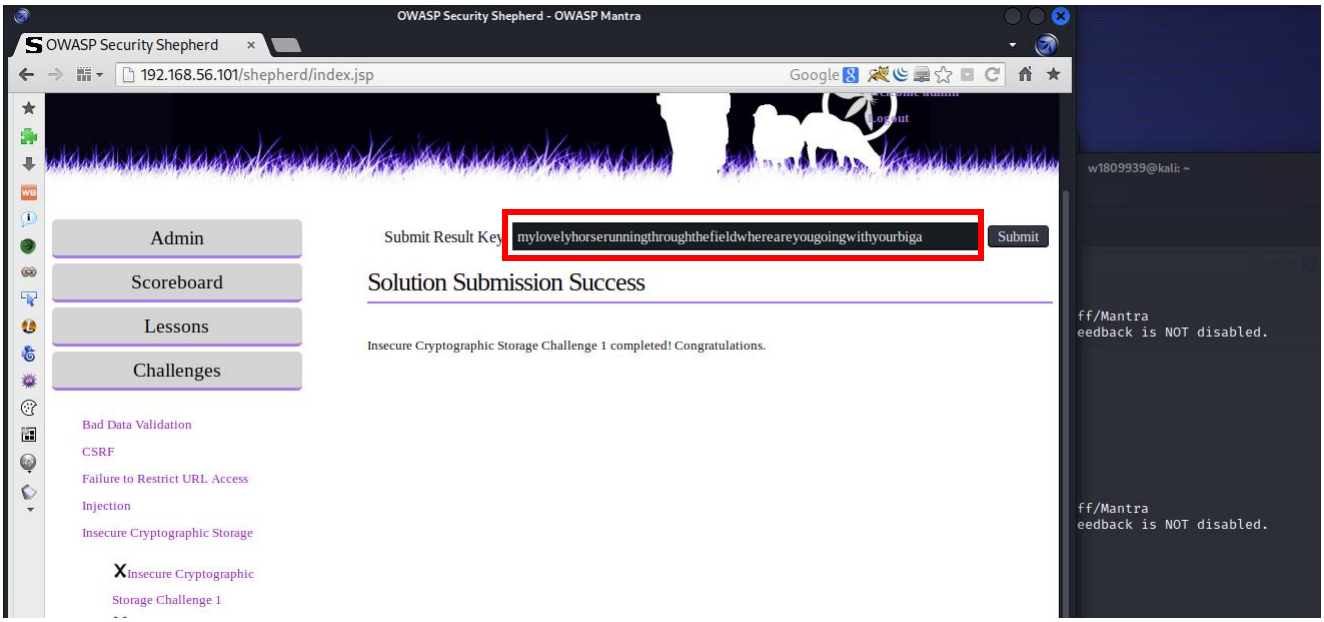

Figure 35: Result key

The result key was successfully figured through using an external tool such as shown in the above illustrations. Therefore, if we apply the same concept to a real-world scenario where encrypted usernames and passwords are stored and retrieved through SQL injections and data gathering phases then we could decrypt the password using an external tool. Such an example is shown below.

- Encrypted password retrieved – ‘cnffjbeq”

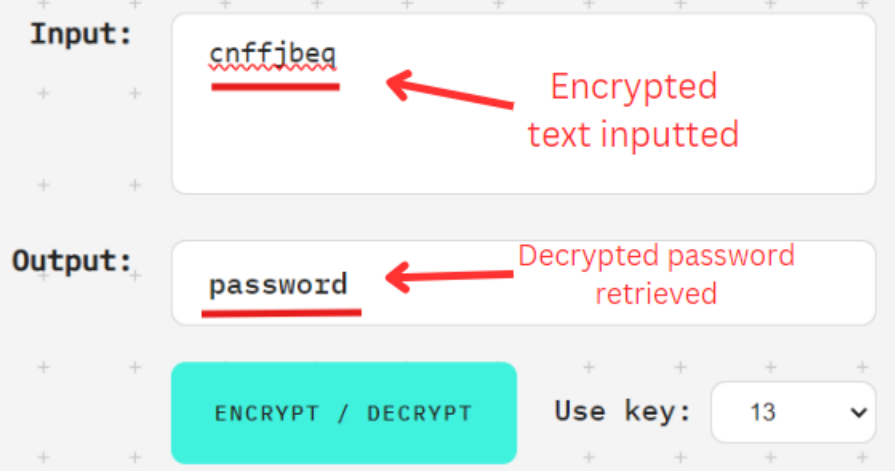

Figure 36: Decrypting a password

This shows a means by which cryptanalysis attacks can be conducted and passwords can be retrieved via deciphering the encrypted stored credential.

B.5.1 Scenario Assessment

An attacker that gains encrypted details from the clinic's web application during the reconnaissance phase could potentially attempt to exploit such encrypted details and credentials. Successful decryption methodologies and techniques such as using an external tool could compromise confidentiality and lead to data breach of patients using the decrypted credentials.

This violates the **confidentiality** tenet of the CIA triad as the attacker could gain unauthorized access via the decrypted data gathered.

PART C – CLIENT-SIDE EXPLOITS

C.1 Man in the Middle Attack (MiTM)

C.1.1 Ettercap

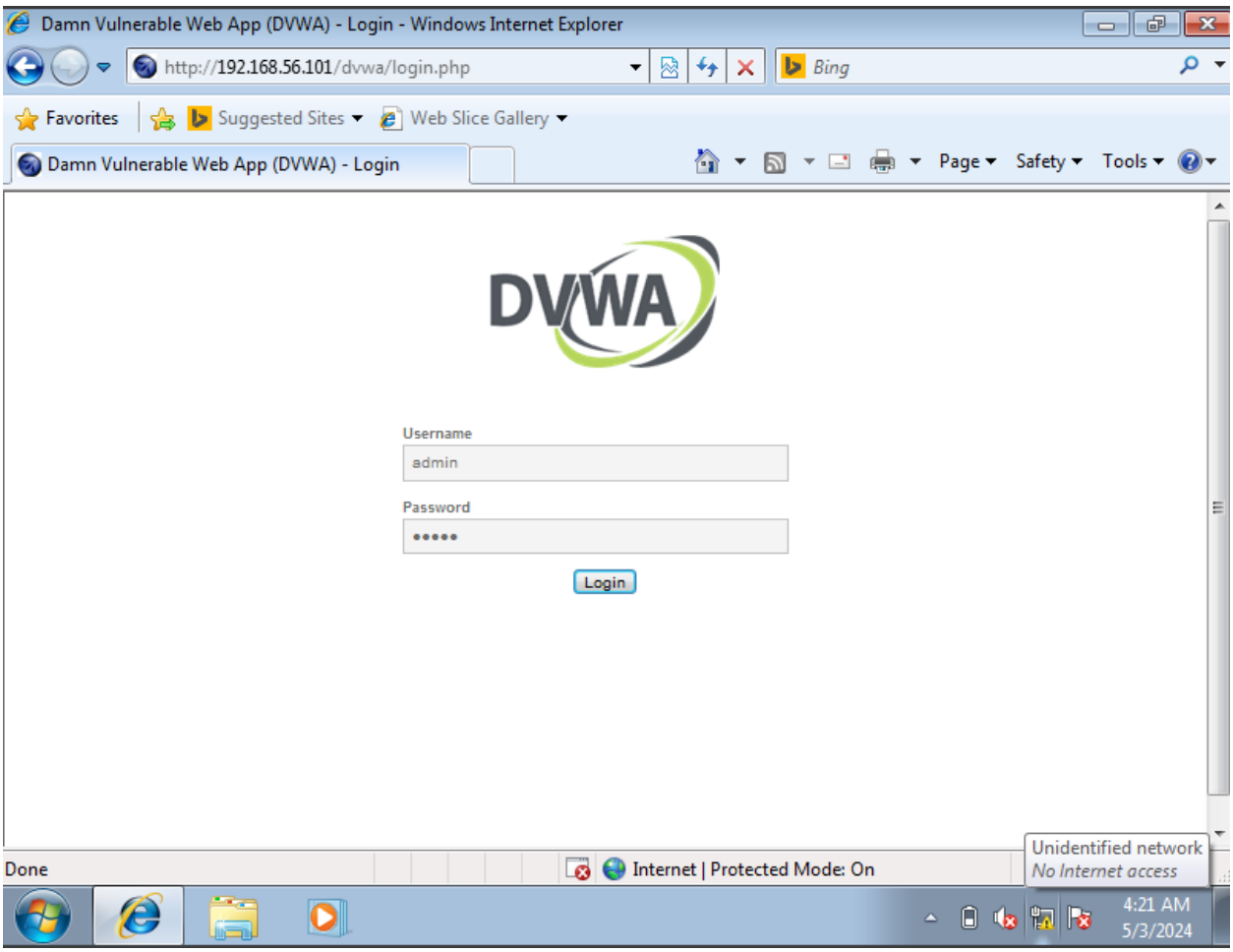

Figure 37: Target application (MiTM)

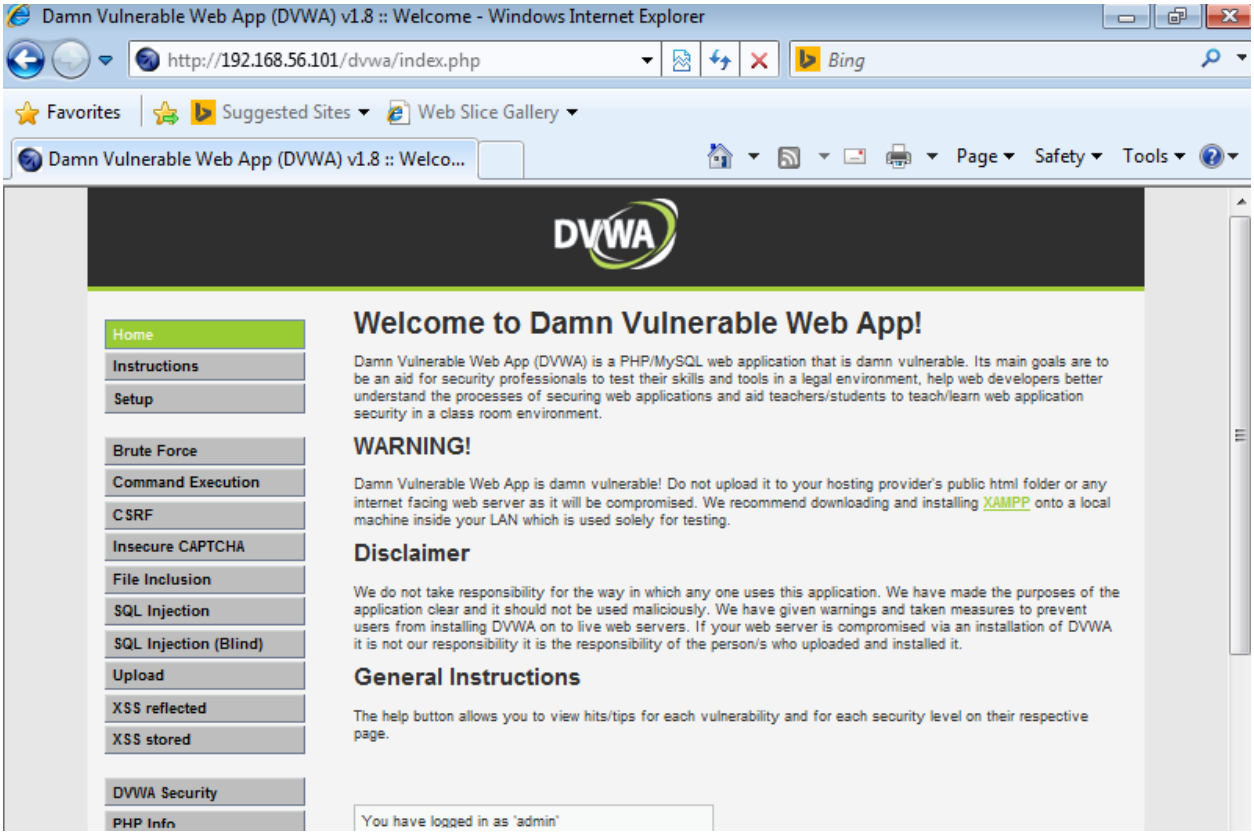

Figure 38: Logging in as admin

By using Ettercap while logging in, an attacker may be able to intercept the login information that was used. In this case, an admin was logged in and Ettercap was setup to capture this information.

- Target 1 was set to the vulnerable server: 192.168.56.101 (OWASP machine)
- Target 2 was set to the vulnerable client: 192.168.56.102 (Windows)

Below illustrations show the setup and results of the login request via Ettercap.

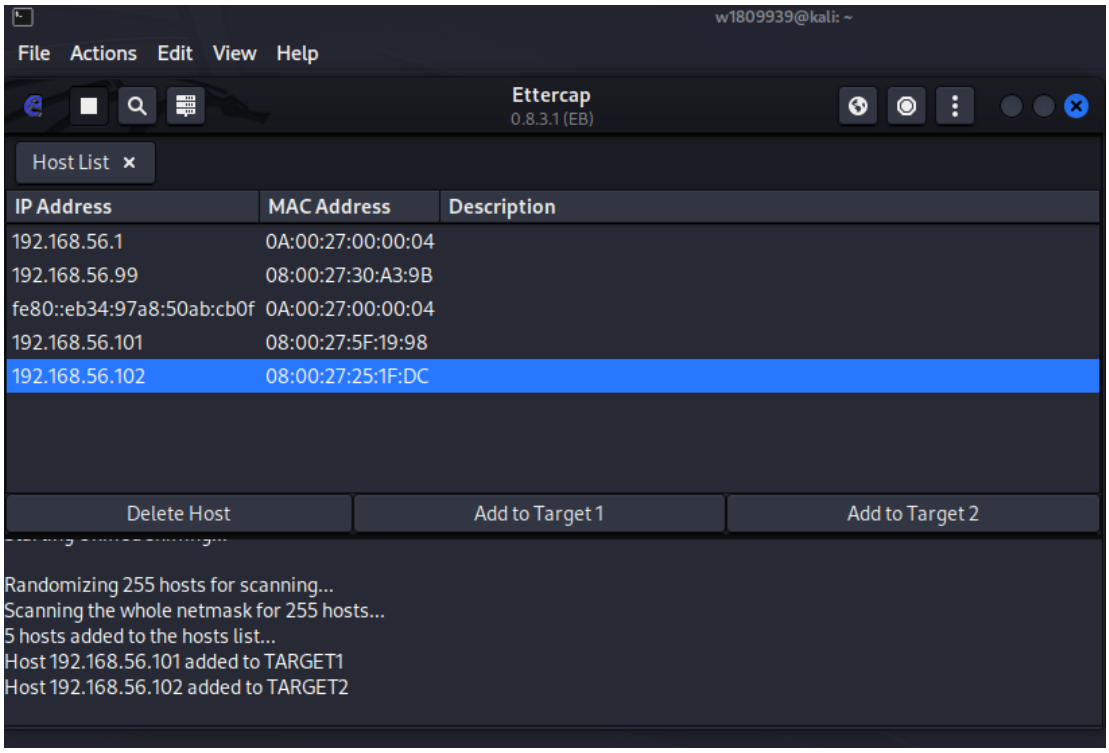

Figure 39: Ettercap setup

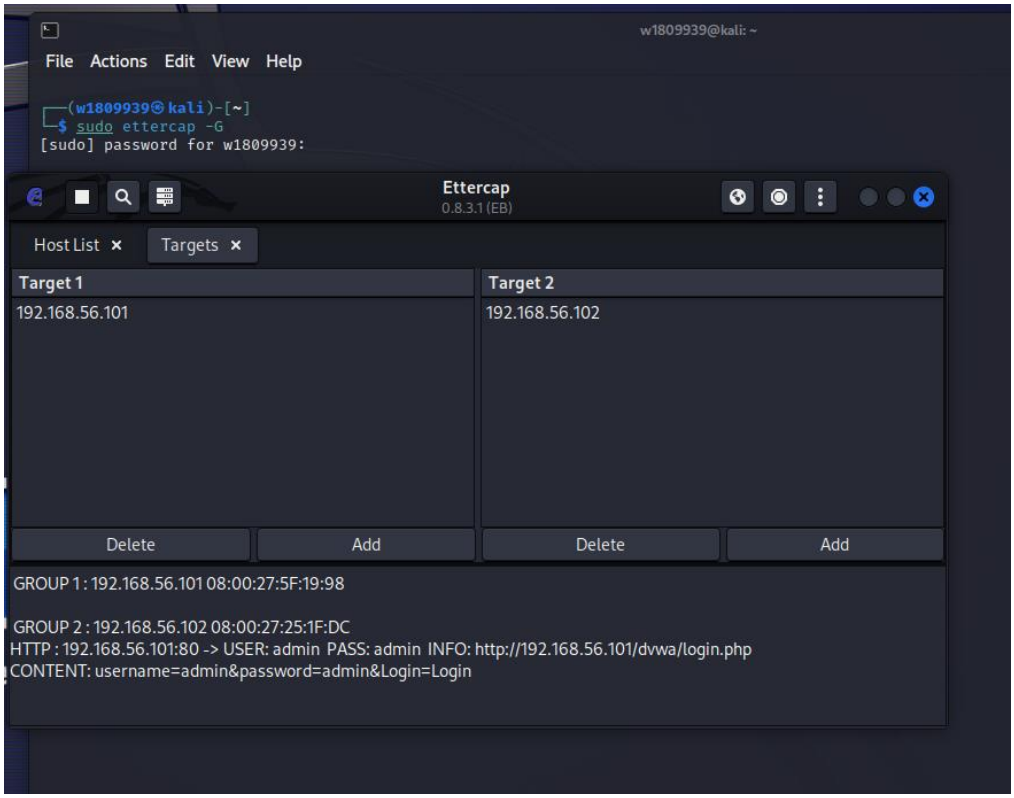

Figure 40: Ettercap interception

In the last figure, you can see the intercepted credentials which was:

- Username=admin
- Password=admin

Therefore, this shows a clear and successful demonstration of a man in the middle attack. By using Ettercap and similar tools, an attacker could position themselves between two parties where the attacker could sniff and eavesdrop on the communication requests and use them for malicious intent.

C.1.2 Scenario Assessment

An attacker could position themselves between a client computer such as a patient and the clinic's server when a request is being sent. By intercepting and manipulating the traffic the attacker could gain sensitive clinic information such as:

1. Login Credentials: This can be used to gain unauthorized access and steal and manipulate records affecting the confidentiality and integrity.
2. Patient Information: Personal details can be intercepted and viewed without permission violating legal and privacy regulations.
3. Transactional Data: Credit card details can be stolen if exposed via the request while a patient makes a transaction leading to theft and financial loss.

Therefore, carrying out a MiTM attack on a clinic's web application using methods such as spoofing, and ARP poisoning could lead to overall reputational damage for the clinic and have serious legal and ethical repercussions since patients trust a clinic to uphold and treat their data with utmost security. Measures such as encryption, network monitoring and strong security should be integrated into the clinic's web application to avoid such cyber threats and attacks.

C.2 Social Engineering Attack

C.2.1 Social Engineering Toolkit

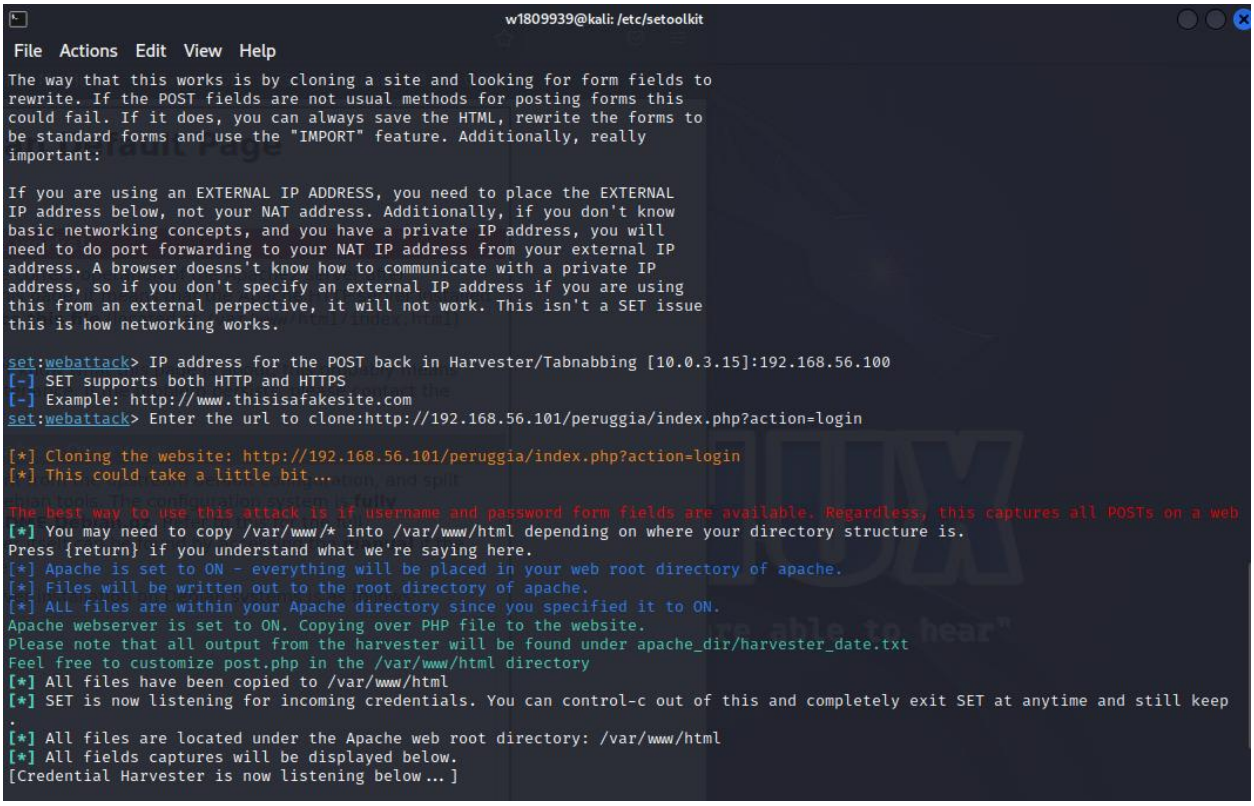

Figure 41: Listening on cloned website

The author utilized setoolkit to clone a websitie and use credential harvester to listen the transmission. Below illustration shows the cloned site which tricks the user to believe it's the original page and input credentials.

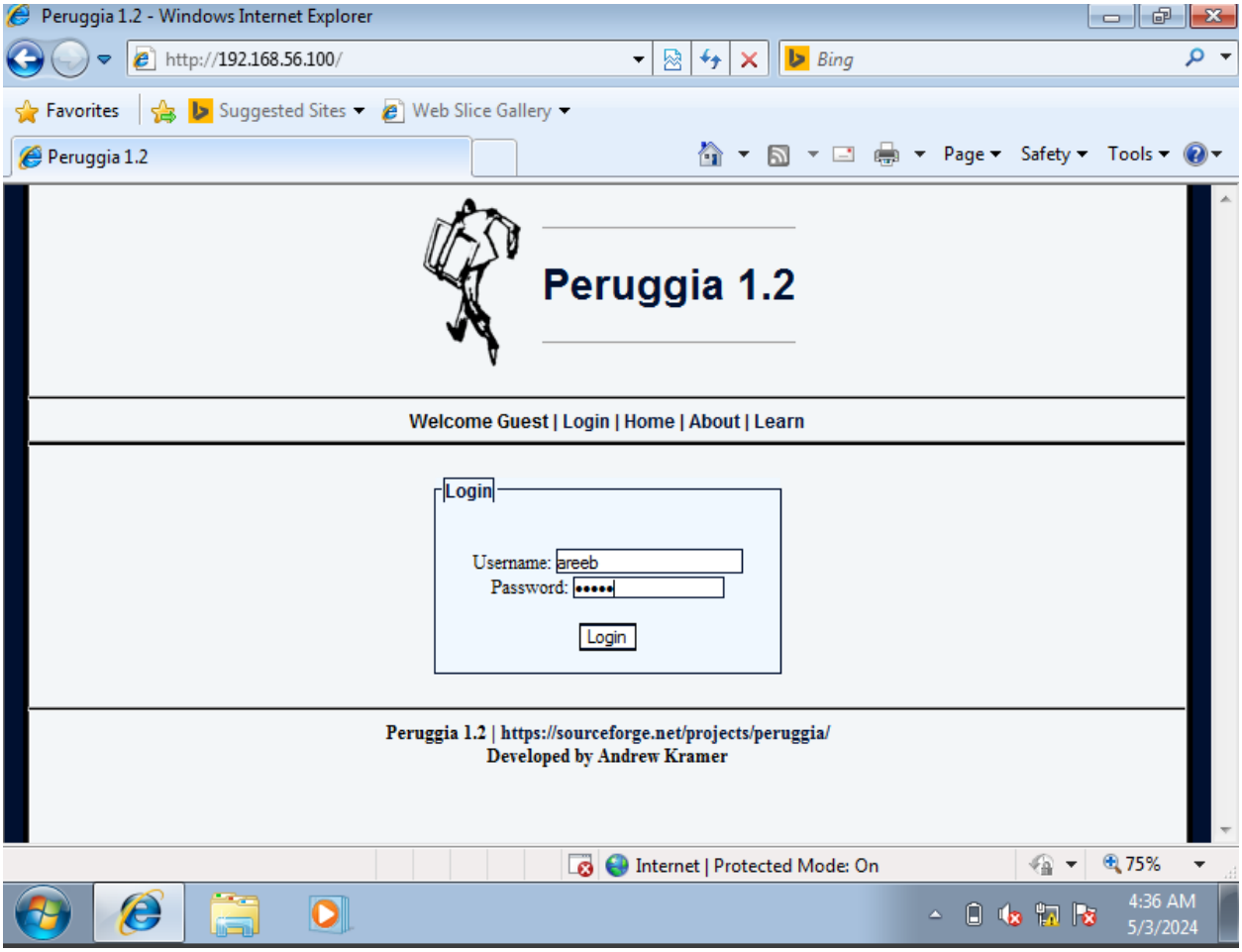

Figure 42: Cloned site

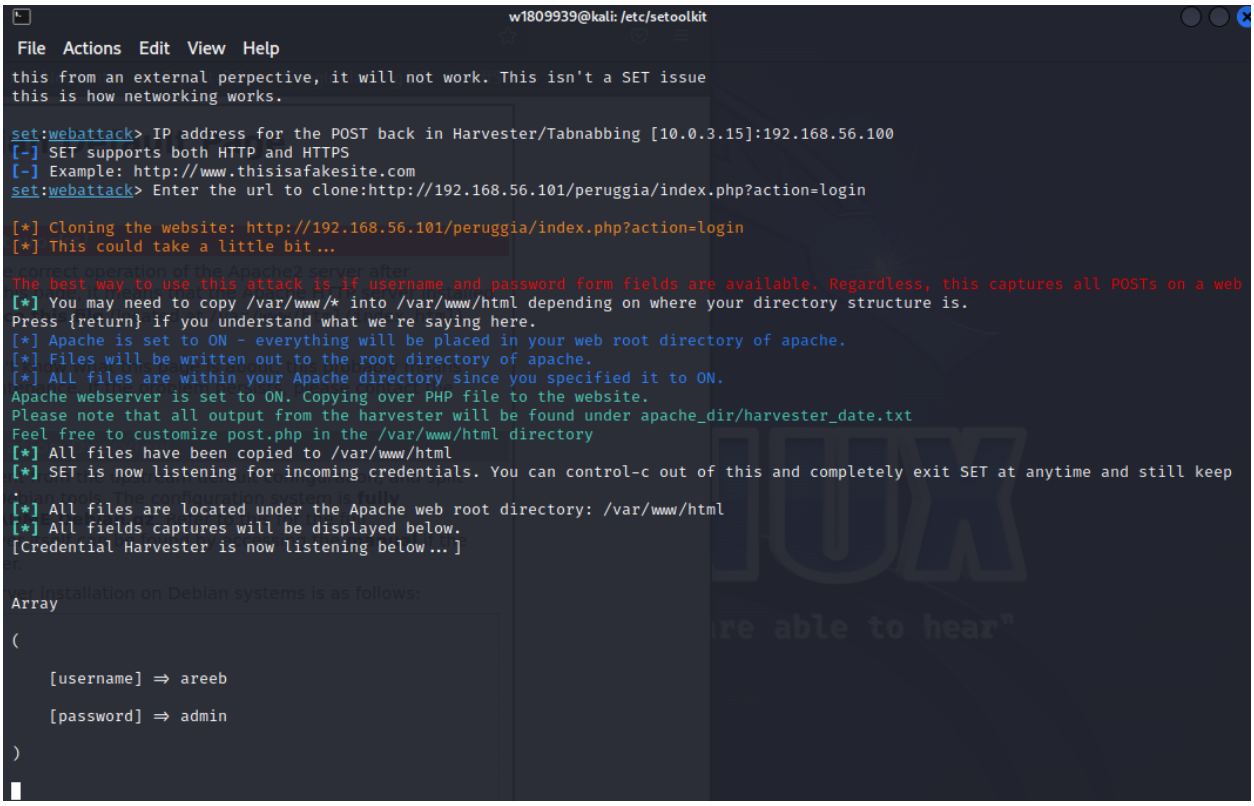

Figure 43: Captured transmission

Here, an attacker listening to the transmission on the cloned site could easily gain the credentials such as the username and password of a user without knowledge of the user.

C.2.2 Scenario Assessment

An attacker could lead a patient or customer of the clinic to a cloned website and gain unauthorized access via the captured credentials. Also, social engineering attacks allow the distribution of creating and sending phishing emails and target phishing attacks. Listed below are some examples of information that can be gathered and its threats.

1. Credentials: Login credentials could be listened to and captured leading to unauthorized access and potential manipulation of patient data and also the clinic operations if it's a stolen credential of an employee.
2. Sensitive data: Phishing attacks and tricking users via clone applications could lead to users inputting personal and sensitive data which may be exposed. This would lead to fraud, malicious activity and identity theft.

3. System information: Details regarding the clinic applications infrastructure could be gained and used as a steppingstone to carry out more advanced attacks.
4. Financial Data: Credit card details can be stolen if the attacker is listening to the request while a patient makes a transaction leading to theft and financial loss.

Therefore, social engineering attacks on a clinic's web application using methods such as credential harvester, phishing and payload generation could lead to violating legal and ethical regulations such as HIPAA and lead to overall reputation damage. Measures such as best security practices, encryption, access control and employee training where they are taught to identify and mitigate such attacks should be implemented to avoid such attacks.

PART D – DENIAL OF SERVICE ATTACKS

D.1 DoS Attack Implementation

Below illustration shows the use of Top command which shows all the processes running on the machine prior to the attack.

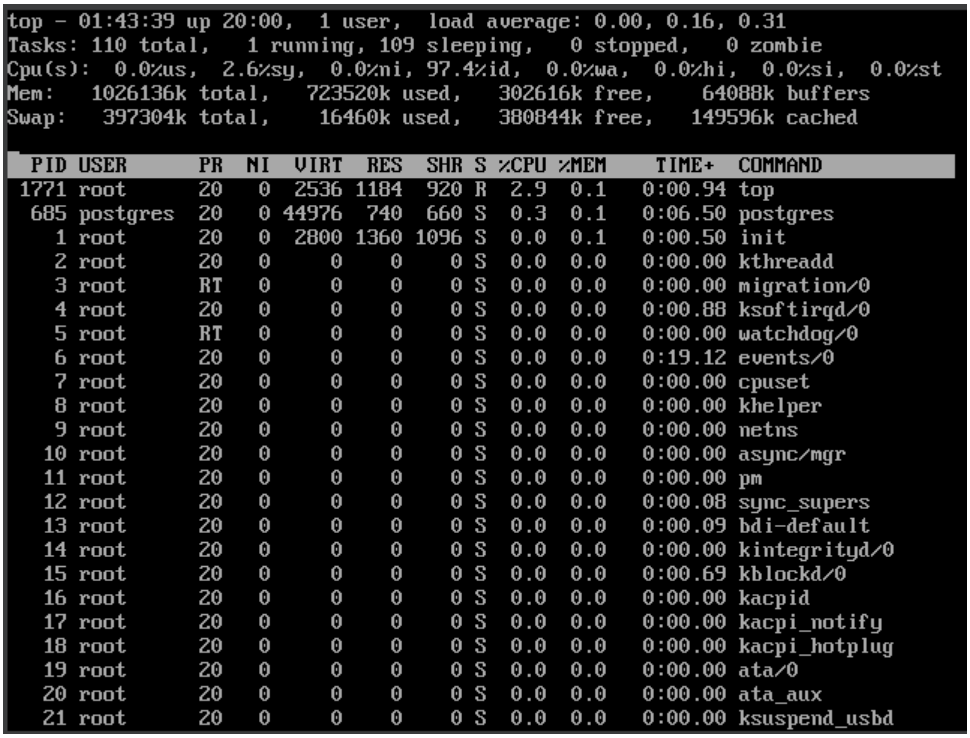

Figure 44: Top prior to attack

The author utilized the TCP SYN flood attack which is one of the most common denial of service attacks in the below illustration.

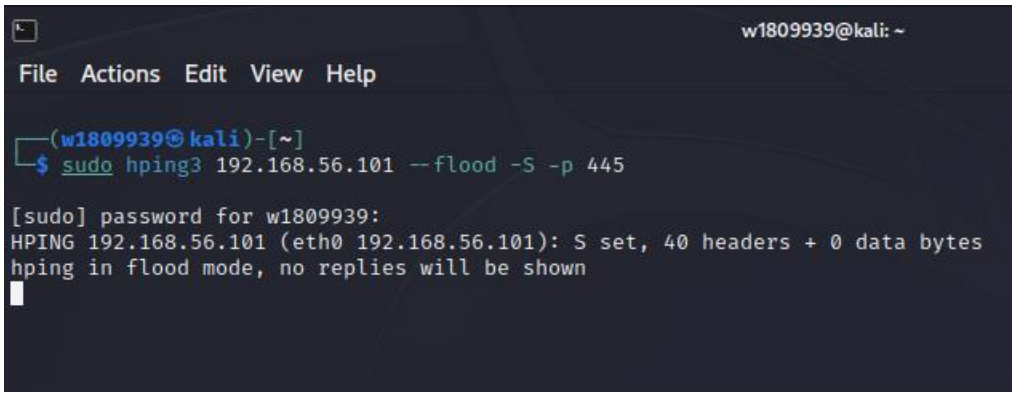

Figure 45: TCP SYN attack

During the attack, when you analyze the processes running on the machine as shown below, you can notice a significant workload increase in the CPU resources in turn making the system slow.

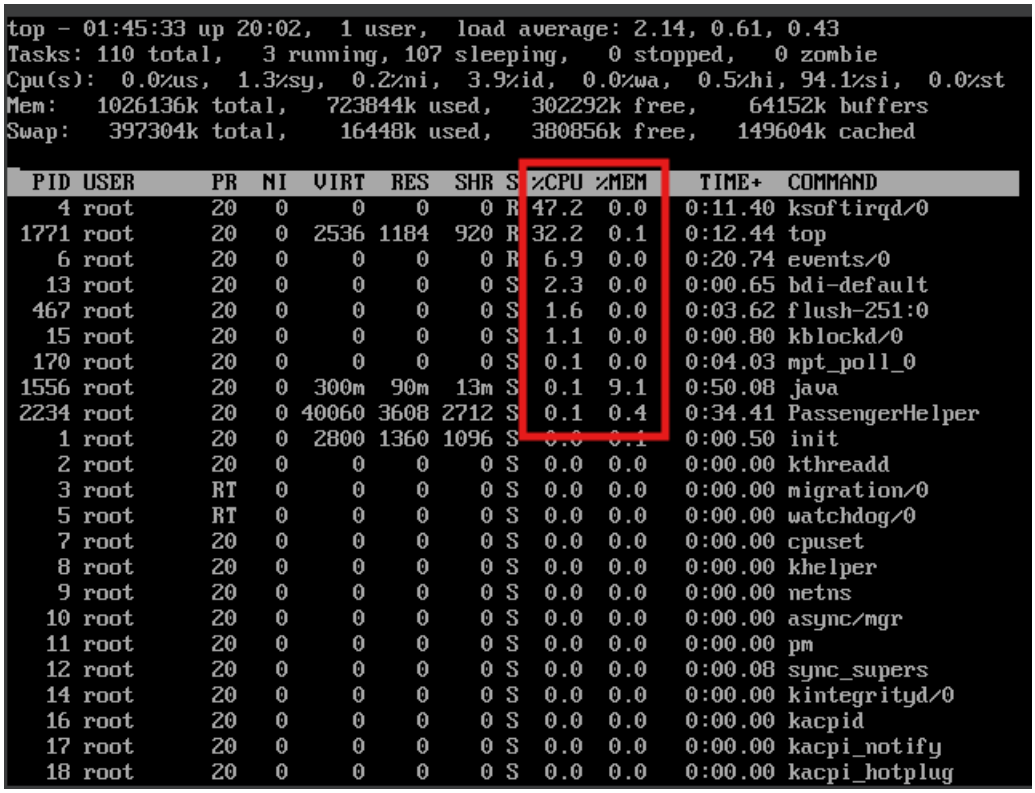

Figure 46: Top during attack

DDoS Ripper

DDoS ripper also can be used to carry out a DDoS attack which is an amplified DoS attack. Below illustration shows the attack and processes prior to the attack.

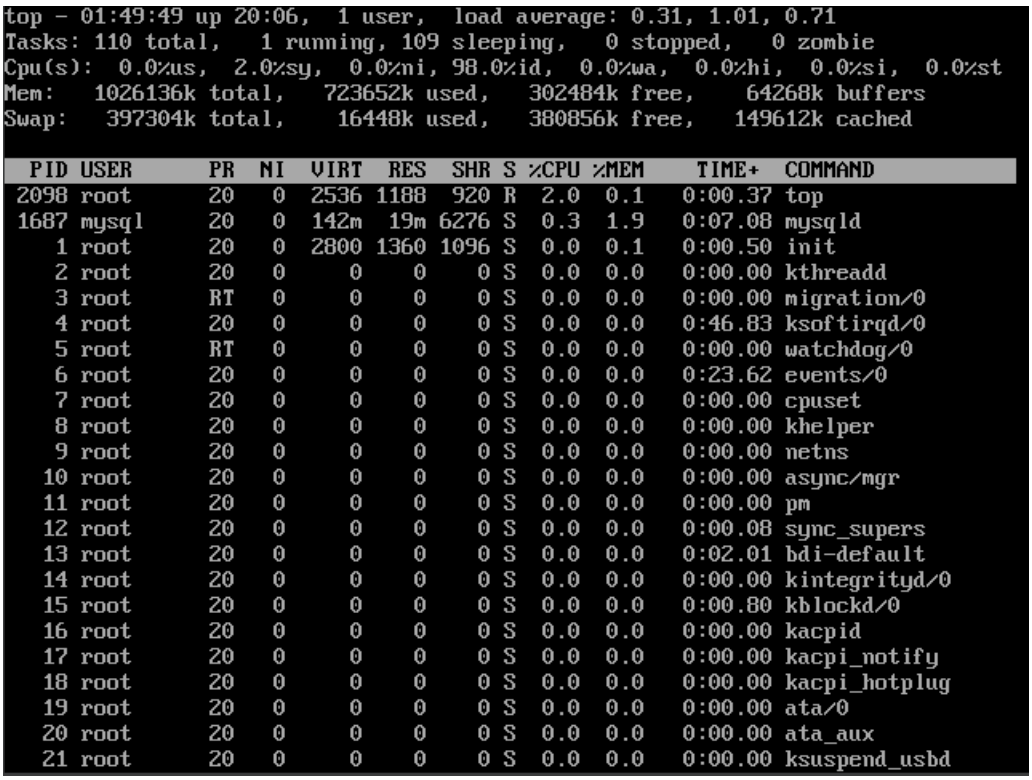

Figure 47: Top prior to DDoS attack

Figure 48: DDoS attack execution

During the attack, you can notice an overwhelming usage but this time across multiple processes rather than a few compared to the previous attack. This would also make the system slow and affect the availability.

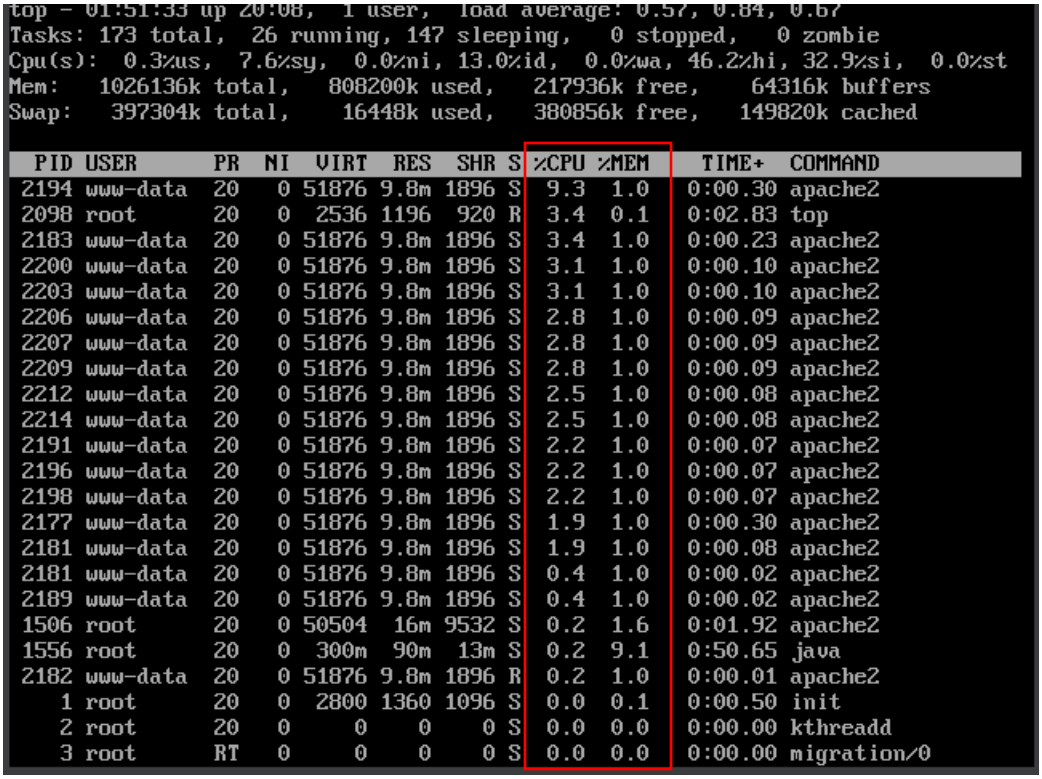

Figure 49: Top during DDoS attack

D.2 Violation of Cyber Security Tenet

A DoS attack violates the cyber security tenet of **availability** as it disrupts and overloads the CPU and memory usage and limits the amount of power and memory for important tasks.

D.3 Scenario Assessment

A DoS attack on the clinic's web application could lead to service disruptions and unavailability. A patient requiring to view lab reports online urgently would not be able to do so and this would damage the reputation of the clinic due to inconvenience.

Moreover, patients wanting to book appointments and make transactions online will not be able to do so due to slow performance and the inability to complete transactions. Again, this could lead to a loss of trust and financial loss as customers are lost.

PART E – RECOMMENDATIONS TO PROTECT THE SCENARIO COMPANY SERVER

E.1 Recommendations to Minimize Threats in Reconnaissance Phase

Threats found in section A.2 were registrar, domain and name server details from WHOIS. ASNs, IPs, name servers and mail server information from TheHarvester and dnsenum. Ways in which it can be minimized are listed below.

1. **Masking Services**: Utilizing masking services to hide sensitive information like hiding true domain name and registrant information to public Whois databases and enabling WHOIS protection services keeps your personal information safe (Borges, 2023).
2. **Network Address Obfuscation**: Utilizing techniques like Network Address Translation (NAT) and IP address rotation to hide the actual network and IP infrastructure of systems make it more difficult for attackers to map out your network infrastructure (Hunt, 2021).
3. **Continuous Monitoring and Alerting**: Using monitoring tools and instant alerting based on suspicion should be implemented to prevent potential exploits (Hunt, 2021).

E.2 Port Knocking and Protection

Port Knocking refers to the attempt to open a port via a specific order of connections made to a blocked port (Paul, 2017). It prevents unauthorized access as users that only know the correct sequence of sending connection requests to specific ports will be allowed access (Pranava, 2020). Some of the open ports identified in the first section include: 22/tcp (SSH), 139/tcp (NetBIOS-SSN), 445/tcp (Microsoft-DS) and etc.

Port knocking protects these open port threats by keeping ports closed by default and adds an authentication layer before granting access to SSH services (Paul, 2017). It also filters out any automation attacks and its ability to integrate with other security methodologies make it difficult for attackers to discover open ports and find entry point to attack a system (Nur et al., 2023).

E.3 SQL Injection Preventions

SQL Injection can be prevented through several key methods as outlined below:

1. **Validate Database Inputs**: Filter and sanitize user inputs to prevent malicious code from being injected into database queries (Kime, 2023).
2. **Restrict Database Code**: Limit the code available to the database by reducing functionality, using stored procedures, whitelisting user inputs, and enforcing prepared statements and parameterization (Kime, 2023).
3. **Restrict Database Access**: Integrating access control to implement limited usage and restricting external access through firewalls will prevent unauthorized access to databases (Ma et al., 2019).
4. **Regular Monitoring and Patching**: Regularly monitor communications and update DBMS regularly with latest patches (Ma et al., 2019).
5. **Adopt Least Privilege Principle**: Access control to perform only the bare minimum in your website by adhering to least privilege would mitigate SQL injection threats (Sundar, 2024).

Database and web application owners can significantly reduce the threat of SQL Injection attacks by using these methods or a combination of these methods and strengthen their database defense.

E.4 Cross Sie Scripting Injection Preventions

XSS injection can be prevented through several key methods as outlined below:

1. **Input Validation and Filtering**: Using input validation to make sure only expected data formats are inputted and filtering out and removing any dangerous characters prevents an attacker from writing harmful scripts (Timonera, 2023).
2. **Output Encoding**: Encoding all output data prior displaying, prevents browsers from misinterpreting and executing scripts and mitigate XSS attacks (Lutkevich, 2021).
3. **Implementing Policies**: Integrating content security polices specifying what is allowed to be loaded and executed would restrict external scripts (Grigorov, 2024).
4. **Regular Scanning and Updates**: Thoroughly scanning and testing web applications post code pushes help identify vulnerabilities at an early stage. Keeping web servers and plugins up to date prevents exploitation from vulnerable components (Timonera, 2023).

Implementing these measures or a combination of these methods would significantly reduce the risk of XSS attacks.

E.5 Prevention of Cryptanalysis Attacks

Prevention of cryptanalysis attacks through several ways discussed below.

- Use the latest, most secure encryption algorithms and protocols recommended by experts and standards – do not use old or default ones (Echelon, 2024).
- Create strong unique encryption keys that are longer in length and harder to crack (Echelon, 2024).
- Ensure the correct implementation of cryptographic systems. Configure cryptographic solutions correctly within an overall security infrastructure integration framework (Prajapat and Singh, 2015).
- Educate staff about cryptography attacks. Make employees aware of different types of cryptanalysis attack methods and how they can be prevented.

Organizations can protect their sensitive data and communications from successful cryptanalysis attacks by following these measures and against password attacks exploited in section B.5.

E.6 Methods to Mitigate Man in the Middle Attacks for a Security Analyst

How a security analyst can protect or minimize Man in the Middle Attacks through several key methods is outlined below.

To detect potential vulnerabilities that are susceptible to MitM attacks, a security analyst can apply and configure comprehensive network architecture and settings (Joshi, Das and Saha, 2009). They have to give precedence to setting up strong encryption systems and seeing to it that all endpoints on the network as well as communication channels have correct certificate validation processes (Conti, Dragoni and Lesyk, 2016). Moreover, an expert may create secure PKI for efficient control over digital certificates (Topping, 2023).

Additionally, the security analyst has a duty to suggest use of secure communication protocols such as TLS/SSL, IPsec, S/MIME among others (Javeed et al., 2020). They also need advocate for regular updates on software programs as well as patch management practices which help in mitigating known bugs (Muncaster, 2023). Another undertaking that can be done by this

professional includes frequent audits carried out within networks' security system. Therefore, these are some of the ways where security analyst can protect or mitigate against man-in-the-middle attacks through implementation of these measures.

E.7 Social Engineering Attacks Protection for Companies

How companies can ensure their users do not fall victim to social engineering attacks are discussed through several key methods outlined below.

1. **Educating and Training Employees**: Teaching common social engineering attacks such as phishing emails and prevention measures to company employees would help identify attacks at an early stage and mitigate them (Arun, 2021).
2. **Multi-Factor Authentication**: Company employees could use more than just passwords like 2FA to add an extra layer of security making it difficult for attackers (Chinnasamy, 2020).
3. **Regular Patches and Security Updates**: The company should make sure their web servers and applications are up to date so that attackers find it difficult to gain access (Chinnasamy, 2020).

Companies can protect their sensitive data and communications from successful social engineering attacks and interception by following these measures.

E.8 Protection against DoS Attacks for Companies

Companies usually protect their web services against DoS attacks by utilizing several key methods discussed below.

1. Integrated Firewalls: Companies nowadays also have integrated firewalls which detect malicious activity and filter out and block any suspicious network traffic
2. Updated Tech Stack: Most companies now use the latest technologies for their web services which have in-built features to tackle DoS attacks. Therefore, using and regularly updating the technologies and libraries mitigates the risk of DoS attacks.
3. IPS and Load balancers: It is another common measure for companies to use intrusion preventions systems (IPS) to identify and drop malicious packets. Also, load balancers to distribute traffic across multiple servers so that one server doesn't crash.

Therefore, these are some of the methods a company utilizes to protect their web services against DoS attacks and is advised to follow if it is not implemented already.

E.9 Intrusion Detection and Prevention Systems

E.9.1 Differences between IDS and IPS

	IDS	IPS
Types (Ledesma, 2022)	• NIDS – network based IDS • HIDS – host based IDS	• WIPS – wireless based IPS • NIDS – host based IPS • NIDS – network based IPS
Function (Swanagan, 2019)	A monitoring and detection system that alerts administrators upon detection of suspicious or malicious activity on the network	On the other hand, not only detects potential threats but also autonomously takes actions to prevent or mitigate them.
Response (Ledesma, 2022)	Based on potential threats, only alerts team or administrators.	Takes action and responds to threat by preventing it from succeeding.
Placement Configuration (Tunggal, 2023)	Placed out of band, monitoring a copy of the network traffic.	Placed inline on the network in direct to the traffic flow.
Self Sufficiency (Ashoor and Gore, 2011)	Passive monitoring system.	Active system that can respond and prevent threats autonomously.

E.9.2 Scenario Assessment

An IPS is recommended for a medium-sized medical clinic as the system is critical and must be available at all times. Some of the key reasons why an IPS is recommended are discussed below.

Mainly because of the fact that it prevents threats in **real-time**, **ensures compliance** with such regulatory frameworks as HIPAA and **continuously monitors security**. IPS actively blocks

potential threats like malware and SQL injections which are necessary to **protect sensitive patient data**. It also protects against zero-day attacks by detecting and preventing suspicious activities before vulnerabilities can be exploited. Furthermore, it assists in network segmentation, thereby **increasing security through access control** between different parts of the network.

IPS guarantees patient privacy, data integrity as well as regulatory compliance which are important for maintaining trust and reputation. In short, it provides active, real time proactive protection against a wider range of cyber attacks making it vital for web applications in a medical scenario.

References

Arun, A. (2021). 6 Ways To Prevent Social Engineering Attacks. Available from https://www.stickmancyber.com/cybersecurity-blog/6-ways-to-prevent-social-engineering-attacks [Accessed 6 May 2024].

Ashoor, A.S. and Gore, S. (2011). Difference between Intrusion Detection System (IDS) and Intrusion Prevention System (IPS). In: Wyld, D.C. Wozniak, M. Chaki, N. et al. (eds.). *Advances in Network Security and Applications*. 2011. Berlin, Heidelberg: Springer, 497–501. Available from https://doi.org/10.1007/978-3-642-22540-6_48.

Awati, R. (2022). What are parameter tampering cyber attacks? *Security*. Available from https://www.techtarget.com/searchsecurity/definition/parameter-tampering [Accessed 2 May 2024].

Borges, E. (2023). SecurityTrails | Domain Security & Solutions — Part 4: How to Protect your Account at your Registrar. Available from https://securitytrails.com/blog/domain-security-part-04-protect-account-registrar [Accessed 6 May 2024].

Chappell, B. (2022). What is an Open Port & What are the Security…. *BeyondTrust*. Available from https://www.beyondtrust.com/blog/entry/what-is-an-open-port-what-are-the-security-implications [Accessed 1 May 2024].

Chinnasamy, V. (2020). Social Engineering Attacks: 10 Ways to Prevent it | Indusface Blog. *Indusface*. Available from https://www.indusface.com/blog/10-ways-businesses-can-prevent-social-engineering-attacks/ [Accessed 6 May 2024].

Choudhary, R.R., Verma, S. and Meena, G. (2021). Detection of SQL Injection attack Using Machine Learning. *2021 IEEE International Conference on Technology, Research, and Innovation for Betterment of Society (TRIBES)*, 1–6. Available from https://doi.org/10.1109/TRIBES52498.2021.9751616.

Cobb, M. (2021). What is a Buffer Overflow? How Do These Types of Attacks Work? *Security*. Available from https://www.techtarget.com/searchsecurity/definition/buffer-overflow [Accessed 6 May 2024].

Conti, M., Dragoni, N. and Lesyk, V. (2016). A Survey of Man In The Middle Attacks. *IEEE Communications Surveys & Tutorials*, 18 (3), 2027–2051. Available from https://doi.org/10.1109/COMST.2016.2548426.

Echelon. (2024). How Physical Security Can Play A Role In Preventing Cryptanalysis Attacks Against A Network | Echelon Protective Services. Available from https://echelonprotectiveservices.com/how-physical-security-can-play-a-role-in-preventing-cryptanalysis-attacks-against-a-network/ [Accessed 6 May 2024].

Einorytė, A. (2024). What are open ports? Risks and security | NordVPN. Available from https://nordvpn.com/blog/what-are-open-ports/ [Accessed 1 May 2024].

Grigorov, P. (2024). The Role of Content Security Policy—Why Is CSP So Important? Available from https://www.telerik.com/blogs/through-digital-landscape-role-content-security-policy-why-csp-important [Accessed 6 May 2024].

Gupta, S. and Gupta, B.B. (2017). Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art. *International Journal of System Assurance Engineering and Management*, 8 (1), 512–530. Available from https://doi.org/10.1007/s13198-015-0376-0.

Hunt, B. (2021). Council Post: To Prevent Cyberattacks, Make Reconnaissance Harder. *Forbes*. Available from https://www.forbes.com/sites/forbestechcouncil/2021/12/02/to-prevent-cyberattacks-make-reconnaissance-harder/ [Accessed 6 May 2024].

Javeed, D. et al. (2020). Man in the Middle Attacks: Analysis, Motivation and Prevention.

Joshi, Y., Das, D. and Saha, S. (2009). Mitigating man in the middle attack over secure sockets layer. *2009 IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA)*. December 2009. 1–5. Available from https://doi.org/10.1109/IMSAA.2009.5439461 [Accessed 6 May 2024].

Kapsamer, R. (2022). The undetectable Cyber Security Threat: Data Tampering. *Tributech*. Available from https://www.tributech.io/blog [Accessed 2 May 2024].

Kime, C. (2023). How to Prevent SQL Injection: 5 Key Prevention Methods. *eSecurity Planet*. Available from https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks/ [Accessed 6 May 2024].

Ledesma, J. (2022). IDS vs. IPS: What Organizations Need to Know. Available from https://www.varonis.com/blog/ids-vs-ips [Accessed 6 May 2024].

Lutkevich, B. (2021). What is Cross-Site Scripting (XSS)? How to Prevent and Fix It. *Security*. Available from https://www.techtarget.com/searchsecurity/definition/cross-site-scripting [Accessed 6 May 2024].

Ma, L. et al. (2019). Research on SQL Injection Attack and Prevention Technology Based on Web. *2019 International Conference on Computer Network, Electronic and Automation (ICCNEA)*. September 2019. 176–179. Available from https://doi.org/10.1109/ICCNEA.2019.00042 [Accessed 6 May 2024].

Mitropoulos, D. et al. (2016). How to Train Your Browser: Preventing XSS Attacks Using Contextual Script Fingerprints. *ACM Transactions on Privacy and Security*, 19 (1), 2:1-2:31. Available from https://doi.org/10.1145/2939374.

Muncaster, P. (2023). What Is a Man-in-the-Middle Attack and How Can I Protect My Organization? *Verizon Business*. Available from

https://www.verizon.com/business/resources/articles/skyword/sw/what-is-a-man-in-the-middle-attack-and-how-can-i-protect-my-organization/ [Accessed 6 May 2024].

Nur, M. et al. (2023). The Effectiveness of the Port Knocking Method in Computer Security. *International Journal of Integrative Sciences*, 2, 873–880. Available from https://doi.org/10.55927/ijis.v2i6.4526.

Paul, M. (2017). What is Port Knocking? *Interserver Tips*. Available from https://www.interserver.net/tips/kb/what-is-port-knocking/ [Accessed 6 May 2024].

Prajapat, S. and Singh, R. (2015). Various Approaches towards Cryptanalysis. *International Journal of Computer Applications*, 127 (14), 15–24. Available from https://doi.org/10.5120/ijca2015906518.

Pranava, K.V. (2020). Port Knocking. *Medium*. Available from https://pranavakumar.medium.com/port-knocking-8d845185a90d [Accessed 6 May 2024].

Pritchard, S. (2020). OSINT: What is open source intelligence and how is it used? *The Daily Swig | Cybersecurity news and views*. Available from https://portswigger.net/daily-swig/osint-what-is-open-source-intelligence-and-how-is-it-used [Accessed 30 April 2024].

Rua, M. et al. (2017). THE IMPACT OF SQL INJECTION ATTACKS ON THE SECURITY OF DATABASES.

Sadotra, P. and Sharma, C. (2017). SQL Injection Impact on Web Server and Their Risk Mitigation Policy Implementation Techniques: An Ultimate solution to Prevent Computer Network from Illegal Intrusion. *International Journal of Advanced Research in Computer Science*. Available from https://www.semanticscholar.org/paper/SQL-Injection-Impact-on-Web-Server-and-Their-Risk-Sadotra-Sharma/57251eb59f6e90093ad656ac6c4952f26cf0b0b5 [Accessed 4 May 2024].

Schrader, D. (2022). Common Open Port Vulnerabilities List. *https://blog.netwrix.com/*. Available from https://blog.netwrix.com/2022/08/04/open-port-vulnerabilities-list/ [Accessed 1 May 2024].

Sundar, V. (2024). How to Prevent SQL Injection Attacks? | Indusface Blog. *Indusface*. Available from https://www.indusface.com/blog/how-to-stop-sql-injection/ [Accessed 6 May 2024].

Swanagan, M. (2019). Intrusion Detection VS Prevention Systems: What's The Difference? *PurpleSec*. Available from https://purplesec.us/intrusion-detection-vs-intrusion-prevention-systems/ [Accessed 6 May 2024].

Timonera, K. (2023). How to Prevent Cross-Site Scripting (XSS) Attacks. *eSecurity Planet*. Available from https://www.esecurityplanet.com/endpoint/prevent-xss-attacks/ [Accessed 6 May 2024].

Titterington, A. (2023). OSINT: what's the danger, and how to stay safe. Available from https://www.kaspersky.com/blog/osint-open-source-intelligence/48911/ [Accessed 30 April 2024].

Topping, S. (2023). Understanding Public Key Infrastructure: Overview and Key Concepts - GlobalSign. Available from https://www.globalsign.com/en/blog/understanding-pki-overview-and-key-concepts [Accessed 6 May 2024].

Tunggal, A. (2023). IDS vs. IPS: What is the Difference? | UpGuard. Available from https://www.upguard.com/blog/ids-vs-ips [Accessed 6 May 2024].

Tunggal, A. (2024). What is an Open Port? Definition & Free Checking Tools | UpGuard. Available from https://www.upguard.com/blog/open-port [Accessed 1 May 2024].

Walkowski, D. (2020). What Is Cross-Site Scripting? *F5 Labs*. Available from https://www.f5.com/labs/learning-center/what-is-cross-site-scripting [Accessed 4 May 2024].

Yasar, K. (2023). What is a SQL Injection? | Definition from TechTarget. *Software Quality*. Available from https://www.techtarget.com/searchsoftwarequality/definition/SQL-injection [Accessed 4 May 2024].