

DVWA: Brute Force Attack Exploitation Lab

By: Belizaire Bassette II

School: Syracuse University

Major: Information Management & Technology (Information Security Concentration)

Minor: Computer Science

Overview & Professional Relevance

This lab was conducted in a controlled environment using the Damn Vulnerable Web Application (DVWA) to simulate a real-world brute force attack scenario. The goal was to gain hands-on experience with the tools, techniques, and workflows commonly used in penetration testing engagements and security operations.

Key outcomes of this exercise include:

- Identifying and exploiting authentication weaknesses.
- Leveraging automated tools for password cracking.
- Understanding attacker workflows for improved defense strategies.

In practice, these skills translate to real-world cybersecurity roles by:

- Strengthening offensive capabilities to identify weaknesses before attackers do.
- Enhancing defensive readiness through recognition of brute force indicators.
- Developing professional reporting skills for both technical and executive audiences.

Lab Environment & Tools Used

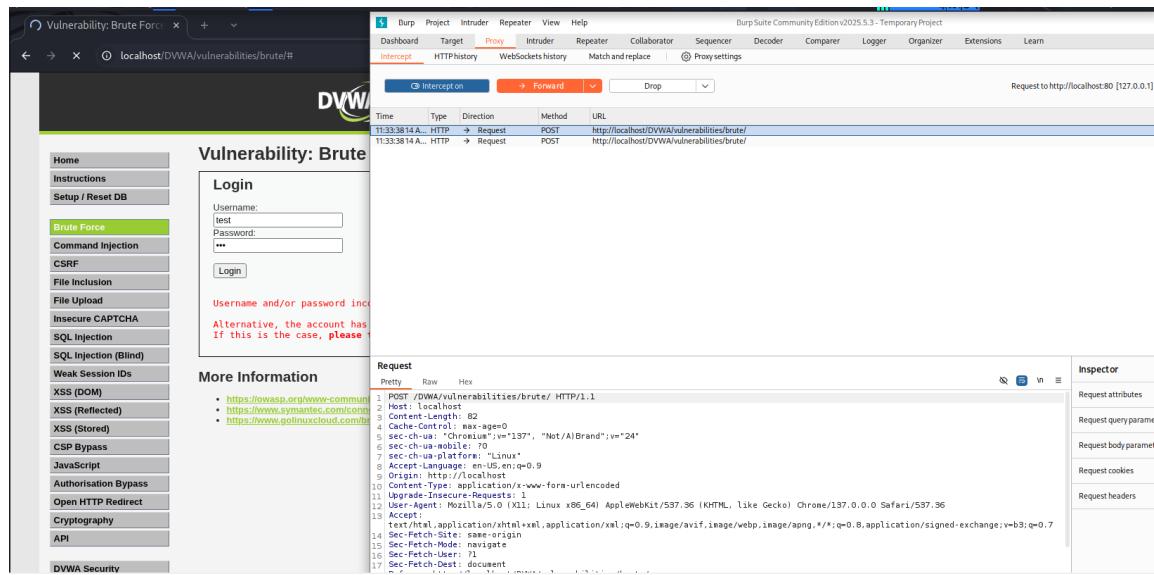
- Operating Systems: Kali Linux, Ubuntu Linux
- Web Application Target: DVWA (Damn Vulnerable Web Application)
- Tools & Utilities:
 - Nmap: Network scanning and service enumeration.
 - Burp Suite: HTTP traffic interception and analysis.
 - Hydra: Automated password cracking for brute force testing.
 - rockyou.txt: Common password wordlist from the 2009 RockYou breach.

Step 1 — Target Identification

Confirmed DVWA was hosted locally at 127.0.0.1 (localhost). Accessed the application and navigated to the Brute Force module.

Step 2 — Manual Reconnaissance

Attempted a manual login to observe the server's response to invalid credentials. Identified the error message: 'Username and/or password incorrect'. This provided the failure condition for Hydra's automated attack.



The screenshot shows the DVWA application running in a browser and Burp Suite running in the background. The DVWA interface displays the 'Brute Force' module. A login attempt is made with 'test' for the username and '***' for the password, resulting in an error message: 'Username and/or password incorrect'. The Burp Suite interface shows the captured POST request for the login attempt. The request details are as follows:

```
POST /DVWA/vulnerabilities/brute/ HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="137", "Not/A)Brand";v="24"
sec-ch-ua-mobile: 70
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Origin: http://localhost
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.96 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.96
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: -1
Sec-Fetch-Dest: document
```

Step 3 — Traffic Interception & Parameter Discovery

Using Burp Suite, intercepted the login request to identify POST parameters and the exact form structure.

Parameters Identified:

- username
- password
- Login (submit field)

The screenshot shows the Burp Suite interface with the following details:

- Proxy settings:** A table showing captured requests with columns: Params, Edited, Status code, Length, MIME type, Extension, Title, Notes, TLS, IP, and Cookies. Most requests are 200 OK with various content types (HTML, PHP, JS) and extensions (php, js, HTML). The IP is 127.0.0.1 and the cookie is PHPSESSID.
- Response:** A detailed view of a captured response. The title is "Welcome :: Damn Vul...". The response content is a Brute Force page with a login form. The code includes HTML, CSS, and a POST form with fields for "username" and "password". It also includes a pre-tag with a user-agent and a password token. The response is rendered in a browser-like view with tabs for "Pretty", "Raw", "Hex", and "Render".
- Inspector:** A sidebar on the right showing the selected text "Username and/or password". It also lists Request attributes, Request body parameters, Request cookies, Request headers, and Response headers.
- Memory:** Shows 142.0MB of memory usage.

Step 4 — Wordlist Preparation

Selected `rockyou.txt` as the password list due to its large dataset of real-world passwords from the 2009 RockYou breach.

```
> wordlists ~ Contains the rockyou wordlist

/usr/share/wordlists
└── amass → /usr/share/amass/wordlists
└── dirb → /usr/share/dirb/wordlists
└── dirbuster → /usr/share/dirbuster/wordlists
└── dnsmap.txt → /usr/share/dnsmap/wordlist_TLAs.txt
└── fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
└── fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists
└── john.lst → /usr/share/john/password.lst
└── legion → /usr/share/legion/wordlists
└── metasploit → /usr/share/metasploit-framework/data/wordlists
└── nmap.lst → /usr/share/nmap/nselib/data/passwords.lst
└── rockyou.txt
└── rockyou.txt.gz
└── sqlmap.txt → /usr/share/sqlmap/data/txt/wordlist.txt
└── wfuzz → /usr/share/wfuzz/wordlist
└── wifite.txt → /usr/share/dict/wordlist-probable.txt
```

Step 5 — Hydra Execution

The following command was executed to perform the brute force attack:

```
hydra -l admin \
    -P /usr/share/wordlists/rockyou.txt \
    127.0.0.1 \
    http-post-form \
```

```
"/DVWA/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:User  
name and/or password incorrect" \  
-V
```

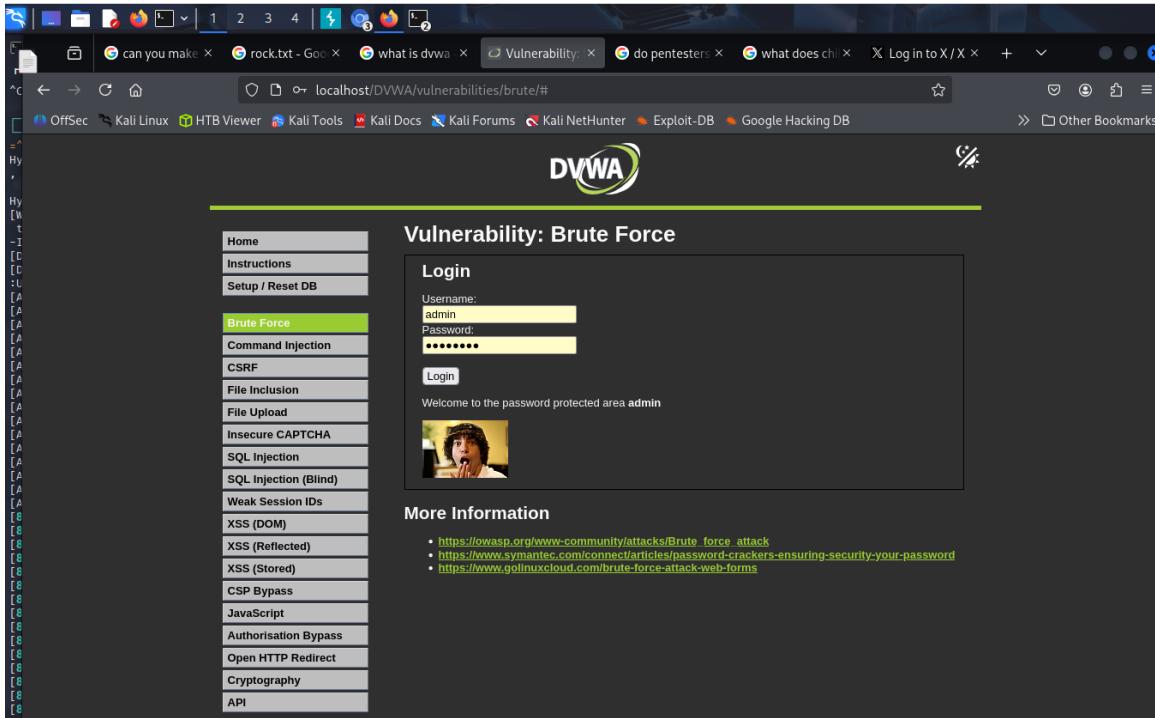
- V

Command Breakdown:

- `-l admin`: Specifies the username to test.
- `-P /usr/share/wordlists/rockyou.txt`: Path to the wordlist.
- ``127.0.0.1``: Target IP (localhost).
- ``http-post-form``: Specifies the attack type (form-based login).
- `"/DVWA/...incorrect"`: Login form path, parameters, and failure message.
- `'-V'`: Verbose mode for detailed output.

Step 6 — Results & Findings

The correct password was found after 4 attempts, successfully logging into the application and capturing the flag.



The screenshot shows a Firefox browser window with multiple tabs open, including 'localhost/DVWA/vulnerabilities/brute/'. The main content is the DVWA Brute Force page. The left sidebar has a 'Brute Force' tab selected. The main area shows a 'Login' form with 'Username: admin' and 'Password:'. Below the form, a message says 'Welcome to the password protected area admin' and shows a small profile picture of a person. At the bottom, there's a 'More Information' section with three links:

- 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>

Step 7 — Security Recommendations

- Implement account lockout policies after a set number of failed attempts.
- Enforce strong password complexity requirements.
- Enable multi-factor authentication (MFA).
- Monitor for repeated failed login attempts using SIEM alerts.