




Writeup On DVWA Bruteforce Attack

Starting up:

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
<input type="checkbox"/>	dvwa d54644005086	 sagikazarmark/dvwa	Running	0.08%	8088:3306 Show all ports (2)	23 minutes ago	 

So we started the docker engine to open up the dvwa.

LOW LEVEL

To start with low level attack we set the level to low first

DVWA Security

Security Level

Security level is currently: **impossible**.

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

1. Low - This security level is complete as an example of how web applications can be used as a platform to teach or learn basic security concepts.
2. Medium - This setting is mainly to give the developer has tried but failed to secure the application using exploitation techniques.
3. High - This option is an extension to the medium level, it uses **practices** to attempt to secure the application, similar in various Capture the Flag (CTF) challenges.
4. Impossible - This level should be set to the highest level of security. Priority to DVWA v1.9, this level was added.

Impossible ▾

Low

Medium

High

Impossible

Submit

So we are going to do the attack

Vulnerability: Brute Force

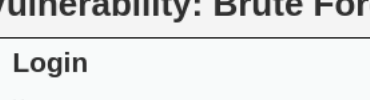
Login

Username:

Password:

Login

Lets take a username called Pablo



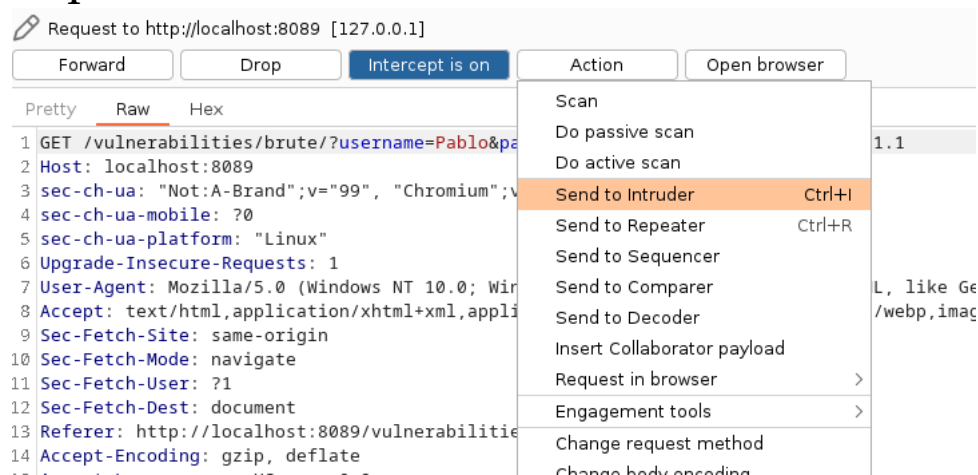
Vulnerability: Brute Force

Login

Username:

Password:

I am entering a random password and intercept the request.



Send this to the intruder

Upload the payload from the seclists and the start attack.

At some positions you can see some change in length that could be the password.

A screenshot of a web application interface. At the top, a large black header bar contains the text 'Vulnerability: Brute Force' in white. Below this, the page has a light gray background. A black rectangular box outlines the main content area. Inside this box, the word 'Login' is displayed in a large, bold, black font. Below 'Login', the text 'Username:' is followed by a white rectangular input field with a thin black border. Underneath the username field, the text 'Password:' is followed by another white rectangular input field with a thin black border. Below the password field, there is a button with a black border and the word 'Login' in black text. At the bottom of the black box, the text 'Welcome to the password protected area Pablo' is displayed. In the bottom-left corner of the entire image, there is a small icon of a document with a green checkmark.

MEDIUM LEVEL



And now we are adjusting the security level to medium and try to attack.

Medium is almost same as low level attack.

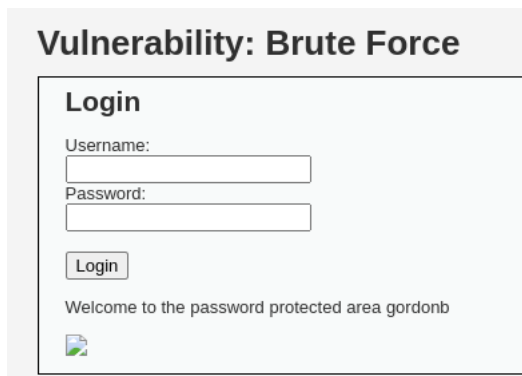
Here Im using a username called gordonb

10	dragon	200	<input type="checkbox"/>	<input type="checkbox"/>	5207
11	123123	200	<input type="checkbox"/>	<input type="checkbox"/>	5207
12	baseball	200	<input type="checkbox"/>	<input type="checkbox"/>	5207
13	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	5265
14	football	200	<input type="checkbox"/>	<input type="checkbox"/>	5207
15	monkey	200	<input type="checkbox"/>	<input type="checkbox"/>	5207
16	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	5207
17	696969	200	<input type="checkbox"/>	<input type="checkbox"/>	5207
18	shadow	200	<input type="checkbox"/>	<input type="checkbox"/>	5207
19	master	200	<input type="checkbox"/>	<input type="checkbox"/>	5207

Request	Response
Pretty	Raw
1 GET /vulnerabilities/brute/?username=gordonb&password=abc123&Login=Login HTTP/1.1	
2 Host: localhost:8089	
3 sec-ch-ua: "Not:A-Brand";v="99", "Chromium";v="112"	
4 sec-ch-ua-mobile: ?0	
5 sec-ch-ua-platform: "Linux"	
6 Upgrade-Insecure-Requests: 1	

Here we can see the password could be abc123

So im going with it



HIGH LEVEL

Here we are going to set the security level to high

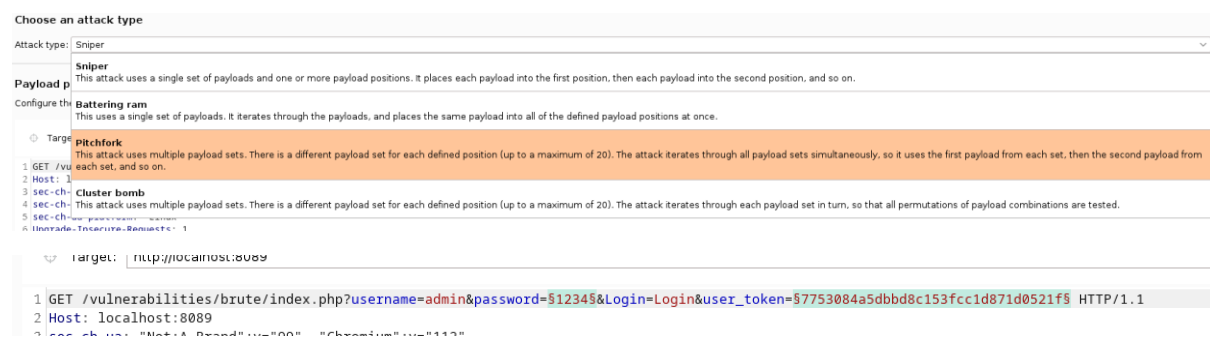
In high level there hidden input called “Tokens”

Which cant be seen by users. The thing is it changes everytime and the server gives a random token. The server checks the username and password and the token to give access to the user. Even if the username and password are correct, the server doesn't allow if the token is incorrect.

```
http?username=admin&password=$12345&Login=Login&user_token=$a6d98695d2de9ea1604e5762cb585964$ HTTP/1.1
```

```
Chromium";v="112"
```

To attack this we are going to use pitchfork attack.



We can see the token changed for each time we refresh the page with same password

You can define one or more payload sets. The number of payload sets depends on the number of payload positions.

Payload set: Payload count: 10,000

Payload type: Request count: 0

Payload settings [Simple list]

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

?

Payload sets

You can define one or more payload sets. The number of payload sets depends on the number of requests.

Payload set:

1

▼

Payload count: unknown

Payload type:

Recursive grep

▼

Request count: 0

?

Payload settings [Recursive grep]

This payload type lets you extract each payload from the response to the payload.

Select the "extract grep" item from which to derive payloads:

▶

Initial payload for first request:

☐ Stop if duplicate payload found

We are using two payloads at a time. So we are using recursive grep and simple list.

Vulnerability: Brute Force

Login

Username:

Password:

Login

Welcome to the password protected area admin

