Informe Laboratorio 2

Sección 1

Kevin Muñoz
kevin.munoz_a@mail.udp.cl

Septiembre de 2023

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) 3
2.4. Identificación de campos a modificar (burp) 5
2.5. Obtención de diccionarios para el ataque (burp) 7
2.6. Obtención de al menos 2 pares (burp) 9
2.7. Obtención de código de inspect element (curl) 14
2.8. Utilización de curl por terminal (curl) 15
2.9. Demuestra 4 diferencias (curl) 16
2.10. Instalación y versión a utilizar (hydra) 17
2.11. Explicación de comando a utilizar (hydra) 18
2.12. Obtención de al menos 2 pares (hydra) 18
2.13. Explicación paquete curl (tráfico) 19
2.14. Explicación paquete burp (tráfico) 19
2.15. Explicación paquete hydra (tráfico) 20
2.16. Mención de las diferencias (tráfico) 21
2.17. Detección de SW (tráfico) 22

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 levantar la aplicacion web se utilizo la imagen de docker con el siguiente comnado:docker run –rm -it -p 80:80 vulnerables/web-dvwa.

```
nejoo@nejoo-System-Product-Name:~$ docker run --rm -it -p 80:80 vulnerables/web-dvwa
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
098cffd43466: Pull complete
b3d64a33242d: Pull complete
Digest: sha256:dae203fe11646a86937bf04db0079adef295f426da68a92b40e3b181f337daa7
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 'Se
rverName' directive globally to suppress this message
. ok
```

Figura 1: Creacion de la aplicacion Web

2.2. Redirección de puertos en docker (dvwa)

continuacion se explicara de manera detallada lo que realiza el comando para la creacion de la aplicacion web. El comando docker run crea y ejecuta un contenedor Docker a partir de la imagen vulnerables/web-dvwa, habilita la interacción con la línea de comandos del contenedor, mapea el puerto 80 del sistema anfitrión al puerto 80 del contenedor y –rm eliminará automáticamente el contenedor cuando se detenga. Esto permite ejecutar la aplicación web DVWA en el puerto 80 del contenedor y acceder a ella desde el sistema anfitrión a través del puerto 80.

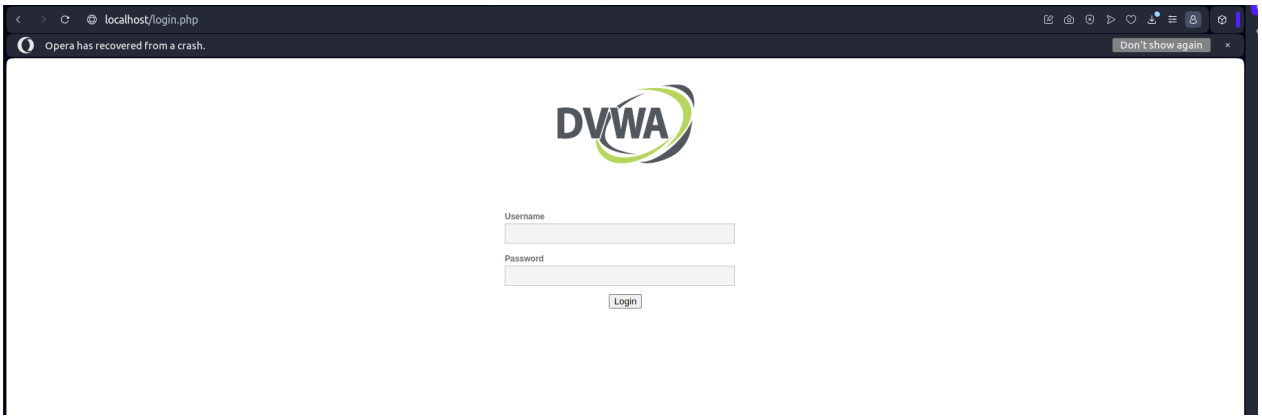


Figura 2: Ejecucion de la aplicacion Web

2.3. Obtención de consulta a replicar (burp)

Una vez que hayas instalado el software Burp, para llevar a cabo un ataque de fuerza bruta, sigue estos pasos: primero, abre Burp y selecciona la pestaña "Proxy". A continuación, abre el navegador integrado en Burp y accede a la página web objetivo. Inicia sesión en la aplicación web utilizando las credenciales proporcionadas en el repositorio de GitHub .ªdminz "password". Luego, dirigirse a la sección "vulnerabilities/brutez comenzar con la replicacion.

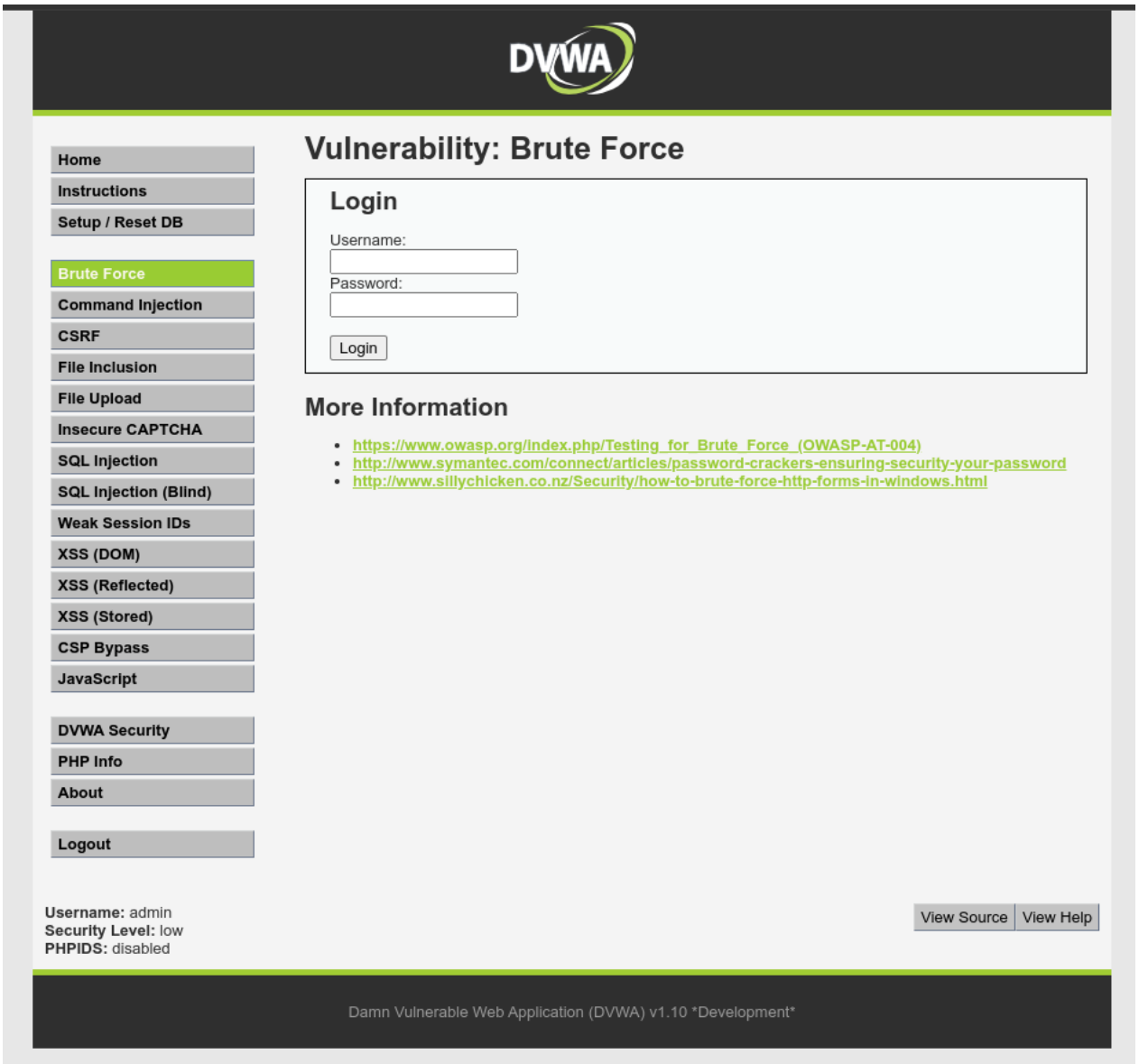


Figura 3: Pagina donde se realizara el ataque por fuerza bruta

Ingresamos credenciales al azar con el fin de capturar los datos HTTP necesarios del formulario de inicio.

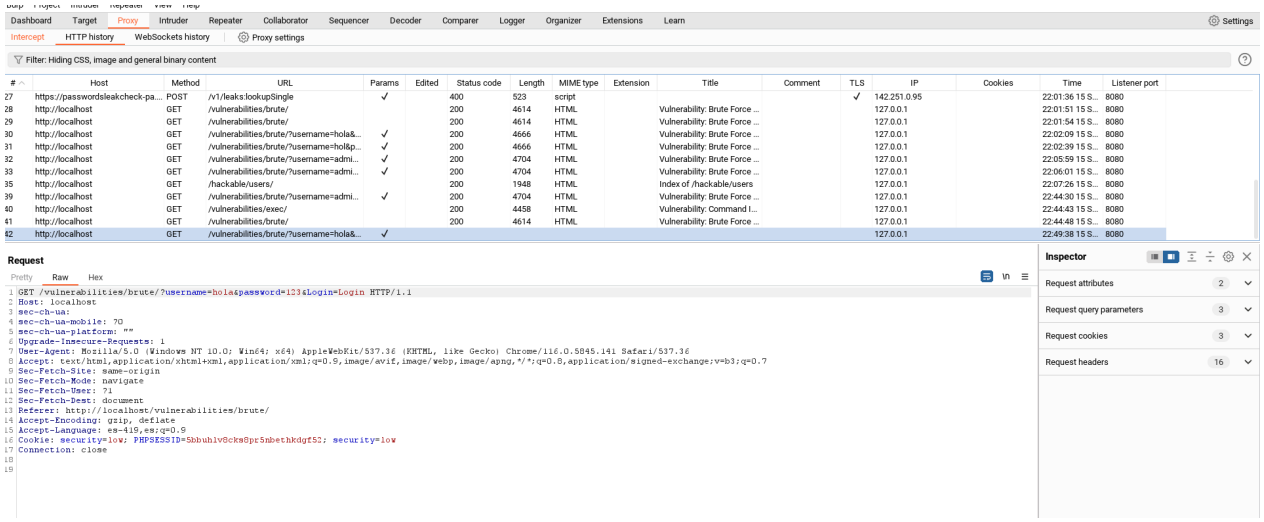


Figura 4: Paquete capturado por burp

Se puede ver como ya tenemos replicada la consulta para realizar su analisi y ver como ejecutar el ataque.

2.4. Identificación de campos a modificar (burp)

Una vez realizado el respectivo analsis de la pagina podemos identificar los campos üsernamez ”password”del formulario.

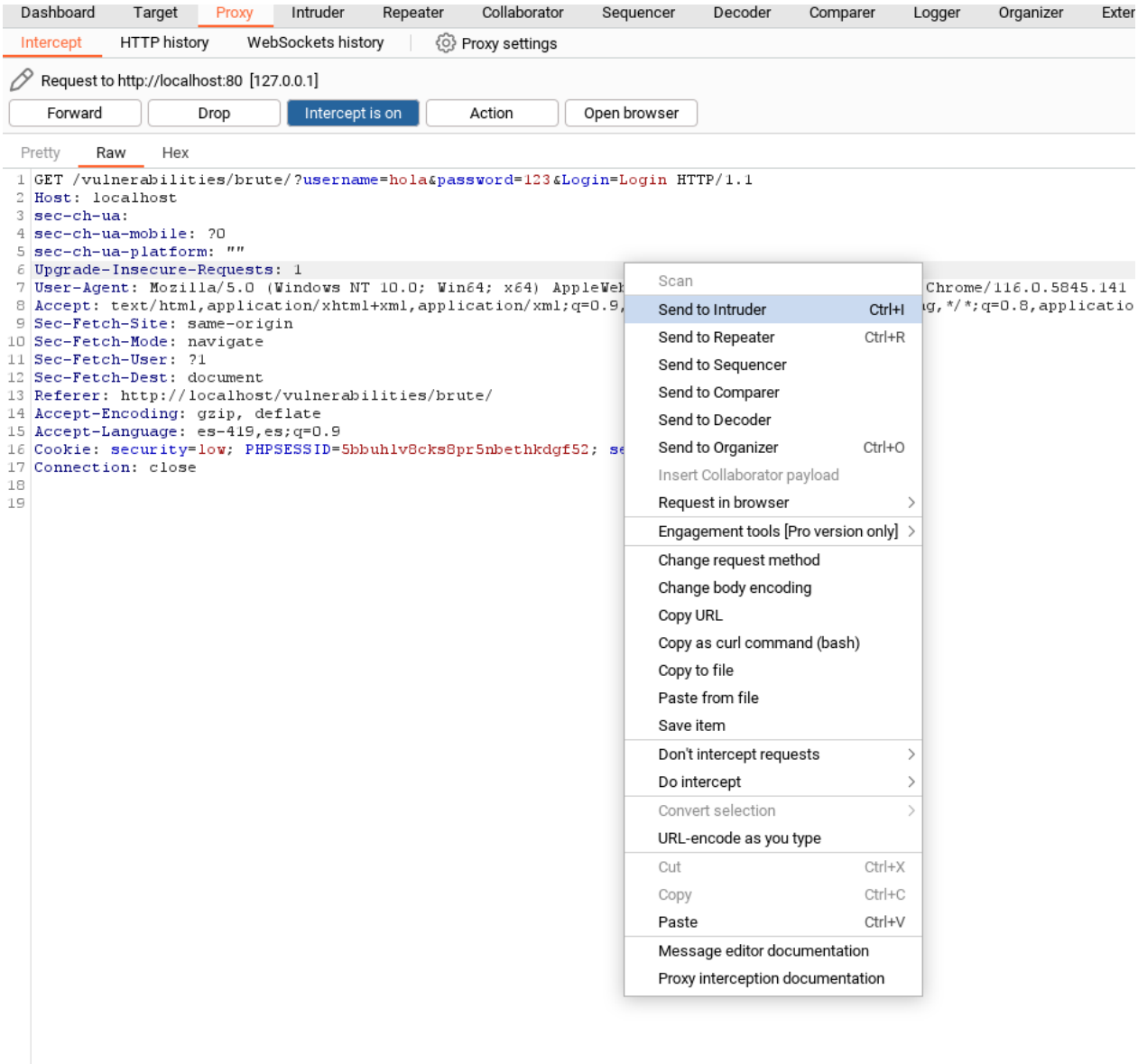


Figura 5: Pagina HTTPS capturada por burp

Una vez que se ha capturado la información del formulario en la página web, esta información es sometida a un análisis detallado y luego se envía al módulo Ïntruder"de una herramienta como Burp Suite. Este proceso se lleva a cabo después de haber seleccionado previamente los campos específicos que se desean completar dentro del formulario. En otras palabras, el análisis y la manipulación de los datos del formulario se realizan de manera precisa y selectiva antes de enviarlos al módulo Intruder para realizar ataques automatizados.

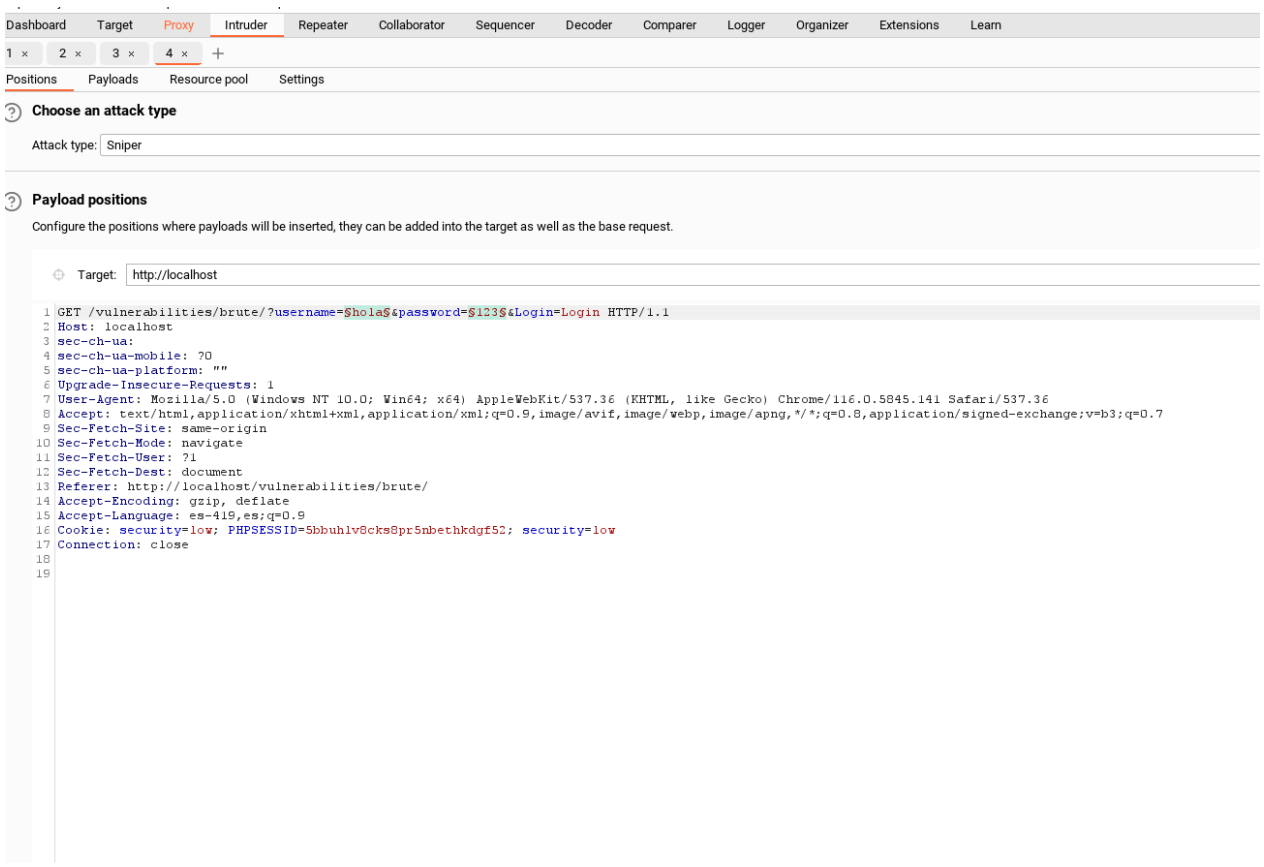


Figura 6: Campos a modificar

2.5. Obtención de diccionarios para el ataque (burp)

Para obtener el diccionario, se realizó una búsqueda en Internet de las combinaciones de usuarios y contraseñas más utilizadas. Estas combinaciones fueron recopiladas a partir de un repositorio en GitHub, donde se habían agregado y consolidado en un archivo de texto (.txt) con el propósito de utilizarlos como parámetros en la herramienta Burp Suite.

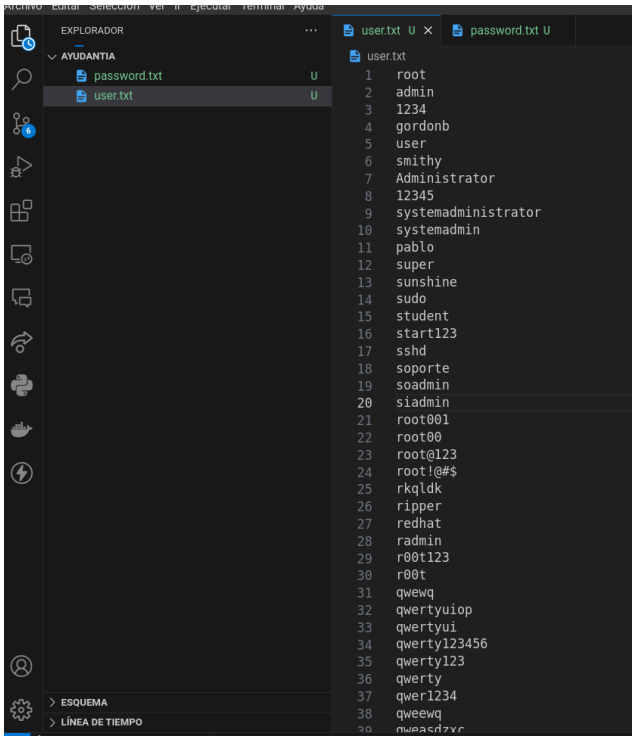


Figura 7: Direccionario de usuario

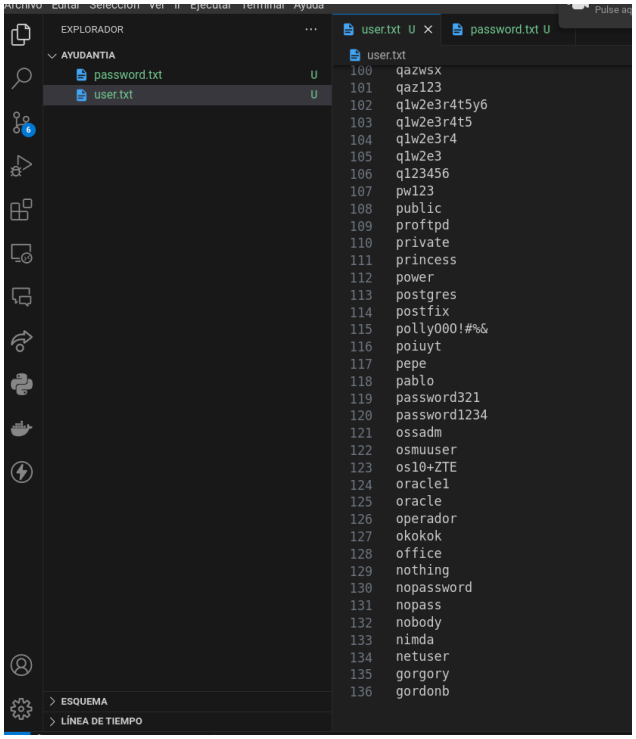


Figura 8: Diccionario de usuario

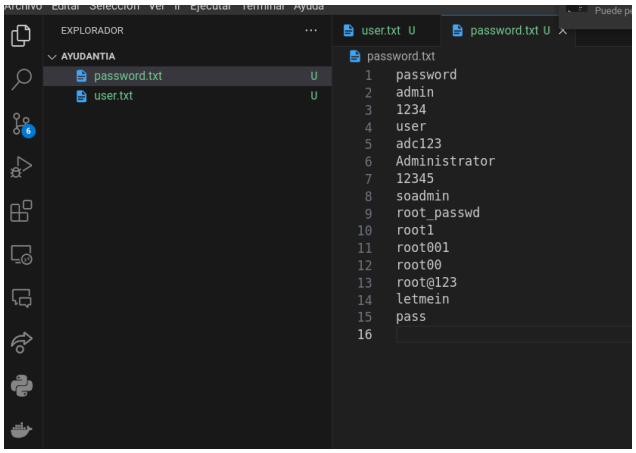


Figura 9: Diccionario de contraseña

2.6. Obtención de al menos 2 pares (burp)

Una vez que los diccionarios han sido cargados con éxito en Burp Suite, como se puede observar en las imágenes a continuación, se procede a iniciar la ejecución del ataque de fuerza bruta. Este proceso implica utilizar las listas de nombres de usuario y contraseñas previamente cargadas para intentar iniciar sesión o acceder a un sistema o servicio objetivo.

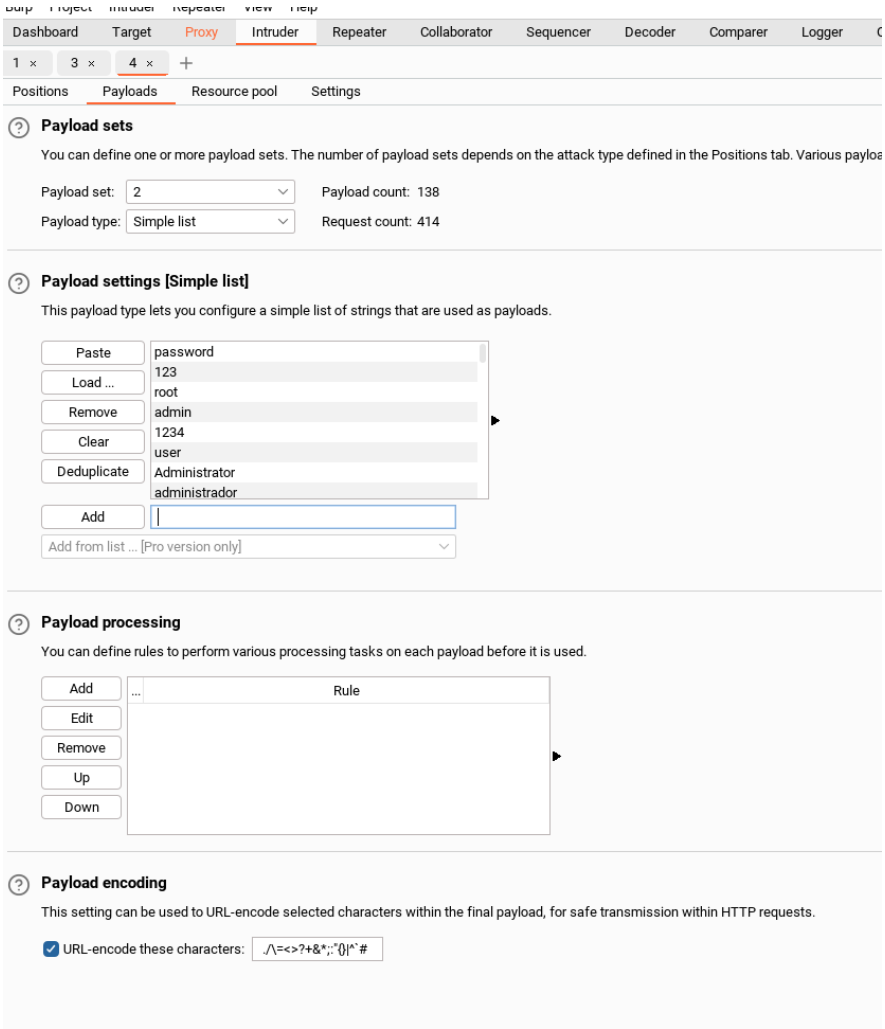


Figura 10: Diccionario de contraseña cargado en Burp Suite

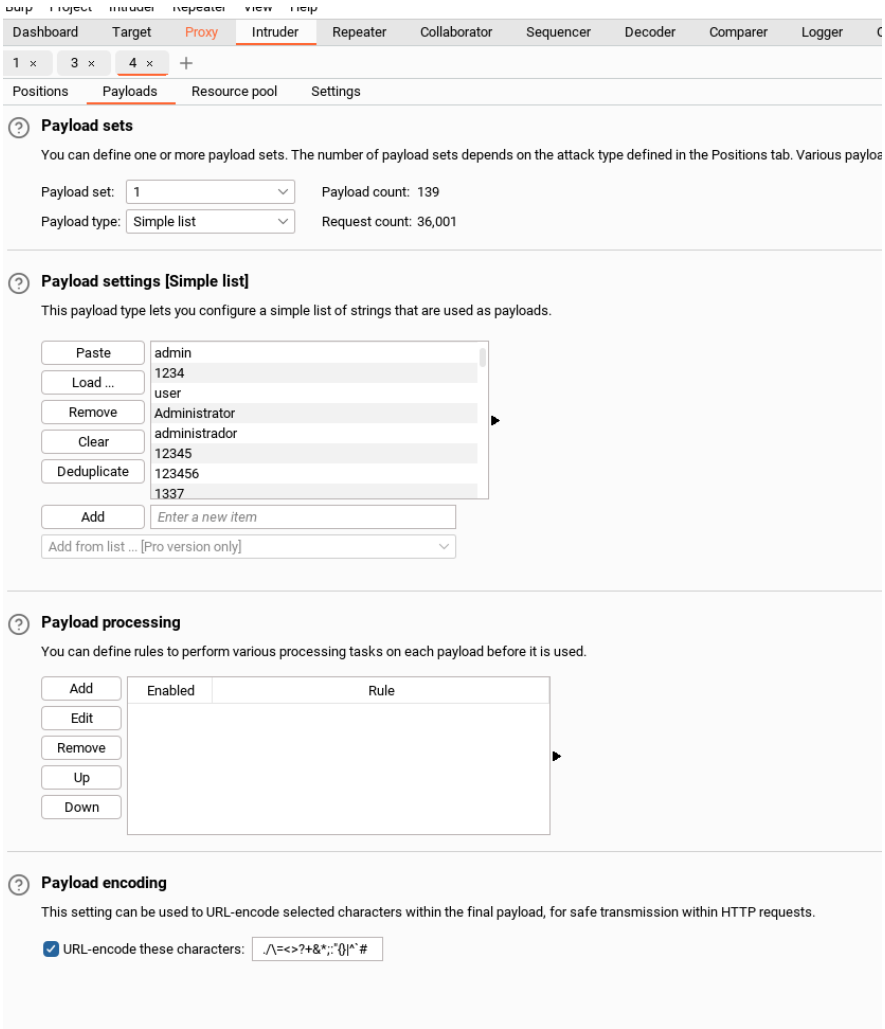
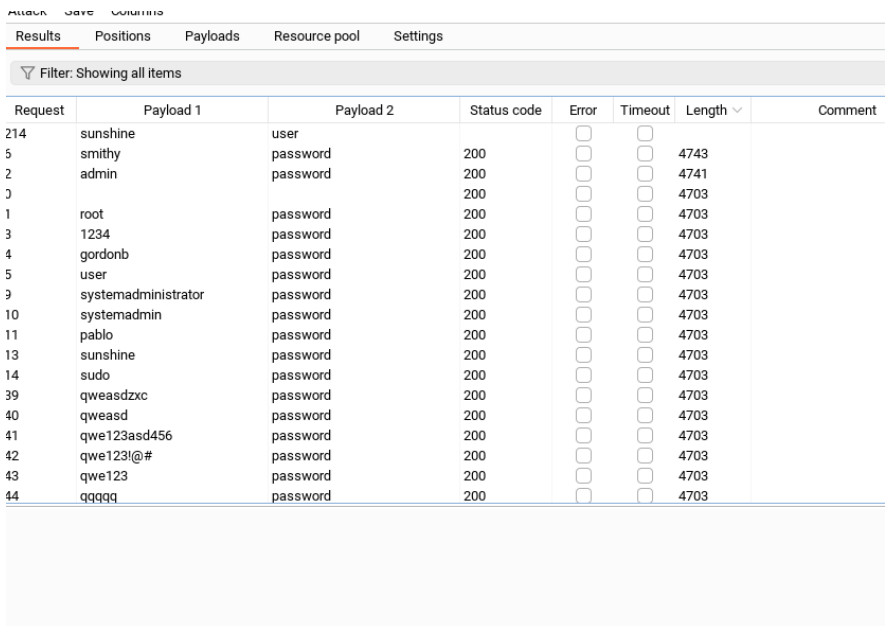


Figura 11: Diccionario de usuario cargado en Burp Suite

El proceso llevó un tiempo considerable, pero finalmente se logró encontrar dos conjuntos de credenciales. Estos conjuntos pudieron ser identificados debido a que su longitud era mayor que la de otras combinaciones y, además, estaban asociados con una imagen de perfil. Esta información adicional proporcionada por la presencia de una foto permitió una identificación más precisa de las cuentas o usuarios correspondientes.



Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
214	sunshine	user					
5	smithy	password	200			4743	
2	admin	password	200			4741	
0			200			4703	
1	root	password	200			4703	
3	1234	password	200			4703	
4	gordonb	password	200			4703	
5	user	password	200			4703	
9	systemadministrator	password	200			4703	
10	systemadmin	password	200			4703	
11	pablo	password	200			4703	
13	sunshine	password	200			4703	
14	sudo	password	200			4703	
39	qweasdzxc	password	200			4703	
40	qweasd	password	200			4703	
41	qwe123asd456	password	200			4703	
42	qwe123!@#	password	200			4703	
43	qwe123	password	200			4703	
44	qqqqq	password	200			4703	

Figura 12: Ataque de fuerza bruta

La credencial 1 se compone de un nombre de usuario smithy y una contraseña ”password”.

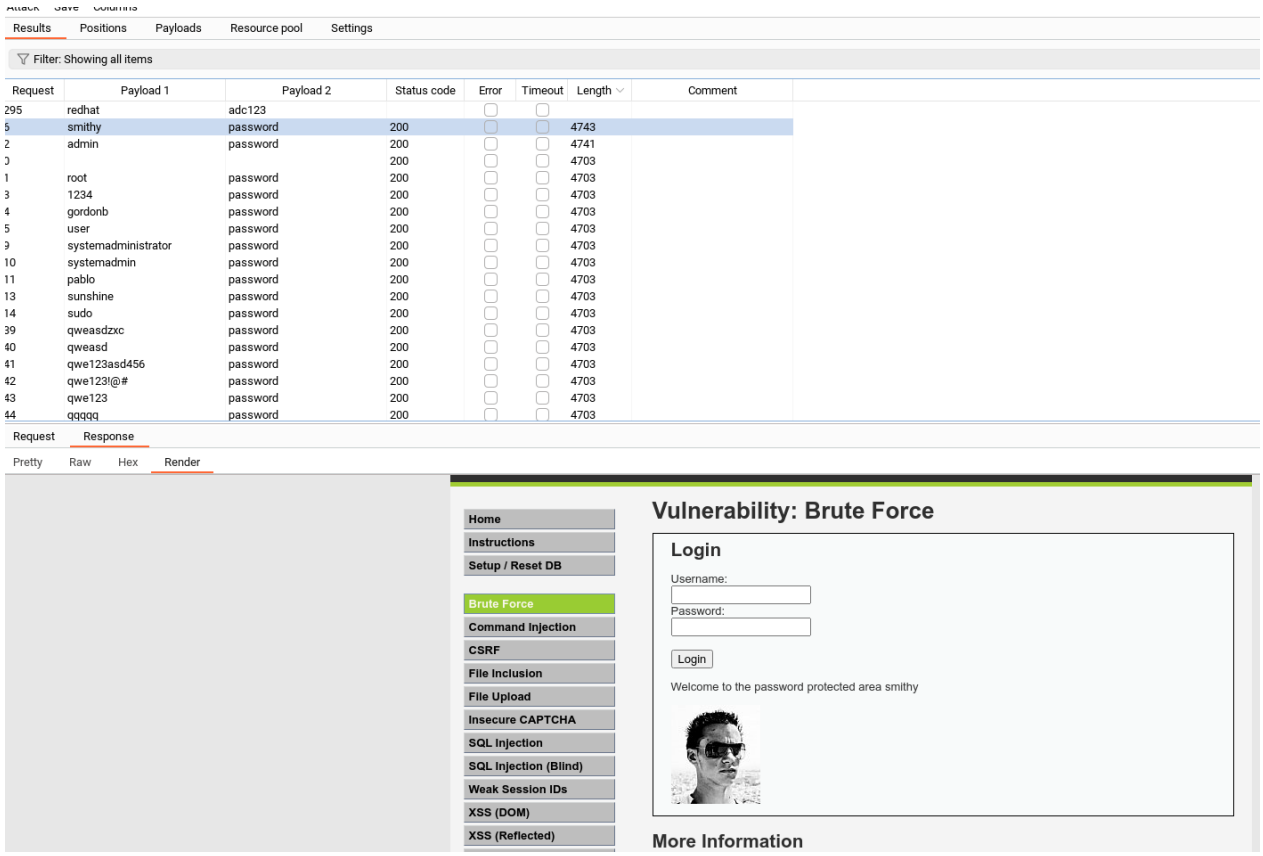


Figura 13: Credencial 1

La credencial 2 se compone de un nombre de usuario admin y una contraseña ”password”.

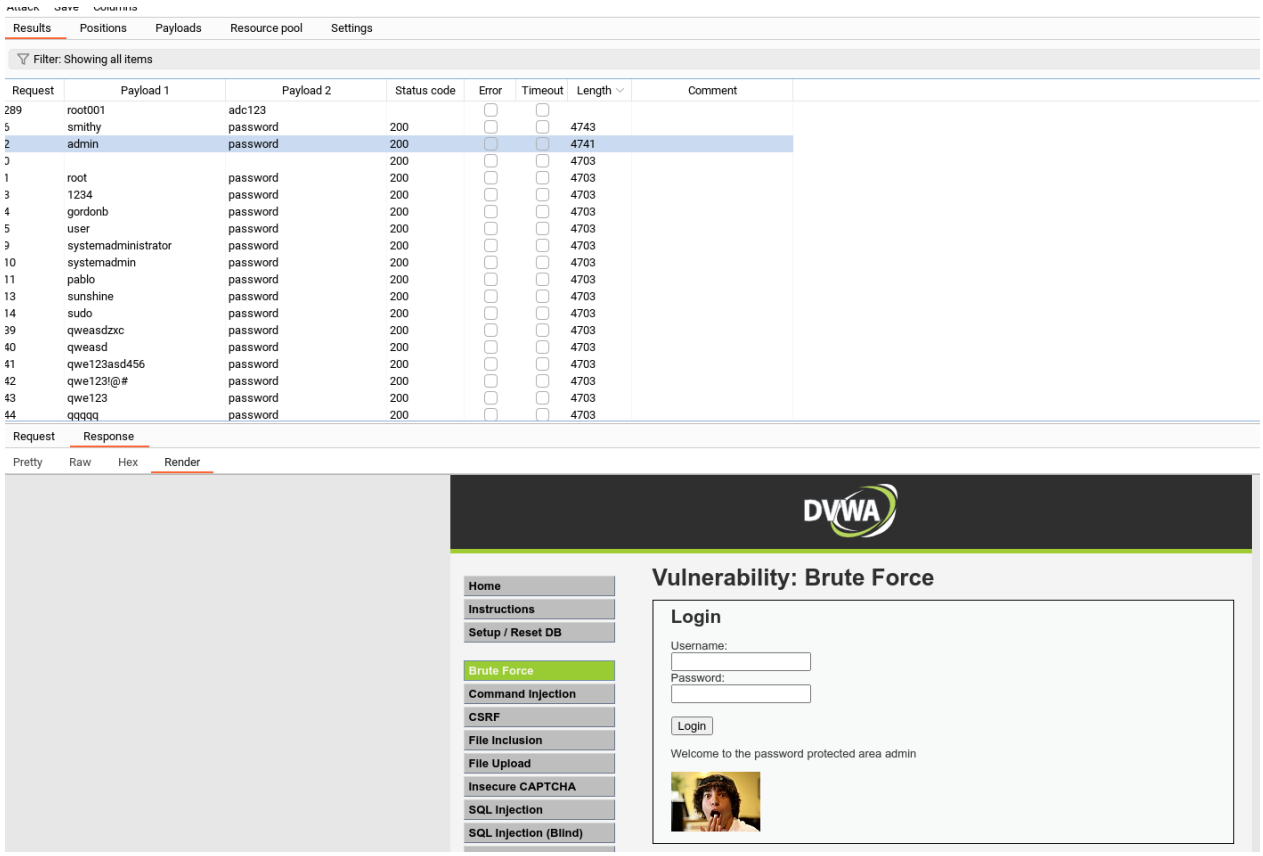


Figura 14: Credencial 2

Es evidente que, aunque ambas respuestas se consideran válidas, difieren en su contenido, lo que resulta en la presentación de imágenes diferentes según el usuario que se ha ingresado. Como regla general, las respuestas válidas tienen una longitud de 4741 para arriba, mientras que el resto de las consultas arrojaron respuestas con una longitud de 4703. No se encontraron más accesos válidos, aparte de estos dos casos.

2.7. Obtención de código de inspect element (curl)

Una vez que se obtuvieron credenciales válidos, se procedió a solicitar el uso de la herramienta çurl"para realizar peticiones específicas. Para llevar a cabo este proceso, fue necesario obtener el código de la solicitud para poder ejecutarlo en la terminal. Para lograr esto, se utilizó el navegador Opera, se accedió a la opción Inspeccionar elementoz se observaron las actividades en la pestaña "Network"de las herramientas de desarrollo.

Se repitió el proceso de realizar una consulta manualmente con el objetivo de crear un registro en el navegador. Posteriormente, se accedió a las herramientas de desarrollador, donde se localizó la solicitud específica y se copió su código en el formato adecuado para su posterior ejecución con la herramienta çurl".

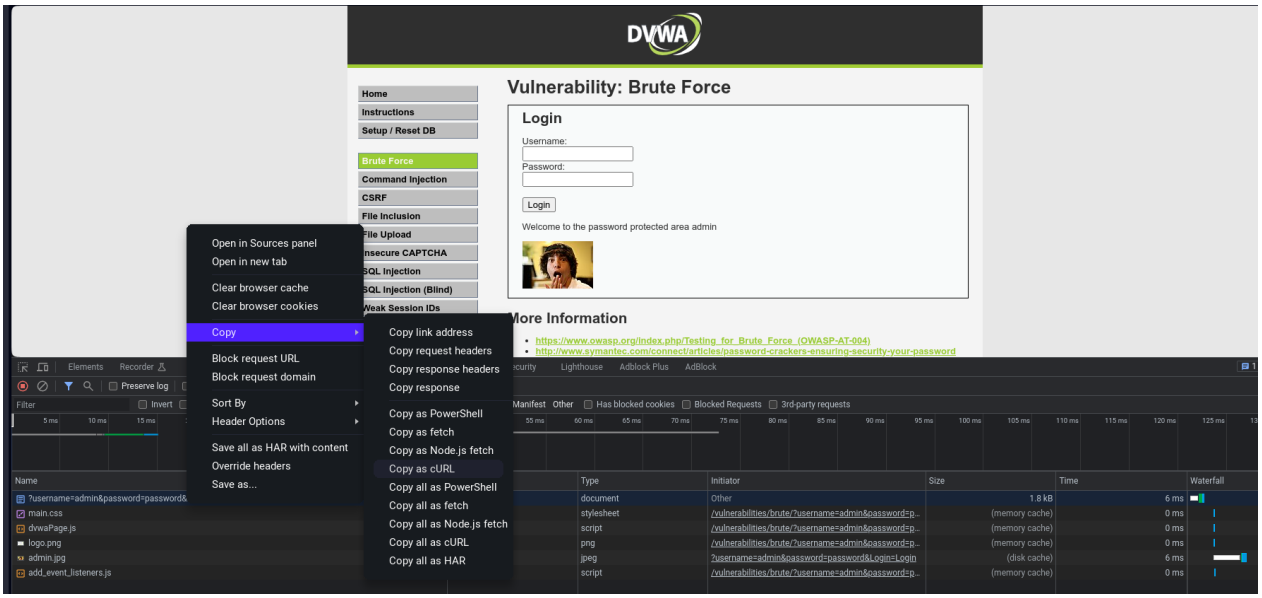


Figura 15: Obtencion de curl

2.8. Utilización de curl por terminal (curl)

Una vez que se obtuvo el comando çurl.ª partir de la sección "Network"de la aplicación web, se procedió a pegarlo o ejecutarlo en la terminal de Ubuntu.

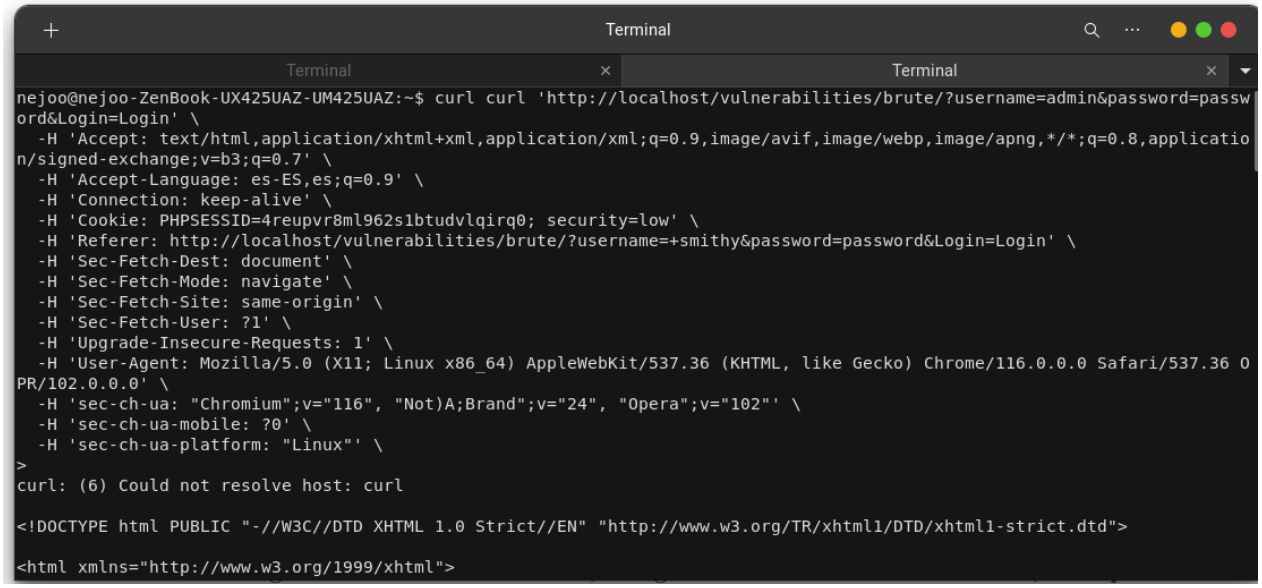


Figura 16: Ejecucion de curl terminal

2.9. Demuestra 4 diferencias (curl)

Una vez ejecutado el comando obtenemos los siguientes resultados.

Credencial valida:

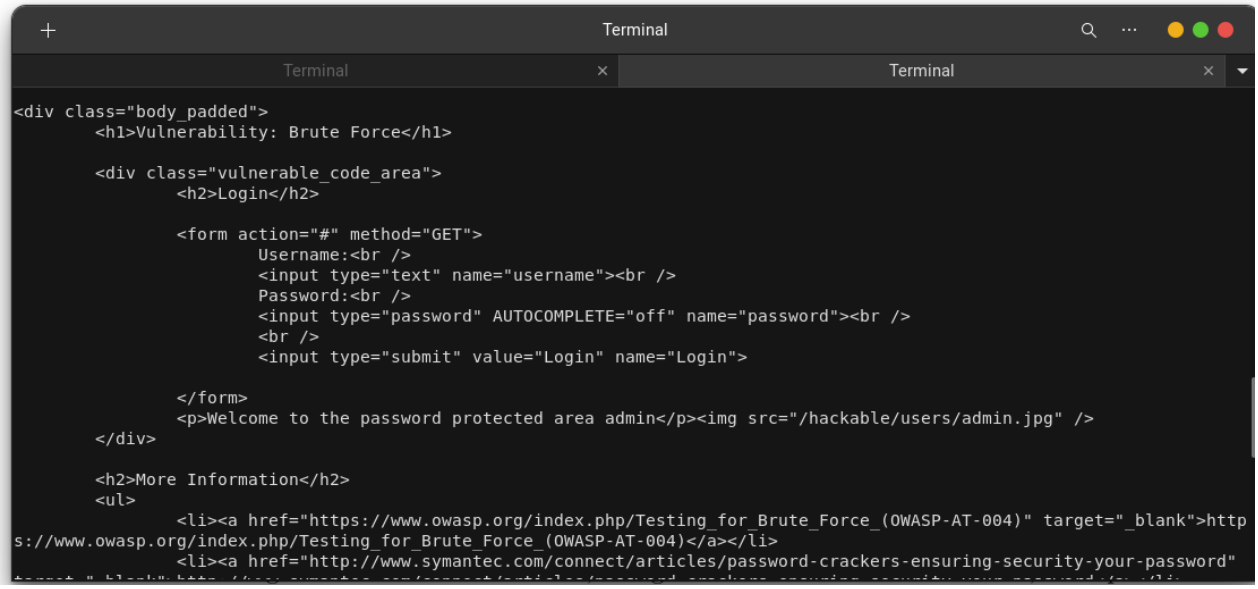


Figura 17: Credencial valida curl

Credencial invalida:

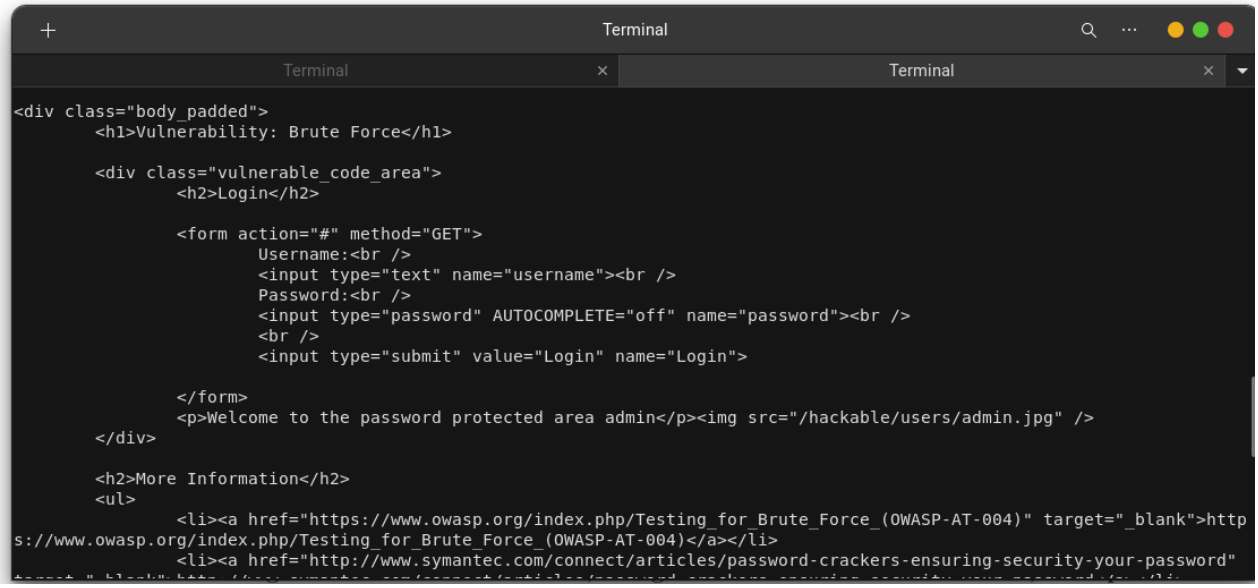


Figura 18: Credencial invalida curl

Se tiene los dos tipos de curl posibles de los cuales se diferencian de la siguiente manera:

- **Contenido del formulario:**En el HTML de la respuesta, se muestra un mensaje adicional debajo del formulario de inicio de sesión: ¡p¿Welcome to the password protected area admin¡/p¿¡img src=/hackable/users/admin.jpg/¿. Esta parte del contenido no está presente en la solicitud CURL.

- **Etiquetas de imagen (img):**En el HTML de la respuesta, se incluye una etiqueta de imagen ¡img src=/hackable/users/admin.jpg/¿. Esto no está presente en la solicitud CURL. La etiqueta de imagen muestra una imagen asociada a la cuenta de administrador.

- **Largo del paquete:**El HTML de la respuesta es más largo y contiene contenido adicional, incluyendo un mensaje de bienvenida y una imagen asociada a la cuenta de administrador, que no están presentes en la solicitud CURL

- **Mensaje de bienvenida:**En el HTML de la respuesta, se muestra un mensaje de bienvenida: ¡p¿Welcome to the password protected area admin¡/p¿. Este mensaje no está presente en la solicitud CURL y el mensaje es diferente al de la CURL invalida.

2.10. Instalación y versión a utilizar (hydra)

Por último, se nos pidió repetir el ataque que habíamos realizado previamente en Burp Suite, pero esta vez usando Hydra. Para lograrlo, primero instalamos el software a través de comandos en la terminal.

```
Leyendo lista de paquetes... Hecho
nejoo@nejoo-ZenBook-UX425UAZ-UM425UAZ:~$ sudo apt-get install hydra
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  firebird3.0-common firebird3.0-common-doc libbson-1.0-0 libfbclient2 libmemcached11 libmongoc-1.0-0 libmongocrypt0
  libmysqlclient21 libserf-1-1 libsvn1 libtommath1 libutf8proc2 mysql-common
Paquetes sugeridos:
  hydra-gtk
Se instalarán los siguientes paquetes NUEVOS:
  firebird3.0-common firebird3.0-common-doc hydra libbson-1.0-0 libfbclient2 libmemcached11 libmongoc-1.0-0
  libmongocrypt0 libmysqlclient21 libserf-1-1 libsvn1 libtommath1 libutf8proc2 mysql-common
0 actualizados, 14 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 4.266 kB de archivos.
Se utilizarán 16,3 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 firebird3.0-common-doc all 3.0.8.33535.ds4-1ubuntu2 [26,8 kB]
Des:2 http://archive.ubuntu.com/ubuntu jammy/universe amd64 firebird3.0-common all 3.0.8.33535.ds4-1ubuntu2 [15,5 kB]
Des:3 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libbson-1.0-0 amd64 1.21.0-1build1 [83,7 kB]
Des:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 libtommath1 amd64 1.2.0-6build3 [56,5 kB]
Des:5 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libfbclient2 amd64 3.0.8.33535.ds4-1ubuntu2 [512 kB]
Des:6 http://archive.ubuntu.com/ubuntu jammy/main amd64 libmemcached11 amd64 1.0.18-4.2ubuntu4 [88,2 kB]
Des:7 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libmongocrypt0 amd64 1.3.0-1ubuntu1 [93,9 kB]
Des:8 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libmongoc-1.0-0 amd64 1.21.0-1build1 [311 kB]
Des:9 http://archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8 [7.212 B]
Des:10 http://archive.ubuntu.com/ubuntu jammy/main amd64 libmysqlclient21 amd64 8.0.28-0ubuntu4 [1.294 kB]
Des:11 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libserf-1-1 amd64 1.3.9-10ubuntu2 [50,0 kB]
Des:12 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libutf8proc2 amd64 2.7.0-3 [73,9 kB]
Des:13 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libsvn1 amd64 1.14.1-3build4 [1.387 kB]
Des:14 http://archive.ubuntu.com/ubuntu jammy/universe amd64 hydra amd64 9.2-1ubuntu1 [266 kB]
```

Figura 19: Instalacion de hydra

2.11. Explicación de comando a utilizar (hydra)

Para llevar a cabo el ataque, fue necesario configurar el comando a utilizar en Hydra. Después de varios intentos, se llegó al siguiente comando:

```
root@nejoo-ZenBook-UX425UAZ-UM425UAZ:/home/nejoo/Escritorio# hydra localhost -s 8080 -L usuarios -P contrasenas http-get-form "/vulnerabilities
/brute/:username=^USER^&password=^PASS^&Login=Login:F=Username and/or password incorrect.:H=Cookie:security=low;PHPSESSID=63mnb3t84ci9f34suhr16
29m27" -I
Hydra v9.2 (c) 2021 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).
```

Figura 20: Comando Hydra

Inicialmente, se especificó la dirección a atacar como ”localhost”. Luego, con la opción s”, se indicó el puerto de operación, que en este caso fue el 8080. Utilizando la opción -L”, se estableció la intención de atacar un listado de usuarios y se proporcionó la ruta al archivo que contenía la lista. De manera similar, se utilizó la opción P”para las contraseñas, proporcionando la ruta al archivo correspondiente.

El parámetro ”http-get-form”se utilizó para definir el tipo de elemento a atacar. A continuación, se incluyó la ruta a la página objetivo, lo que involucró señalar con flechas hacia arriba los campos de usuario y contraseña en la solicitud GET. Esto indicaba que en estos campos se reemplazarían con los elementos de la lista durante el ataque.

También se especificó el mensaje de error esperado en caso de un inicio de sesión incorrecto. Esto permitiría a Hydra identificar los accesos válidos durante el proceso.

Por último, se incluyeron las cookies obtenidas del navegador, que incluían información sobre el nivel de seguridad configurado en DVWA y el ID de la sesión.

2.12. Obtención de al menos 2 pares (hydra)

Al igual que Burp Suite, Hydra es una herramienta potente que se destaca en la prueba de fuerza bruta para descubrir credenciales de acceso. Sin embargo, la distinción principal radica en la velocidad y eficiencia con la que Hydra logra este objetivo. Aunque ambos programas tienen como finalidad descubrir las credenciales, Hydra se distingue por su velocidad para realizar un gran número de intentos en un período de tiempo muy corto, pues lo que tardo hydra no se compara en nada con lo que tardo Burp Suite.

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-09-17 02:40:28
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1005 login tries (l:67/p:15), ~63 tries per task
[DATA] attacking http-get-form://localhost:8080/vulnerabilities/brute/:username=^USER^&password=^PASS^
&Login=Login:F=Username and/or password incorrect.:H=Cookie:security=low;PHPSESSID=acr84ov1mg6njponi8v
8p9lfv6
[8080][http-get-form] host: localhost   login: admin   password: password
[8080][http-get-form] host: localhost   login: smithy   password: password
[8080][http-get-form] host: localhost   login: pablo   password: letmein
1 of 1 target successfully completed, 3 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-17 02:40:42
```

Figura 21: Credenciales capturadas por Hydra

2.13. Explicación paquete curl (tráfico)

Se requería realizar un análisis del tráfico generado por cada programa. Para lograr esto, se utilizó Wireshark para capturar una consulta de cada programa utilizando las credenciales .ªdminz "password".

En el caso de Curl, se obtuvieron los siguientes resultados:

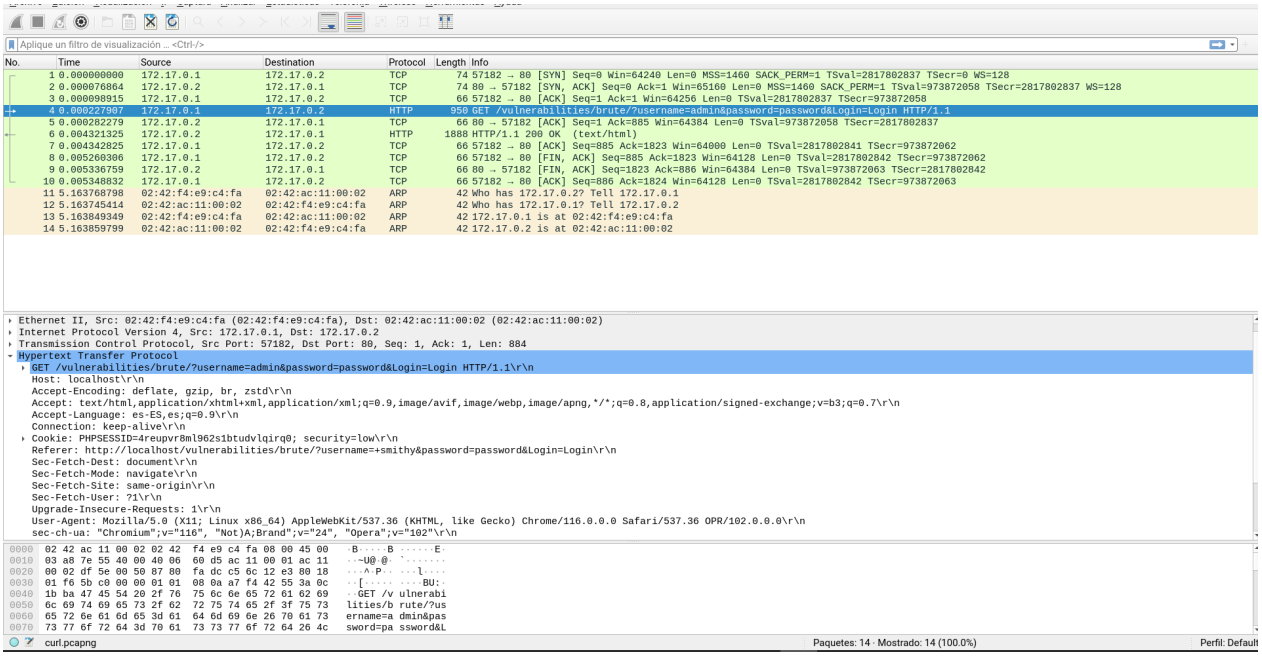


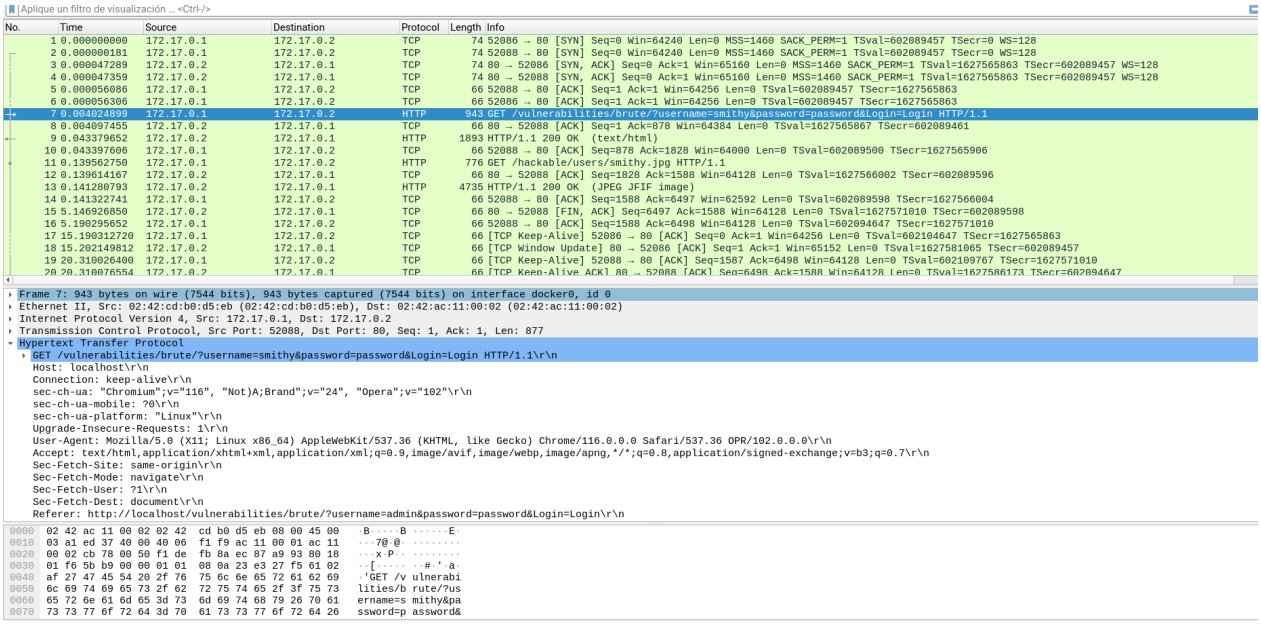
Figura 22: Captura wireshark de curl

Es evidente que, como era de esperar, el paquete de información del método GET es visible en texto plano cuando se observa a través de Wireshark. Esto se debe a que se utiliza el protocolo HTTP en lugar de HTTPS.

Al examinar con más detalle el contenido del paquete, se puede observar que incluye una gran cantidad de información sobre el solicitante. Se pueden identificar detalles como el navegador utilizado, las preferencias de idioma, la cookie de la sesión actual, así como la indicación de mantener la conexión abierta (keep-alive). Además, el paquete incluye una serie de encabezados Sec-fetch, que proporcionan información sobre el tipo de datos solicitados y otros detalles relacionados.

2.14. Explicación paquete burp (tráfico)

Del mismo modo, se llevó a cabo el procedimiento y se capturó el paquete utilizando Wireshark para su análisis, tal como se describió en el punto anterior.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.17.0.1	172.17.0.2	TCP	74	52086 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=602089457 TSecr=0 WS=128
2	0.000000181	172.17.0.1	172.17.0.2	TCP	74	52088 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=602089457 TSecr=0 WS=128
3	0.000047289	172.17.0.2	172.17.0.1	TCP	74	80 → 52086 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1627565863 TSecr=602089457 WS=128
4	0.000047359	172.17.0.2	172.17.0.1	TCP	74	80 → 52088 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1627565863 TSecr=602089457 WS=128
5	0.000056086	172.17.0.1	172.17.0.2	TCP	66	52088 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=602089457 TSecr=1627565863
6	0.000056306	172.17.0.1	172.17.0.2	TCP	66	52086 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=602089457 TSecr=1627565863
7	0.004024899	172.17.0.1	172.17.0.2	HTTP	943	GET /vulnerabilities/brute/?username=smithy&password=password&Login=Login HTTP/1.1
8	0.004097455	172.17.0.2	172.17.0.1	TCP	66	80 → 52088 [ACK] Seq=1 Ack=878 Win=64384 Len=0 TSval=1627565867 TSecr=602089461
9	0.043379652	172.17.0.2	172.17.0.1	HTTP	1893	HTTP/1.1 200 OK (text/html)
10	0.043397606	172.17.0.1	172.17.0.2	TCP	66	52088 → 80 [ACK] Seq=878 Ack=1828 Win=64000 Len=0 TSval=602089500 TSecr=1627565906
11	0.139562750	172.17.0.1	172.17.0.2	HTTP	776	GET /hackable/users/smithy.jpg HTTP/1.1
12	0.139614167	172.17.0.2	172.17.0.1	TCP	66	80 → 52088 [ACK] Seq=1828 Ack=1588 Win=64128 Len=0 TSval=1627566002 TSecr=602089596
13	0.141280793	172.17.0.2	172.17.0.1	HTTP	4735	HTTP/1.1 200 OK (JPEG JFIF image)
14	0.141322741	172.17.0.1	172.17.0.2	TCP	66	52088 → 80 [ACK] Seq=1588 Ack=6497 Win=62592 Len=0 TSval=602089598 TSecr=1627566004
15	5.146926850	172.17.0.2	172.17.0.1	TCP	66	80 → 52088 [FIN, ACK] Seq=6497 Ack=1588 Win=64128 Len=0 TSval=1627571010 TSecr=602089598
16	5.190295652	172.17.0.1	172.17.0.2	TCP	66	52088 → 80 [ACK] Seq=1588 Ack=6498 Win=64128 Len=0 TSval=602094647 TSecr=1627571010
17	15.190312720	172.17.0.1	172.17.0.2	TCP	66	[TCP Keep-Alive] 52086 → 80 [ACK] Seq=0 Ack=1 Win=64256 Len=0 TSval=602104647 TSecr=1627565863
18	15.202149812	172.17.0.2	172.17.0.1	TCP	66	[TCP Window Update] 80 → 52086 [ACK] Seq=1 Ack=1 Win=65152 Len=0 TSval=1627581065 TSecr=602089457
19	20.310026400	172.17.0.1	172.17.0.2	TCP	66	[TCP Keep-Alive] 52088 → 80 [ACK] Seq=1587 Ack=6498 Win=64128 Len=0 TSval=602109767 TSecr=1627571010
20	20.310076554	172.17.0.2	172.17.0.1	TCP	66	[TCP Keep-Alive ACK] 80 → 52088 [ACK] Seq=6498 Ack=1588 Win=64128 Len=0 TSval=1627586173 TSecr=602094647

```
Frame 7: 943 bytes on wire (7544 bits), 943 bytes captured (7544 bits) on interface docker0, id 0
Ethernet II, Src: 02:42:cd:b0:d5:eb (02:42:cd:b0:d5:eb), Dst: 02:42:ac:11:00:02 (02:42:ac:11:00:02)
Internet Protocol Version 4, Src: 172.17.0.1, Dst: 172.17.0.2
Transmission Control Protocol, Src Port: 52088, Dst Port: 80, Seq: 1, Ack: 1, Len: 877
Hypertext Transfer Protocol
  GET /vulnerabilities/brute/?username=smithy&password=password&Login=Login HTTP/1.1\r\n
  Host: localhost\r\n
  Connection: keep-alive\r\n
  sec-ch-ua: "Chromium";v="116", "Not)A;Brand";v="24", "Opera";v="102"\r\n
  sec-ch-ua-mobile: ?0\r\n
  sec-ch-ua-platform: "Linux"\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36 OPR/102.0.0.0\r\n
  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\r\n
  Sec-Fetch-Site: same-origin\r\n
  Sec-Fetch-Mode: navigate\r\n
  Sec-Fetch-User: ?1\r\n
  Sec-Fetch-Dest: document\r\n
  Referer: http://localhost/vulnerabilities/brute/?username=admin&password=password&Login=Login\r\n

0000  02 42 ac 11 00 02 02 42  cd b0 d5 eb 08 00 45 00   ·B·····B ######E·
0010  03 a1 ed 37 40 00 40 06  f1 f9 ac 11 00 01 ac 11   ###7@·@· ########
0020  00 02 cb 78 00 50 f1 de  fb 8a ec 87 a9 93 80 18   ###x·P·· ########
0030  01 f6 5b b9 00 00 01 01  08 0a 23 e3 27 f5 61 02   ··[····· ··#·'·a·
0040  af 27 47 45 54 20 2f 76  75 6c 6e 65 72 61 62 69   ·'GET /v ulnerabi
0050  6c 69 74 69 65 73 2f 62  72 75 74 65 2f 3f 75 73   lities/b rute/?us
0060  65 72 6e 61 6d 65 3d 73  6d 69 74 68 79 26 70 61   ername=s mithy&pa
0070  73 73 77 6f 72 64 3d 70  61 73 73 77 6f 72 64 26   ssword=p assword&
```

Figura 23: Captura wireshark de Burp

Podemos notar que el paquete es similar al enviado por Curl, ya que ambos utilizan el mismo protocolo, incluyen información del navegador, cookies y los mismos encabezados Sec-fetch. Sin embargo, el tamaño del paquete es ligeramente mayor en el caso de Burp Suite en comparación con Curl. Esto podría atribuirse a diferencias simples en los contenidos de los campos.

Por ejemplo, Burp Suite incluye referencias a los navegadores Chrome y Safari en su encabezado üser-agent", mientras que Curl no lo hace. En términos generales, ambos paquetes mantienen una estructura similar y proporcionan detalles sobre el cliente que realiza la solicitud.

2.15. Explicación paquete hydra (tráfico)

Al igual que los paquetes anteriores se analizo con Wireshark los paquetes de Hydra.

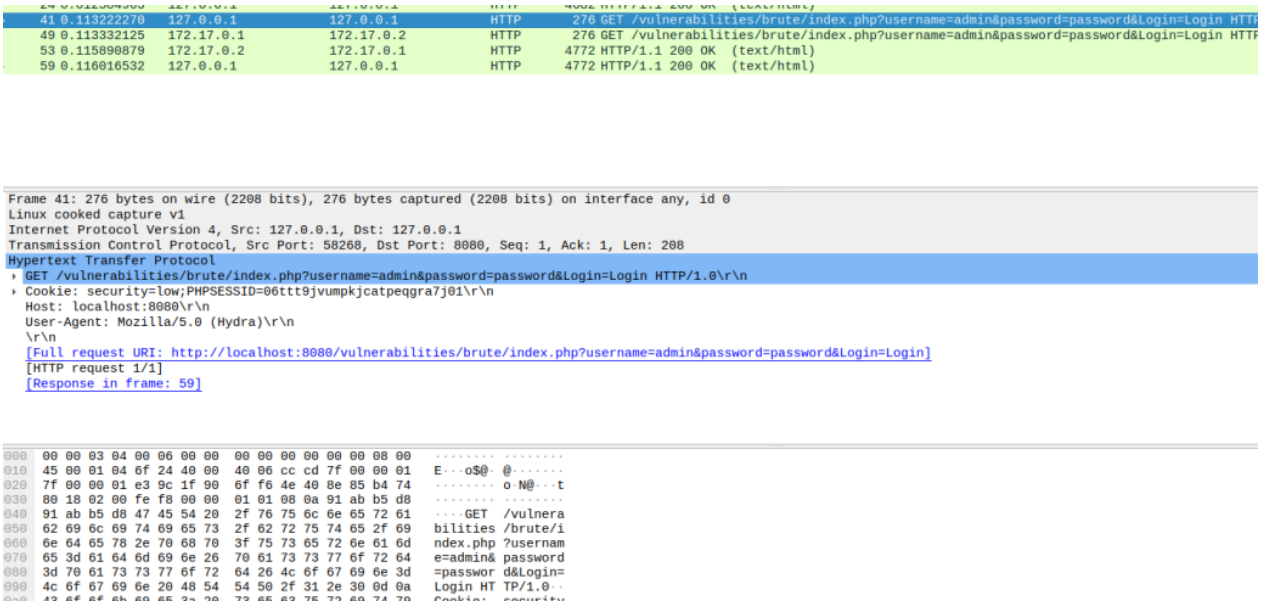


Figura 24: Captura wireshark de Hydra

En el caso de Hydra, observamos que el paquete es significativamente más corto en comparación con los casos anteriores. Esto se debe a que Hydra se especializa en ataques de fuerza bruta, y su objetivo es reducir al mínimo el tamaño de las solicitudes. En consecuencia, en la solicitud generada por Hydra, solo se incluyen los campos esenciales necesarios para una solicitud GET.

A diferencia de los otros métodos, Hydra no incorpora opciones relacionadas con el navegador ni los campos Sec-fetch. Esto se debe a que su enfoque principal es probar diferentes combinaciones de credenciales de acceso de manera eficiente y sin agregar información innecesaria a las solicitudes. La simplicidad en la estructura del paquete en Hydra permite un procesamiento más rápido de las solicitudes de fuerza bruta.

2.16. Mención de las diferencias (tráfico)

Hydra se destaca por su enfoque en ataques de fuerza bruta, lo que se refleja en sus solicitudes de tráfico que son de tamaño mínimo y carecen de información adicional, como datos de navegadores o encabezados Sec-fetch. Su objetivo principal es probar diferentes combinaciones de credenciales de acceso de manera eficiente.

Por otro lado, Curl y Burp Suite, aunque comparten algunas similitudes en la captura de tráfico, difieren en sus propósitos. Curl, una herramienta versátil de línea de comandos, incluye información detallada en sus solicitudes, como datos del navegador, preferencias de idioma, cookies y encabezados Sec-fetch. Por otro lado, Burp Suite es una herramienta integral de seguridad de aplicaciones web que proporciona capacidades avanzadas para pruebas de seguridad y análisis de tráfico, y también presenta solicitudes HTTP con detalles similares a Curl. Ambas herramientas utilizan HTTP en lugar de HTTPS, lo que significa que sus solicitudes son visibles en texto plano en la red.

2.17. Detección de SW (tráfico)

Al analizar minuciosamente los paquetes de tráfico de red, es posible distinguir las herramientas utilizadas según ciertos patrones y características específicas. Cada una de las herramientas en cuestión, Hydra, Burp Suite y Curl, presenta diferencias notables en cómo generan y manipulan las solicitudes de red.

Hydra, reconocido por su capacidad en ataques de fuerza bruta y diccionario, se caracteriza por generar múltiples intentos de inicio de sesión en un corto período de tiempo. Los paquetes generados por Hydra a menudo contienen comandos específicos para configurar diccionarios de contraseñas y nombres de usuario. Además, es común que estos paquetes se acompañen de mensajes de error particulares que indican el resultado de cada intento de inicio de sesión.

Por otro lado, Burp Suite es ampliamente utilizado para interceptar el tráfico web y llevar a cabo pruebas de seguridad en aplicaciones web. Los paquetes generados por Burp Suite generalmente incluyen solicitudes HTTP personalizadas que han sido modificadas para incorporar nombres de usuario y contraseñas específicas. Esta herramienta tiene la capacidad de agregar encabezados HTTP o cookies específicas a las solicitudes para evaluar la seguridad de la aplicación. La presencia de solicitudes de proxy intermedio en el tráfico indica que el tráfico web está siendo redirigido a través de Burp Suite.

Por último, Curl es una herramienta de línea de comandos diseñada para realizar solicitudes HTTP y, en general, genera paquetes más simples en comparación con Burp Suite y Hydra. Los paquetes de Curl contienen solicitudes HTTP estándar, y cada paquete suele representar una única solicitud. Es menos común que Curl realice múltiples intentos de inicio de sesión en una sola conexión, a menos que se utilice en un script para automatizar una secuencia de solicitudes. En resumen, cada una de estas herramientas tiene su propio propósito y enfoque en la generación de paquetes de tráfico de red.

Conclusiones y comentarios

El laboratorio que se llevó a cabo proporcionó una experiencia práctica valiosa en la evaluación de la seguridad de aplicaciones web y el análisis de tráfico de red. Durante las actividades, se desplegó la aplicación DVWA en un entorno controlado de Docker, lo que permitió realizar pruebas de seguridad de manera segura. Se demostró el poder de herramientas como Burp Suite, cURL y Hydra en la identificación de vulnerabilidades y credenciales débiles en aplicaciones web.

El ataque de fuerza bruta realizado con Burp Suite reveló la importancia de contar con políticas de seguridad sólidas y contraseñas seguras en las aplicaciones web. Se obtuvieron pares de usuario/contraseña válidos, lo que resalta la necesidad de proteger adecuadamente los sistemas contra estos tipos de ataques. El uso de cURL para automatizar las solicitudes HTTP desde la línea de comandos mostró su versatilidad en la realización de pruebas de seguridad. Finalmente, Hydra destacó por su velocidad y eficiencia en ataques de fuerza bruta, lo que lo convierte en una herramienta valiosa para evaluar la resistencia de contraseñas.

En la comparación de los paquetes de tráfico generados por Hydra, Burp Suite y CURL, se identificaron diferencias significativas en la estructura y contenido de los paquetes. Cada herramienta tenía su propio enfoque y objetivo, desde generar solicitudes mínimas hasta incluir información detallada del navegador. Este laboratorio enfatizó la importancia de la seguridad en aplicaciones web y cómo diferentes herramientas pueden utilizarse para evaluarla.

La experiencia resultó enriquecedora desde una perspectiva reflexiva, ya que permitió un significativo aprendizaje a pesar de los problemas encontrados durante la ejecución de este laboratorio. A pesar de los obstáculos, el contenido del laboratorio resultó ser interesante y entretendido, contribuyendo a un proceso de aprendizaje valioso.