Informe Laboratorio 2

Sección 1

Sofía Ignacia Belmar Alvarez
e-mail: sofia.belmar@mail.udp.cl

Abril de 2024

Índice

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 comenzar con el desarrollo de este laboratorio, se debe ejecutar un contenedor de Docker utilizando la imagen 'vulnerables/web-dvwa', la cual es una aplicación virtual virtual vulnerable a los ciberataques diseñada con la finalidad de entender cómo funcionan los ataques.
Para el levantamiento de este, se debió ejecutar el comando de 'docker run –rm -it -p 8880:80 –platform linux/amd64 vulnerables/web-dvwa', donde los parámetros utilizados se explicarán en la siguiente tabla.

Parámetro	Funcionamiento
docker run	Inicia un contenedor a partir de una imagen
–rm	Elimina el contenedor una vez que se detenga
-it	Habilita la interactividad con el contenedor
-p 8880:80	Permite que se pueda acceder a la aplicación en el puerto 8880
–platform linux/amd64	Especifica la plataforma del contenedor
vulnerables/web-dvwa	Imagen del contenedor

Una vez teniendo esto en cuenta, se coloca el comando en la terminal como se puede ver en la siguiente imagen.

Figura 1: Levantamiento de Docker con imagen 'vulnerables/web-dvwa'.

Como se puede observar en la Figura 1, una vez que se ejecuta el comando, descarga la imagen en caso de que no esté presente localmente en el dispositivo y generará un contenedor con la aplicación DVWA a partir de la imagen entregada, la cuál es interactiva.

2.2. Redirección de puertos en docker (dvwa)

Como se mencionó en el punto anterior, al ejecutar el comando, este mapea los puertos del contenedor Docker a los puetos de la máquina. Al ejecutar el contenedor de Docker con la aplicación de DVWA, se debe asegurar que se pueda acceder a la aplicación desde el navegador.
Por este motivo fue necesario agregar '-p 8880:80' dentro del comando, ya que se le da la instrucción de que tome el puerto 80 del contenedor y lo mapee al puerto 8880 en la máquina. Esto permite que luego se pueda acceder a la dirección 'http://localhost:8880/vulnerabilities/brute/'.

2.3. Obtención de consulta a replicar (burp)

El proceso comenzó configurando el navegador para que apuntara a localhost. Esto se logró accediendo a la configuración de red del sistema y habilitando el proxy con la dirección

URL 127.0.0.1, que corresponde a la dirección IP de localhost. Esta configuración se muestra en la Figura 2:

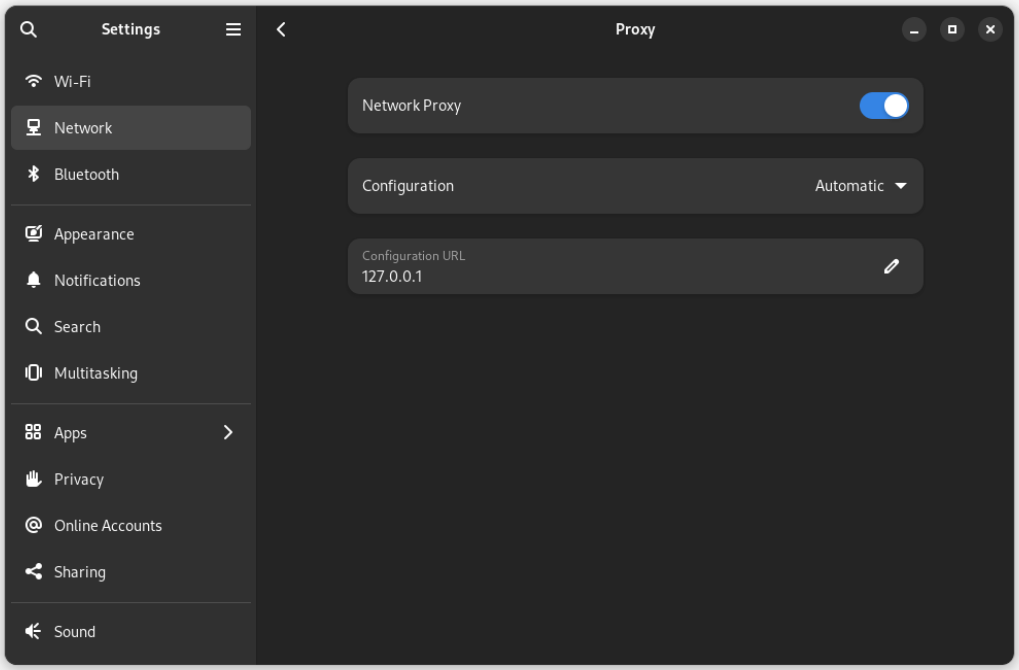

Figura 2: Configuración del proxy para la interceptación de tráfico.

Seguido de esto, se debe abrir la aplicación de BurpSuite, en donde se debe abrir el navegador y comenzar a interceptar.
Una vez que se tenga el navegador abierto, se debe buscar la dirección 'http://localhost:8880/vulnerabilities/brute/'. Lo primero que se logra ver es un inicio de sesión, el cual se puede observar en la Figura 3.

Figura 3: Interfaz de inicio de sesión de DVWA.

Para replicar la consulta, era necesario completar los campos de autenticación con las credenciales de prueba. Se utilizaron 'admin' como usuario y 'password' como contraseña. Tras la autenticación, se puede observar una nueva ruta, la cual dará la opción de crear y/o resetear la base de datos. Dicha ruta se puede observar en la Figura 4.

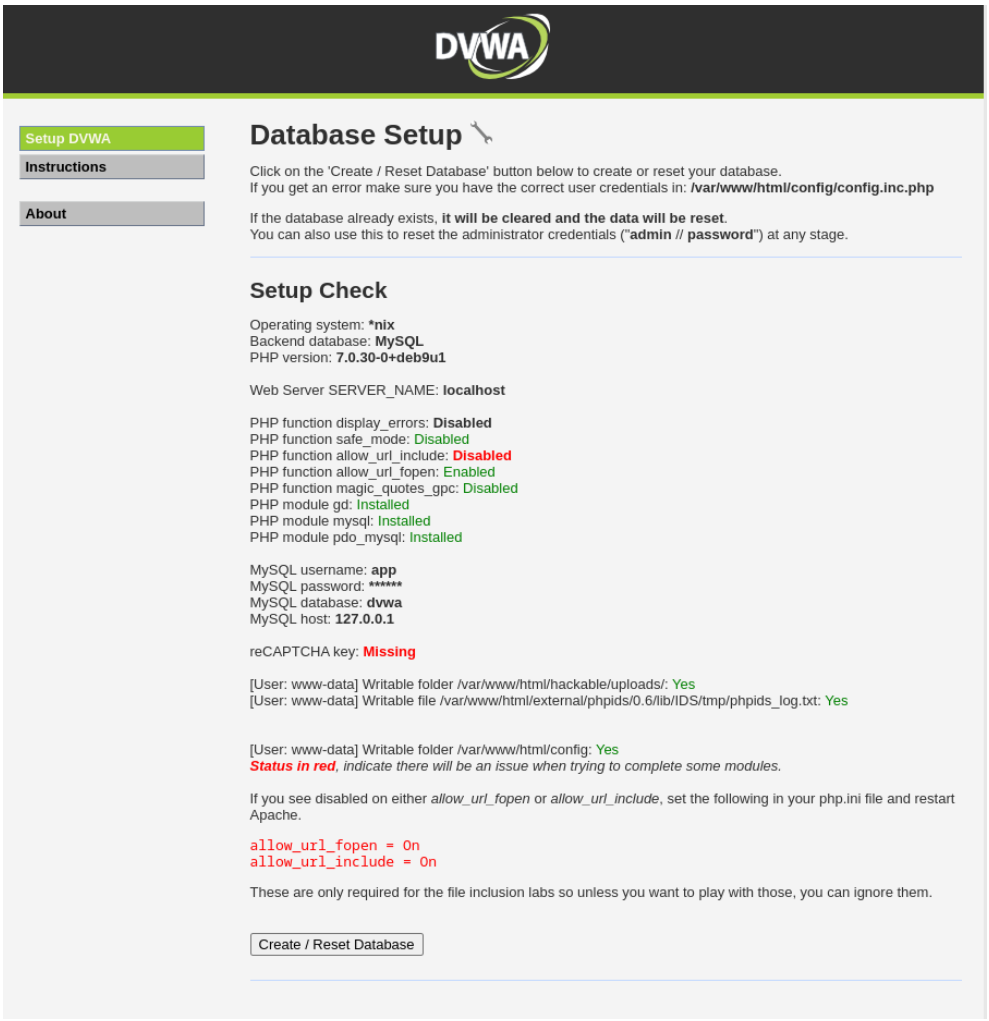

Figura 4: Opción para la gestión de la base de datos en DVWA.

Esta opción, una vez seleccionada, redirige a una nueva ruta visualizable en la Figura 5.

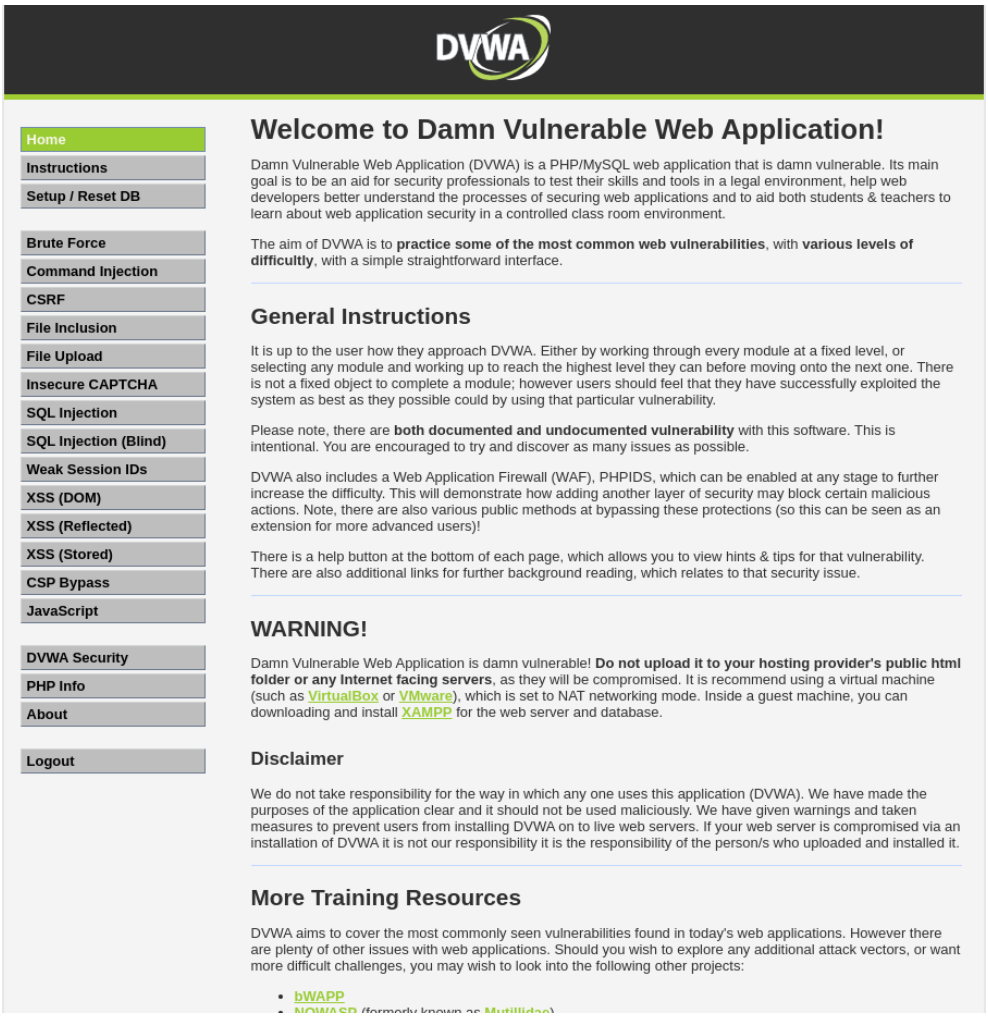

Figura 5: Redirección tras la gestión de la base de datos de DVWA.

Como se puede apreciar en la imagen, la aplicación muestra diversos ítems al costado. El ítem relevante para continuar es 'Brute Force'. Al seleccionarlo, se inicia el proceso de ataque por fuerza bruta.

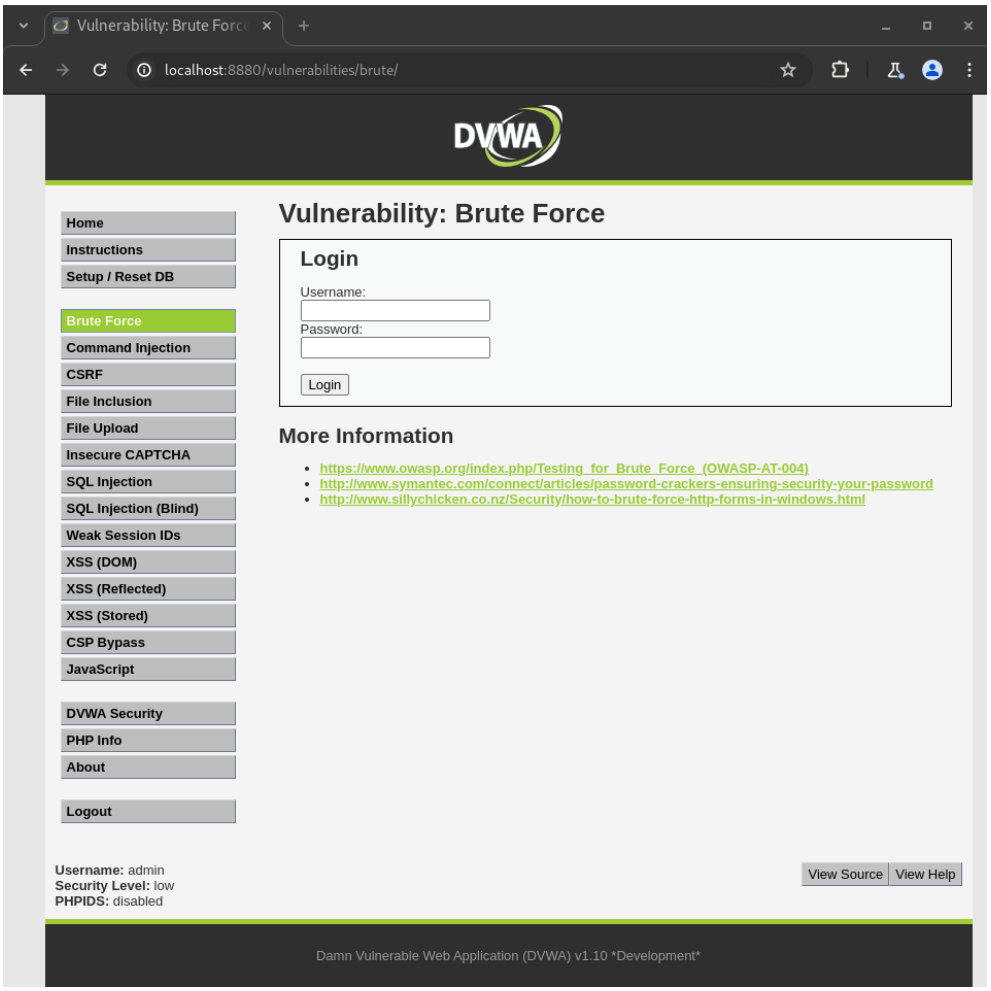

Figura 6: Interfaz para iniciar el ataque de fuerza bruta.

En la Figura 6 se muestra la interfaz de inicio de sesión, la cual será objetivo de un ataque por fuerza bruta. Para iniciar este proceso, se ingresaron 'admin' y 'password' en los campos de usuario y contraseña respectivamente. Una vez ingresados estos datos, BurpSuite intercepta la información enviada durante el proceso de autenticación.

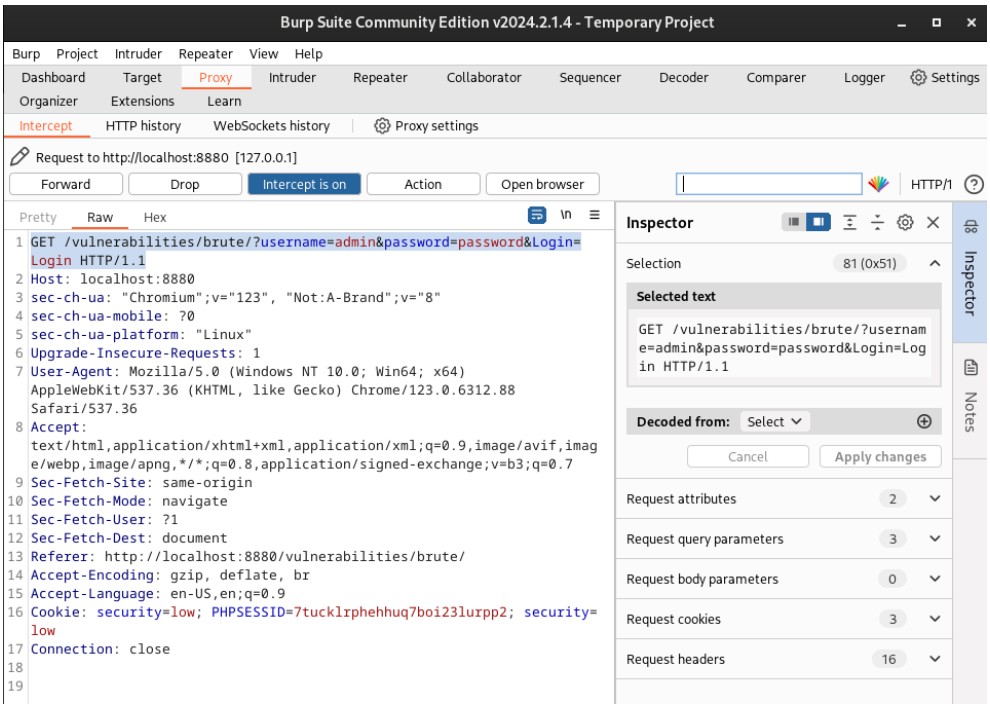

Figura 7: Información interceptada con BurpSuite.

En la Figura 7 se puede observar la información interceptada al momento de autenticar. Teniendo esta información, esta es enviada a 'Intruder', en donde se tomarán los campos y se realizará el ataque de fuerza bruta.

2.4. Identificación de campos a modificar (burp)

Cuando la información interceptada es enviada desde el 'Proxy' a 'Intruder', se mostrará en este campo como se ilustra en la imagen a continuación.

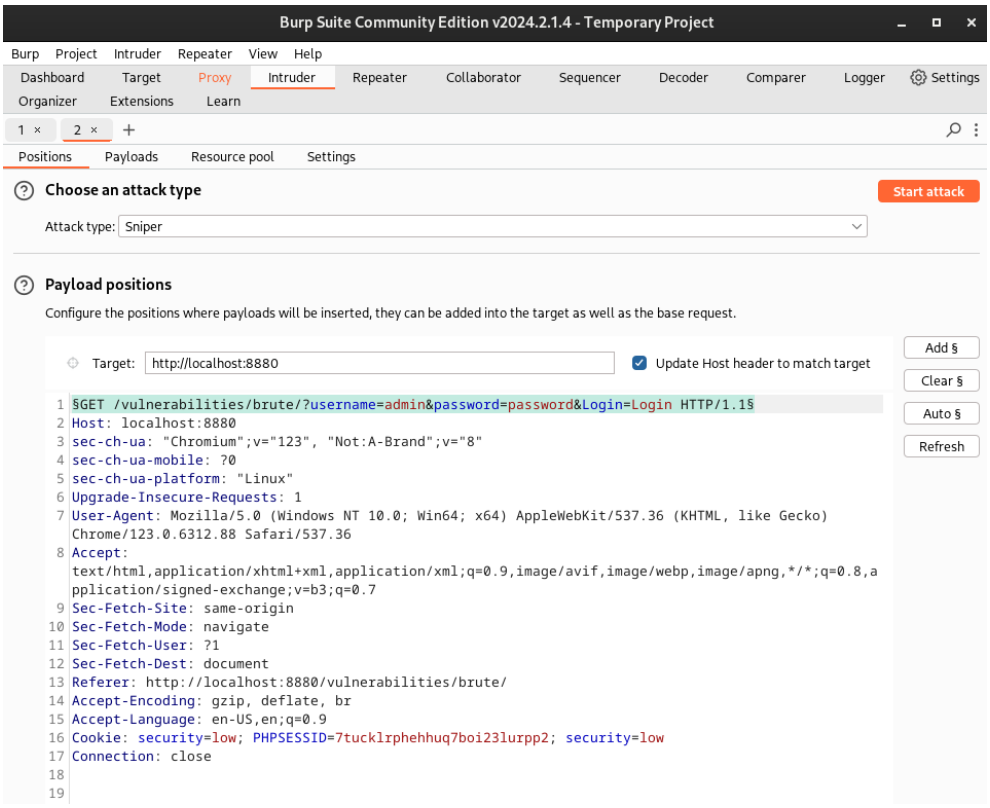

Figura 8: Información interceptada por BurpSuite.

Teniendo esta petición, es necesario definir las posiciones de los payloads. Esto implica especificar dónde se insertarán los intentos de nombres de usuario y contraseñas en la solicitud.
Para establecer estos campos como payloads, en este caso, 'admin' y 'password', se deben seleccionar y posteriormente utilizar la opción de 'Add'.
La descripción mencionada se puede observar en la Figura 9.

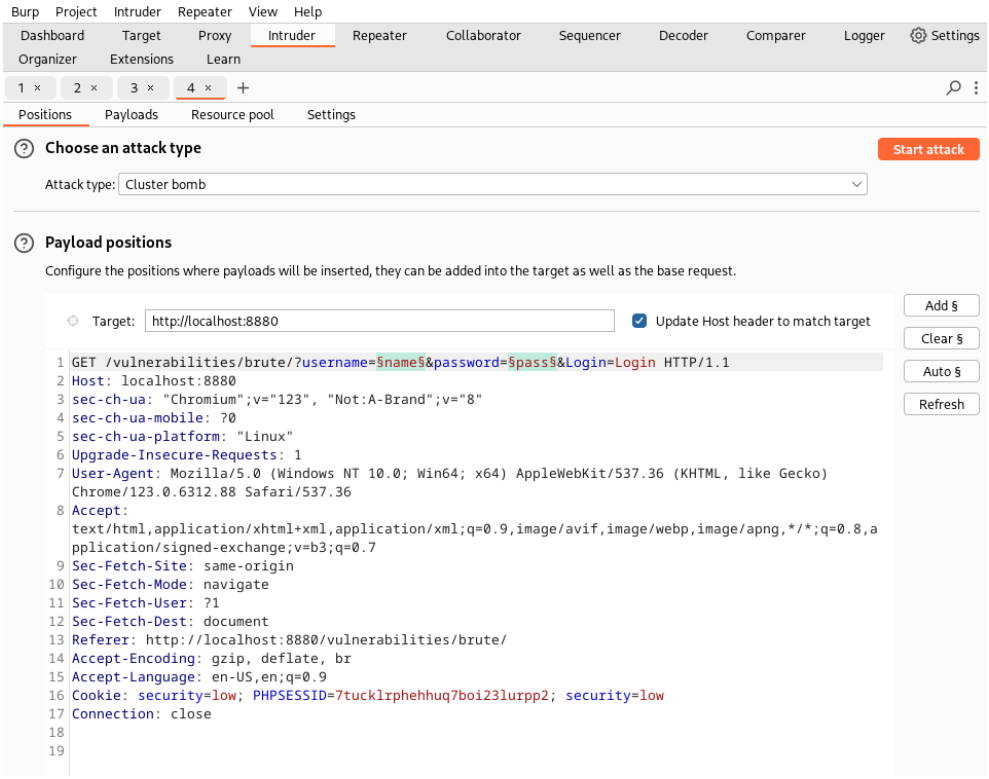

Figura 9: Definición de payloads.

Cabe mencionar que para generar el ataque por fuerza bruta, fue necesario colocar 'Cluster bomb' como tipo de ataque.

2.5. Obtención de diccionarios para el ataque (burp)

Para adquirir credenciales válidas, se utilizó información de la página 'https://medium.com/@aayanx41/dvwa-brute-force-c2901f630c3d', donde se listan usuarios y contraseñas que funcionan en el sitio objetivo.

Para elaborar el diccionario de ataque, se seleccionaron tres credenciales de esta fuente y se complementaron con otros nombres de usuario y contraseñas extraídos de una base de datos que contiene las credenciales más frecuentemente utilizadas.

2.6. Obtención de al menos 2 pares (burp)

Para obtener credenciales válidas mediante un ataque de fuerza bruta, es necesario probar diversas combinaciones de usuarios y contraseñas contenidas en el diccionario previamente elaborado. Este proceso se realiza en la sección 'Intruder/Payload' de BurpSuite.
En esta área, se debe seleccionar la opción 'Payload Sets', que permite especificar si está configurado el payload para nombres de usuario o contraseñas. Posteriormente, en 'Payload

Settings', se ingresan los datos del diccionario preparado para el ataque.

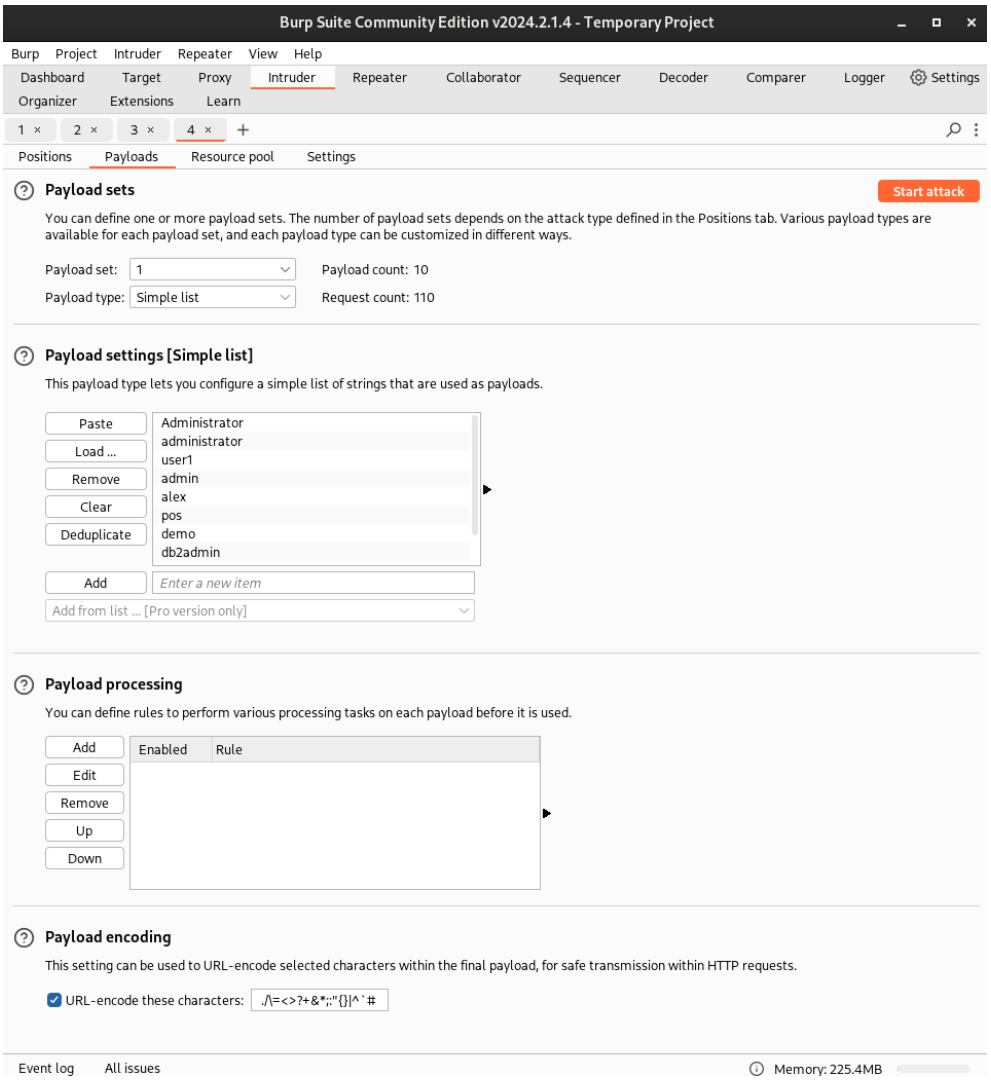

Figura 10: Configuración de payload.

En la Figura 10 se pueden observar las credenciales introducidas para el ataque.

Seguido de esto, se debe seleccionar la opción 'Start attack'. Al hacerlo, Burp Suite iniciará el ataque de fuerza bruta, probando todas las combinaciones de usuarios y contraseñas configuradas en el diccionario.

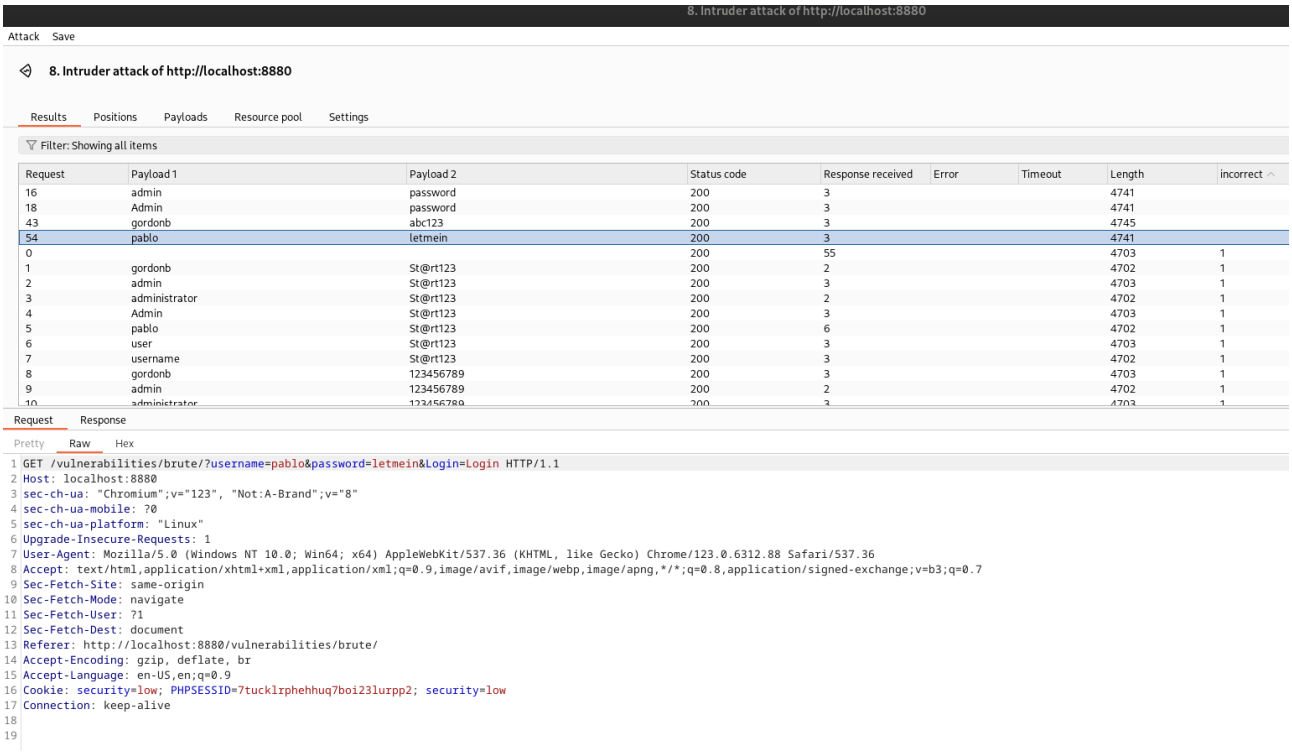

Figura 11: Ataque por fuerza bruta con BurpSuite.

En la Figura 11 se pueden observar los ataques y las credenciales que se utilizaron para esto. Además, es posible apreciar que en el apartado de 'incorrect', aquellos espacios en blanco indican que las credenciales son válidas para ingresar a la página, mientras que un '1' significa que las credenciales no son válidas.

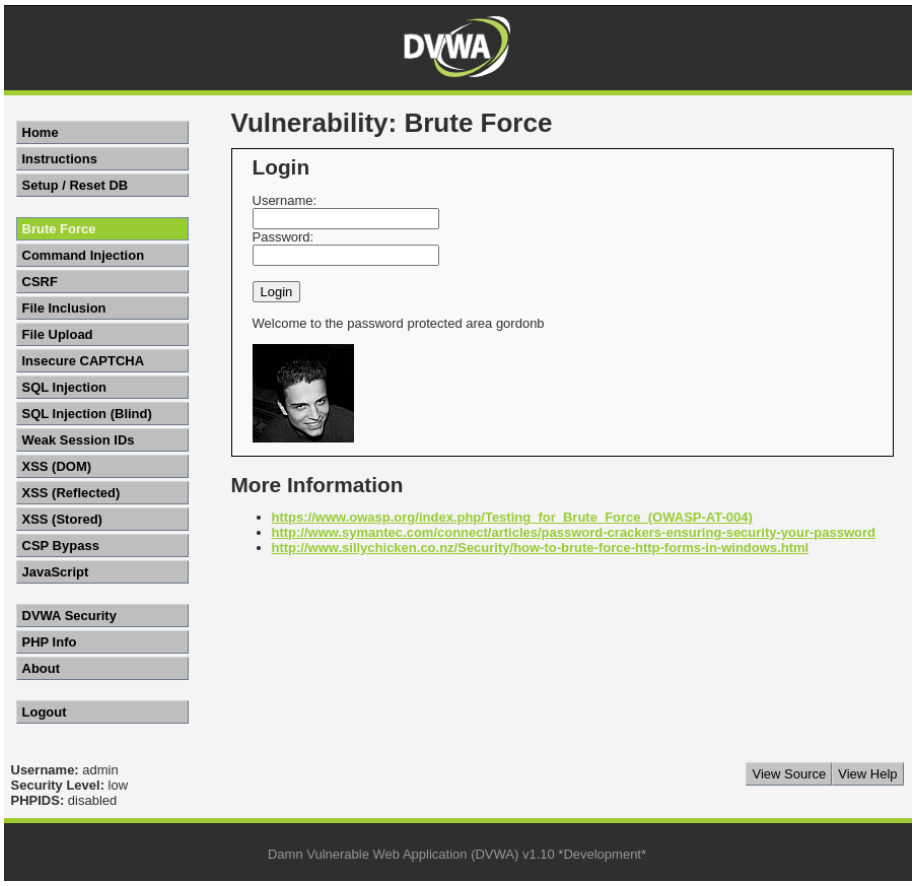

Figura 12: Inicio de sesión con credenciales válidas en DVWA.

Finalmente, para verificar la autenticidad de estas credenciales, se prueban en la página de inicio de sesión. Como se puede observar en la Figura 12, la verificación se confirma con un mensaje y una imagen que indica que las credenciales son válidas.

2.7. Obtención de código de inspect element (curl)

Para utilizar cURL eficazmente, es necesario estar en la misma sesión que se usó durante el proceso de ataque de fuerza bruta con Burp Suite. Primero, se deben abrir las herramientas de desarrollador del navegador y navegar a la pestaña de "Network".

A continuación, se debe iniciar sesión con alguna de las credenciales válidas. En la pestaña "Network", se podrá visualizar la consulta generada por este inicio de sesión, como se muestra en la Figura 13.

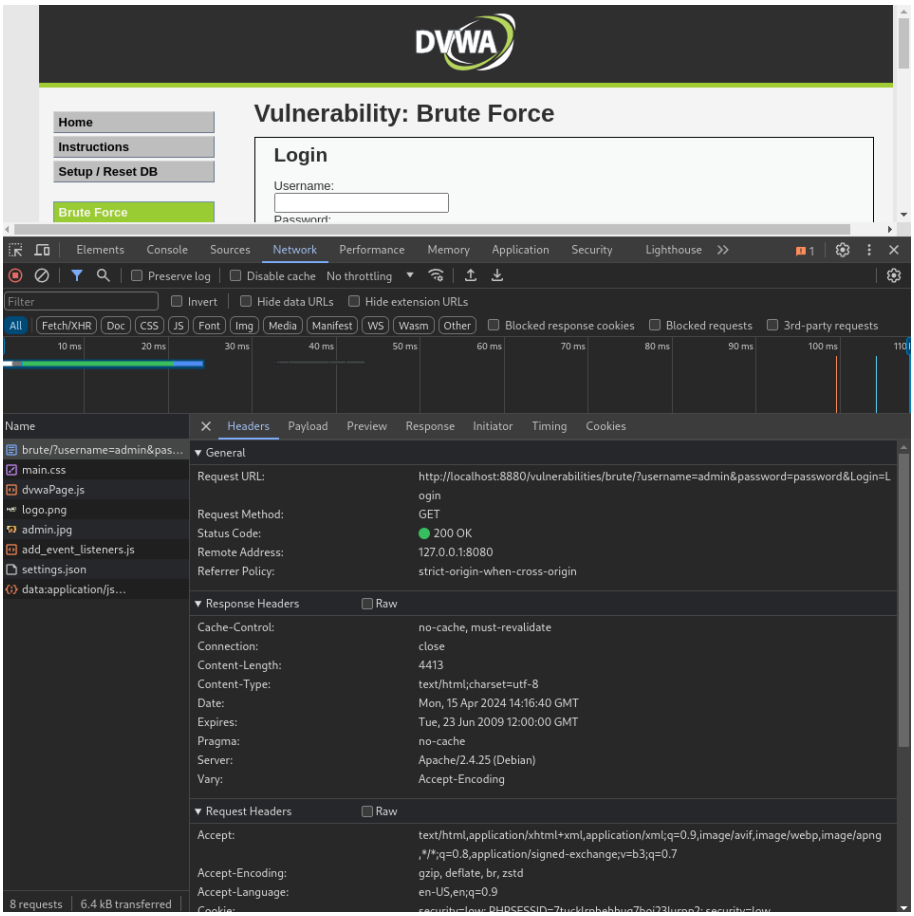

Figura 13: Herramientas de desarrollador en DVWA.

Una vez identificada la consulta, se debe copiar como cURL para poder replicar la solicitud y visualizar la página de manera independiente utilizando esta herramienta de línea de comandos.

2.8. Utilización de curl por terminal (curl)

Una vez obtenido el comando cURL, este se introduce en la línea de comandos de la terminal. Al ejecutarlo, se recibirá como respuesta el contenido de la página web, que se puede visualizar de la siguiente manera:

```
~ git:(master) (0.244s)
curl 'http://localhost:8880/vulnerabilities/brute/?username=administrator&password=pass&Login=Login' \
  -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7' \
  -H 'Accept-Language: en-US,en;q=0.9' \
  -H 'Cookie: security=low; PHPSESSID=7tucklrphehhuq7boi23lurpp2; security=low' \
  -H 'Proxy-Connection: keep-alive' \
  -H 'Referer: http://localhost:8880/vulnerabilities/brute/?username=a&password=b&Login=Login' \
  -H 'Sec-Fetch-Dest: document' \
  -H 'Sec-Fetch-Mode: navigate' \
  -H 'Sec-Fetch-Site: same-origin' \
  -H 'Sec-Fetch-User: ?1' \
  -H 'Upgrade-Insecure-Requests: 1' \
  -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.88 Safari/537.36' \
  -H 'sec-ch-ua: "Chromium";v="123", "Not:A-Brand";v="8"' \
  -H 'sec-ch-ua-mobile: ?0' \
  -H 'sec-ch-ua-platform: "Linux"'

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

<html xmlns="http://www.w3.org/1999/xhtml">

        <head>
                <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

                <title>Vulnerability: Brute Force :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>

                <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />

                <link rel="icon" type="\image/ico" href="../../favicon.ico" />

                <script type="text/javascript" src="../../dvwa/js/dvwaPage.js"></script>

        </head>

        <body class="home">
                <div id="container">

                        <div id="header">

                                <img src="../../dvwa/images/logo.png" alt="Damn Vulnerable Web Application" />

                        </div>

                        <div id="main_menu">

                                <div id="main_menu_padded">
                                <ul class="menuBlocks"><li class=""><a href="../../.">Home</a></li>
<li class=""><a href="../../instructions.php">Instructions</a></li>
<li class=""><a href="../../setup.php">Setup / Reset DB</a></li>
</ul><ul class="menuBlocks"><li class="selected"><a href="../../vulnerabilities/brute/">Brute Force</a></li>
<li class=""><a href="../../vulnerabilities/exec/">Command Injection</a></li>
<li class=""><a href="../../vulnerabilities/csrf/">CSRF</a></li>
<li class=""><a href="../../vulnerabilities/fi/.?page=include.php">File Inclusion</a></li>
```

Figura 14: Línea de comandos al introducir cURL con credenciales válidas.

Este proceso se repite con credenciales inválidas y posteriormente se comparan las páginas. En la Figura 15 se puede observar la línea de comandos al introducir cURL con credenciales inválidas.

```
~ git:(master) (0.244s)
curl 'http://localhost:8880/vulnerabilities/brute/?username=administrator&password=pass&Login=Login' \
  -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7' \
  -H 'Accept-Language: en-US,en;q=0.9' \
  -H 'Cookie: security=low; PHPSESSID=7tucklrphehhuq7boi23lurpp2; security=low' \
  -H 'Proxy-Connection: keep-alive' \
  -H 'Referer: http://localhost:8880/vulnerabilities/brute/?username=a&password=b&Login=Login' \
  -H 'Sec-Fetch-Dest: document' \
  -H 'Sec-Fetch-Mode: navigate' \
  -H 'Sec-Fetch-Site: same-origin' \
  -H 'Sec-Fetch-User: ?1' \
  -H 'Upgrade-Insecure-Requests: 1' \
  -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.88 Safari/537.36' \
  -H 'sec-ch-ua: "Chromium";v="123", "Not:A-Brand";v="8"' \
  -H 'sec-ch-ua-mobile: ?0' \
  -H 'sec-ch-ua-platform: "Linux"'

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

<html xmlns="http://www.w3.org/1999/xhtml">

        <head>
                <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

                <title>Vulnerability: Brute Force :: Damn Vulnerable Web Application (DVWA) v1.10 *Development*</title>

                <link rel="stylesheet" type="text/css" href="../../dvwa/css/main.css" />

                <link rel="icon" type="\image/ico" href="../../favicon.ico" />

                <script type="text/javascript" src="../../dvwa/js/dvwaPage.js"></script>

        </head>

        <body class="home">
                <div id="container">

                        <div id="header">

                                <img src="../../dvwa/images/logo.png" alt="Damn Vulnerable Web Application" />

                        </div>

                        <div id="main_menu">

                                <div id="main_menu_padded">
                                <ul class="menuBlocks"><li class=""><a href="../../.">Home</a></li>
<li class=""><a href="../../instructions.php">Instructions</a></li>
<li class=""><a href="../../setup.php">Setup / Reset DB</a></li>
</ul><ul class="menuBlocks"><li class="selected"><a href="../../vulnerabilities/brute/">Brute Force</a></li>
<li class=""><a href="../../vulnerabilities/exec/">Command Injection</a></li>
<li class=""><a href="../../vulnerabilities/csrf/">CSRF</a></li>
<li class=""><a href="../../vulnerabilities/fi/.?page=include.php">File Inclusion</a></li>
```

Figura 15: Línea de comandos al introducir cURL con credenciales inválidas.

2.9. Demuestra 5 diferencias (curl)

Dentro de las diferencias que se encontraron al ejecutar el comando cURL, se pudieron evidenciar las siguientes:

1. **Mensaje de la página**: Dentro de los errores que existen, el más evidente es el mensaje que muestra al ingresar con alguna credencial válida o inválida. Al ingresar con una credencial inválida, la página muestra el mensaje 'Username and/or password incorrect' bajo el formulario de inicio de sesión, mientras que al ingresar con alguna credencial válida, se muestra el mensaje 'Welcome to the password protected area admin'.

2. **Cookies**: A pesar de que estas no sean visibles en el HTML de la página, sí se puede observar un cambio en las cookies. Al iniciar sesión de manera exitosa, se envía una cookie distinta, en caso contrario, la cookie no cambia.

3. **Tiempo de respuesta**: Otra de las diferencias que existen es el tiempo de respuesta del servidor, donde en el inicio de sesión con credenciales válidas, se tardó 0.244 segundos,

mientras que en el inicio de sesión con credenciales inválidas se tardó 0.213 segundos.

4. **Recursos**: Al comparar las páginas, también se puede evidenciar que, al ingresar con credenciales válidas, la página retorna una imagen relacionada con el usuario autenticado. Esto sólo ocurre al inicio de sesión con credenciales válidas.

5. **Longitud de la respuesta**: Finalmente, se puede notar una diferencia entre las longitudes de las respuestas, en donde al iniciar sesión con credenciales válidas se obtiene una respuesta de 3213 bytes (cuando se inicia con el usuario 'admin'), y una respuesta de 3159 bytes al ingresar con credenciales inválidas. Esto se puede dar debido a que el código HTML incrementa cuando se ingresa de manera exitosa, ya que se incluye una imagen y un texto. Cabe mencionar que la longitud de la respuesta de inicio de sesión de una credencial válida puede variar dependiendo de la imagen que esté relacionada a ese usuario.

2.10. Instalación y versión a utilizar (hydra)

Para la instalación de Hydra, es necesario utilizar el comando 'sudo dnf install hydra'. Una vez instalado, se verifica la version de Hydra instalada en el sistema con el comando 'rpm -q hydra'. Lo que hace este comando es consultar a la base de datos RPM sobre la información del paquete de Hydra, proporcionando su versión. Ambos comandos se pueden apreciar en la Figura 16.

Figura 16: Comandos de instalación y versión de Hydra.

2.11. Explicación de comando a utilizar (hydra)

Para realizar un ataque de fuerza bruta utilizando Hydra, se empleó el siguiente comando:

```
hydra -L <username.txt> -P <passwords.txt> localhost -s 8880 http-get-form
"/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:H=Cookie\:
PHPSESSID=83ohku3mgeqaj3n6o42bionm80; security=low:F=Username and/or password
incorrect"
```

Este comando especifica listas de nombres de usuario y contraseñas ubicadas en los archivos username.txt y passwords.txt, respectivamente. En la ejecución, Hydra sustituye los marcadores 'USER' y 'PASS' en los parámetros de username y password con los datos de estas listas. Cada solicitud incluye también una cookie con un PHPSESSID específico, lo que permite mantener la sesión entre múltiples intentos de acceso. Además, el comando logra identificar respuestas fallidas a través de la frase 'Username and/or password incorrect'.

2.12. Obtención de al menos 2 pares (hydra)

Inicialmente, se crearon dos listas con los datos a probar: username.txt y passwords.txt. Estas listas incluyen los caracteres que se intentarán durante el ataque de fuerza bruta con Hydra. Posteriormente, es necesario incluir el PHPSESSID en la consulta. Para esto, se accede a la aplicación y se abren las herramientas de desarrollador, navegando hasta la sección 'Applications' donde se encuentra el ítem de cookies. Aquí, se puede visualizar el PHPSESSID. A continuación, se puede observar la Figura 17, la cual muestra este proceso con mayor detalle.

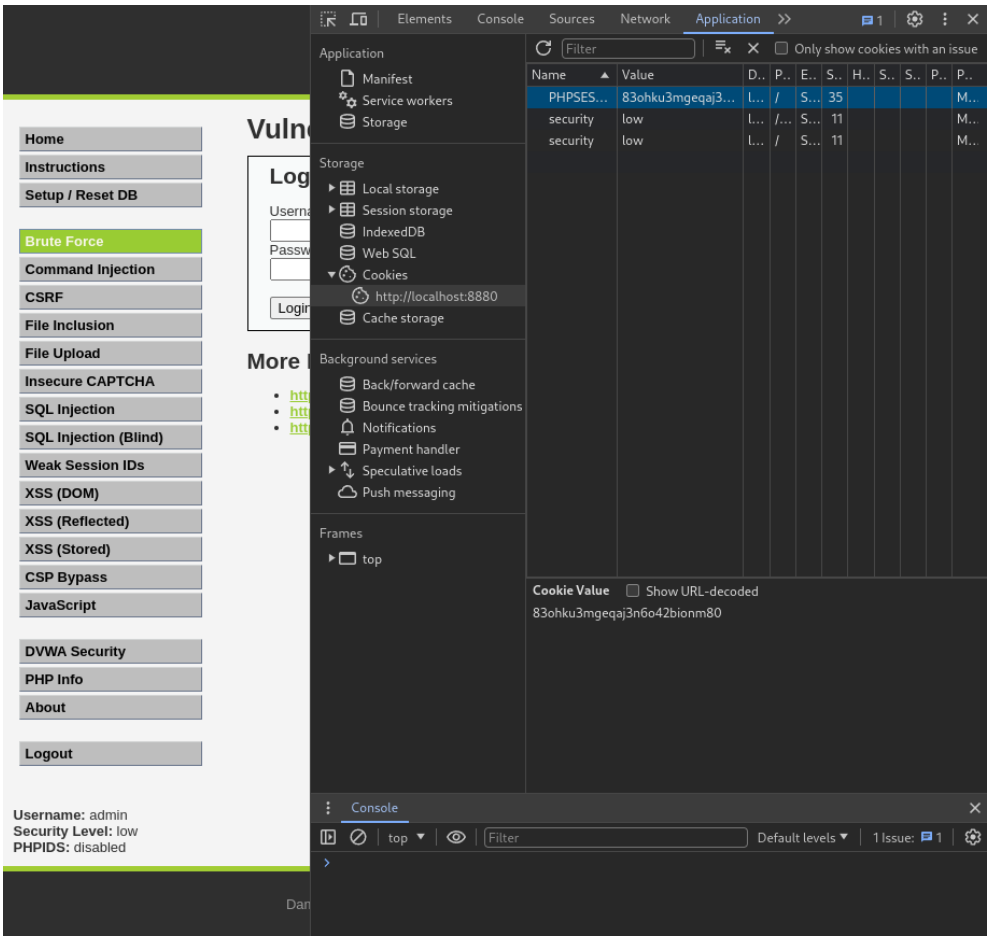

Figura 17: Herramientas de desarrollador para obtener Cookies.

Una vez recopilados estos datos, se utiliza el comando descrito anteriormente, sustituyendo las listas con los nombres de los archivos que se encuentren en la máquina, y el PHPSESSID con el valor obtenido de las Cookies.

```
~/Documents/Universidad/Cripto/Lab2/Data git:(main)±26 (1.697s)
hydra -L username.txt -P passwords.txt localhost -s 8880 http-get-form "/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:H=Cookie\:PHPSESSID=mmtj86okolpkeq9h80i06dli23; security=low:F=Username and/or password incorrect"

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-18 08:24:09
[INFORMATION] escape sequence \: detected in module option, no parameter verification is performed.
[DATA] max 16 tasks per 1 server, overall 16 tasks, 49 login tries (l:7/p:7), ~4 tries per task
[DATA] attacking http-get-form://localhost:8880/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:H=Cookie\:PHPSESSID=mmtj86okolpkeq9h80i06dli23; security=low:F=Username and/or password incorrect
[8880][http-get-form] host: localhost   login: admin   password: password
[8880][http-get-form] host: localhost   login: pablo   password: letmein
[8880][http-get-form] host: localhost   login: gordonb   password: abc123
[8880][http-get-form] host: localhost   login: Admin   password: password
1 of 1 target successfully completed, 4 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-18 08:24:10
```

Figura 18: Comando para ejecutar ataque por fuerza bruta con Hydra.

Al ejecutar este comando, como se muestra en la Figura 18, Hydra realiza el ataque de fuerza bruta con las credenciales proporcionadas en los archivos .txt y posteriormente identifica y muestra aquellas que resultaron ser válidas.

2.13. Explicación paquete curl (tráfico)

Al momento de ejecutar el comando de cURL, se hizo uso del software Wireshark, el cual ayudó a capturar los paquetes durante su ejecución.

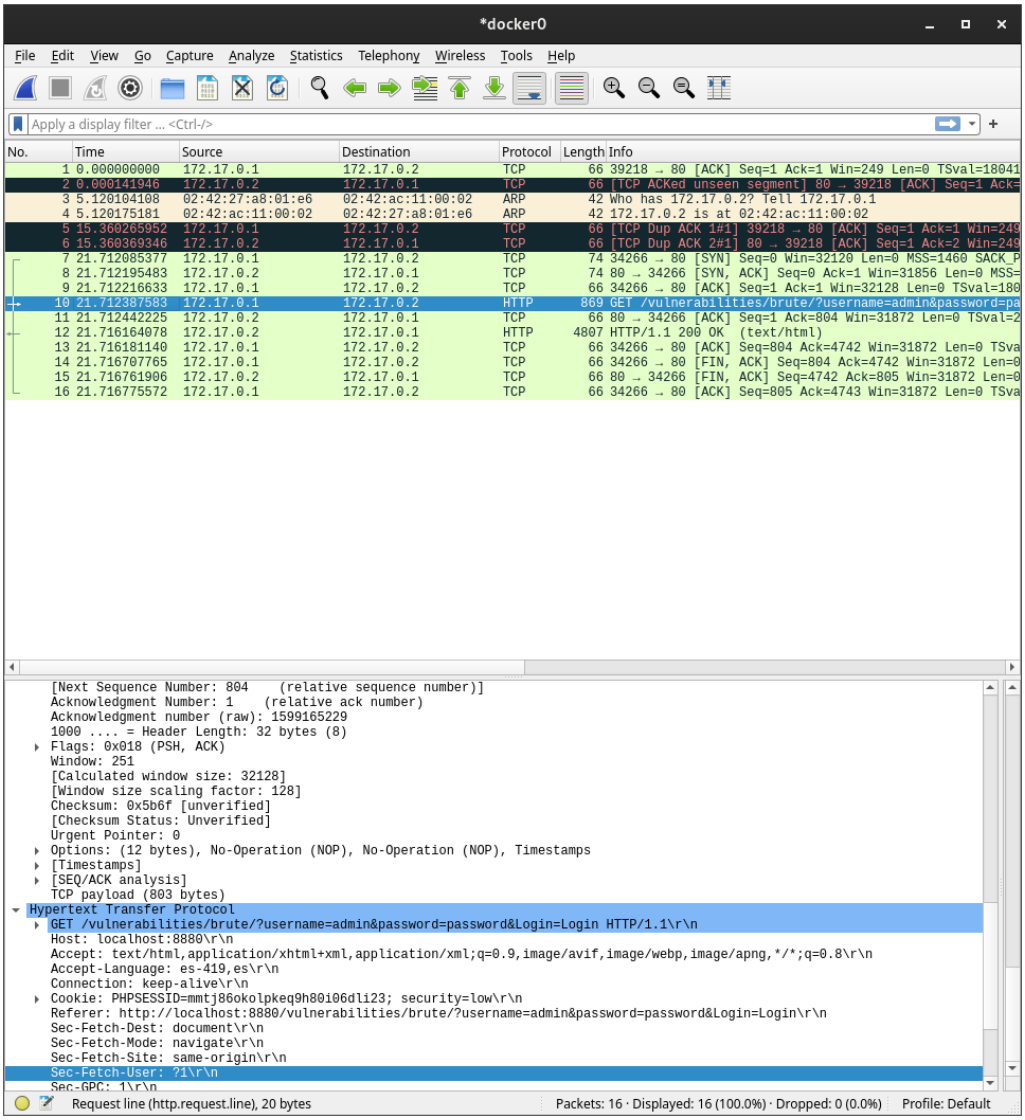

Figura 19: Captura de paquetes durante un ataque de fuerza bruta utilizando cURL.

Como se observa en la Figura 19, los paquetes capturados corresponden a solicitudes HTTP GET enviadas desde un cliente a un servidor dentro de la red de Docker. Los flags PSH y ACK indican que los paquetes están transmitiendo datos y confirmando la recepción de paquetes anteriores respectivamente.

La captura muestra un intento de inicio de sesión, en el que los parámetros 'username' y 'password' son transmitidos como parte de la URL. Esto es característico de un ataque de fuerza bruta, donde se prueban combinaciones de credenciales para acceder a áreas protegidas de la aplicación web.

2.14. Explicación paquete burp (tráfico)

Al igual que como se mencionó en el ítem anterior, se realizó una captura de paquetes durante el ataque por fuerza bruta con BurpSuite.

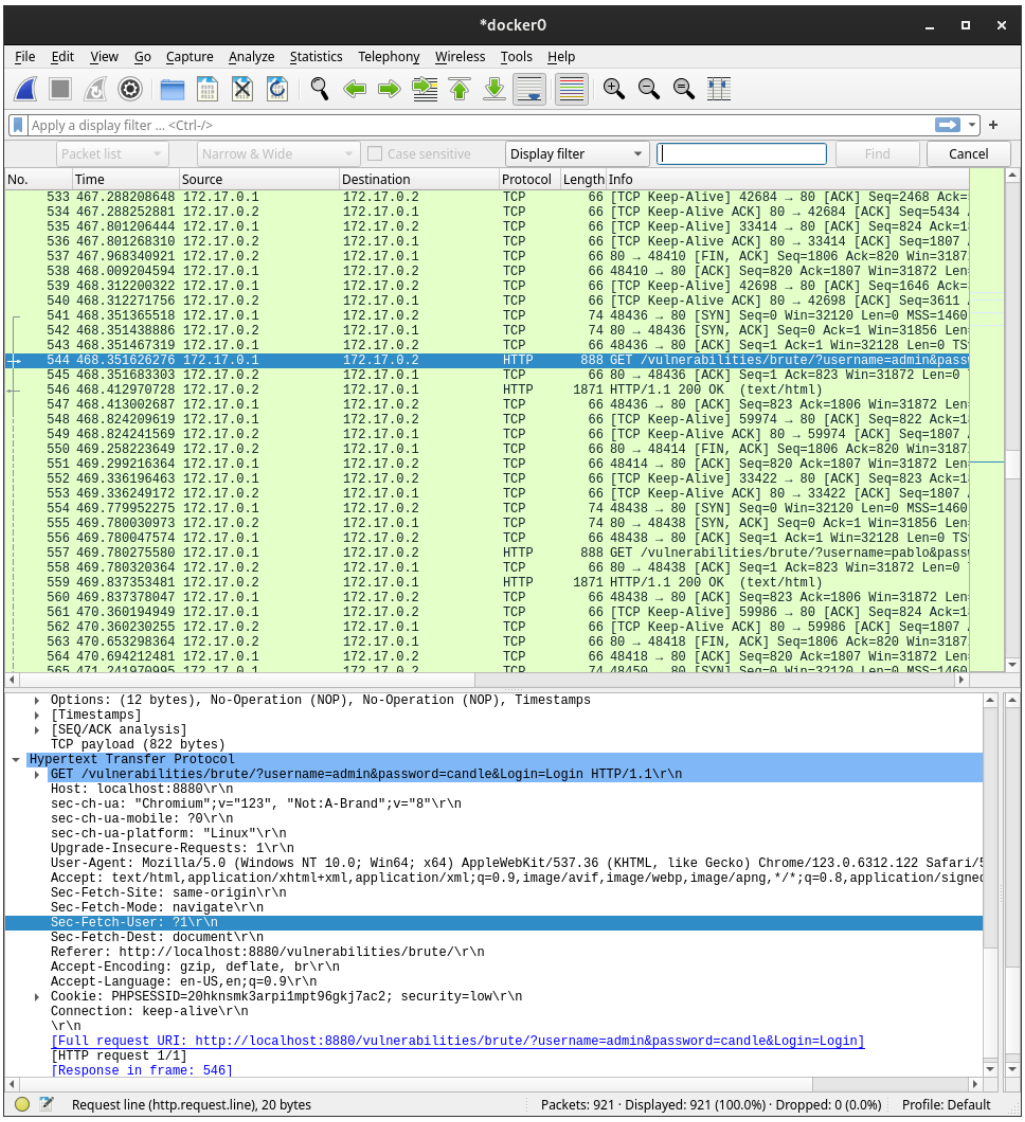

Figura 20: Captura de paquetes durante un ataque de fuerza bruta utilizando cURL.

En la Figura 20, se pueden apreciar los paquetes TCP que emplean el método GET, con un TTL (Time to Live) establecido en 64 y un encabezado de 32 bytes de longitud. Estos paquetes contienen detalles de la solicitud HTTP, incluyendo cookies y 'Sec-Fetch', entre otros elementos relacionados con la conexión.
La solicitud HTTP GET intenta acceder a la ruta entregada a través de una URL que incluye un nombre de usuario y una contraseña, formando parte de la consulta.

2.15. Explicación paquete Hydra (tráfico)

Finalmente se analiza el paquete capturado de Hydra. En este se pueden observar varias solicitudes HTTP GET utilizadas en un ataque de fuerza bruta. En la Figura 21 muestra uno de estos paquetes interceptados.

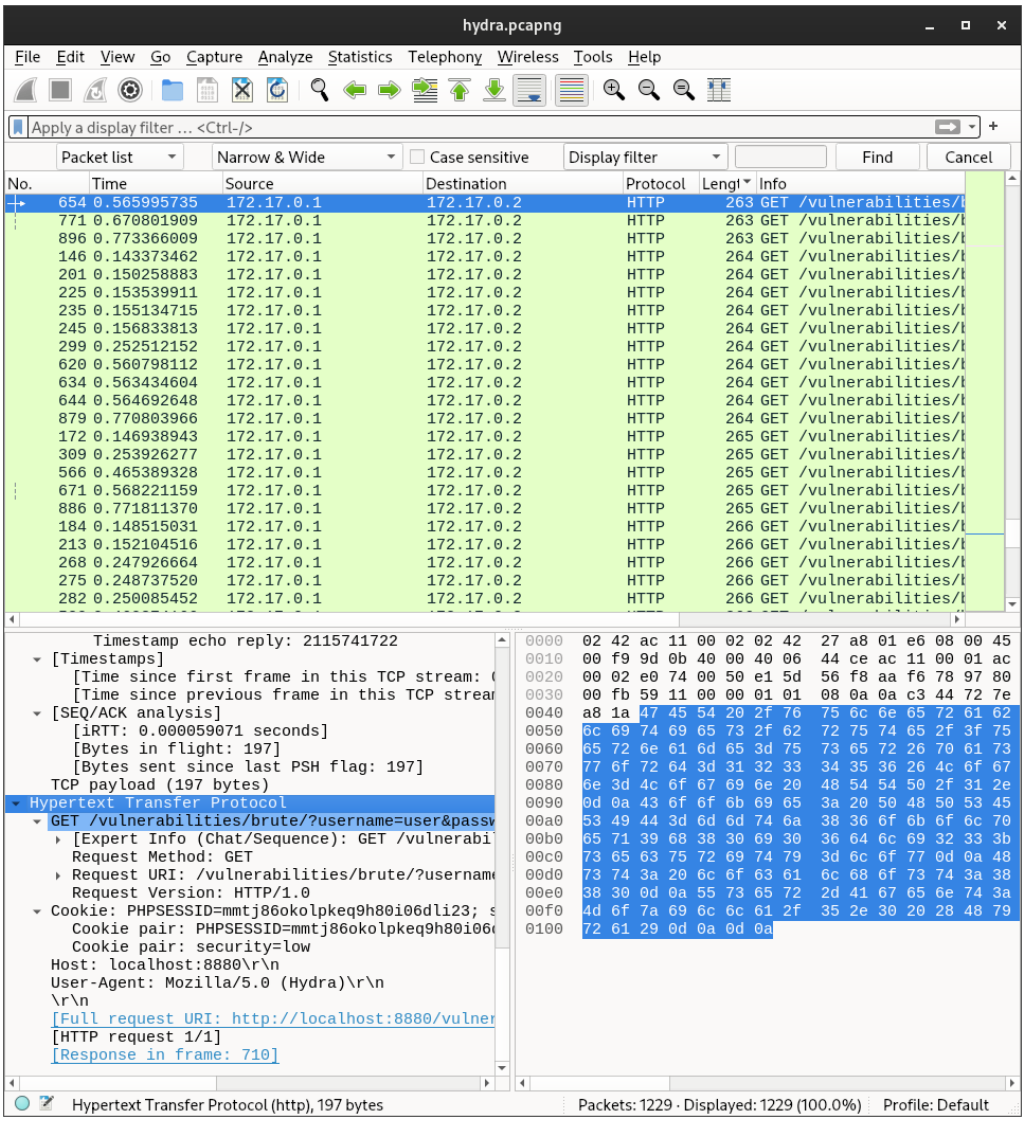

Figura 21: Captura de paquetes durante un ataque de fuerza bruta utilizando Hydra.

Los paquetes capturados en Wireshark evidencian solicitudes HTTP GET que incluyen la URI solicitada, la cual indica la ruta de acceso al servidor, y las cookies correspondientes (que se pueden ver como 'PHPSESSID'), que muestra que Hydra intenta mantener la sesión entre intentos. Además, estas solicitudes revelan el nombre de usuario y la contraseña que se están probando en el proceso de autenticación.

Al igual que en capturas anteriores, se observan detalles adicionales, como la longitud de los paquetes, marcas de tiempo, números de secuencia, y otros metadatos relevantes para el análisis del tráfico de red.

2.16. Mención de las diferencias (tráfico)

Dentro de las diferencias que se encuentran entre estas capturas de tráfico, se pueden observar las siguientes:

1. **Puertos de origen**: Todos utilizan un puerto distinto, siendo estos los puertos 57822, 40066 y 43216 correspondientes a Hydra, BurpSuite y cURL respectivamente.

2. **User-Agents**: Los agentes de usuario entre estos también difieren, ya que, en Hydra se puede observar que el User-Agent corresponde a 'Mozilla/5.0 (Hydra)', en BurpSuite el User-Agent corresponde a 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6312.122 Safari/537.36' y el de cURL es 'Mozilla/5.0 (X11; Linux x86-64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36'.

3. **Versión HTTP**: Mientras que en Hydra y cURL utilizan la versión HTTP/1.0, BurpSuite hace uso de la versión HTTP/1.1.

4. **Largo del paquete**: Se puede ver una diferencia notable entre el largo de los paquetes. Por ejemplo, en esta captura se puede notar que el largo de los paquetes capturados en Hydra varían entre 263 a 275 bytes, en cambio, en BurpSuite, estos van de 899 a 907 bytes.

2.17. Detección de SW (tráfico)

Para detectar el tráfico de DVWA es necesario utilizar la herramienta de Wireshark y navegar por la ruta. Siguiendo con esto, se inicio sesión con las credenciales válidas y se capturó el tráfico de red del software.

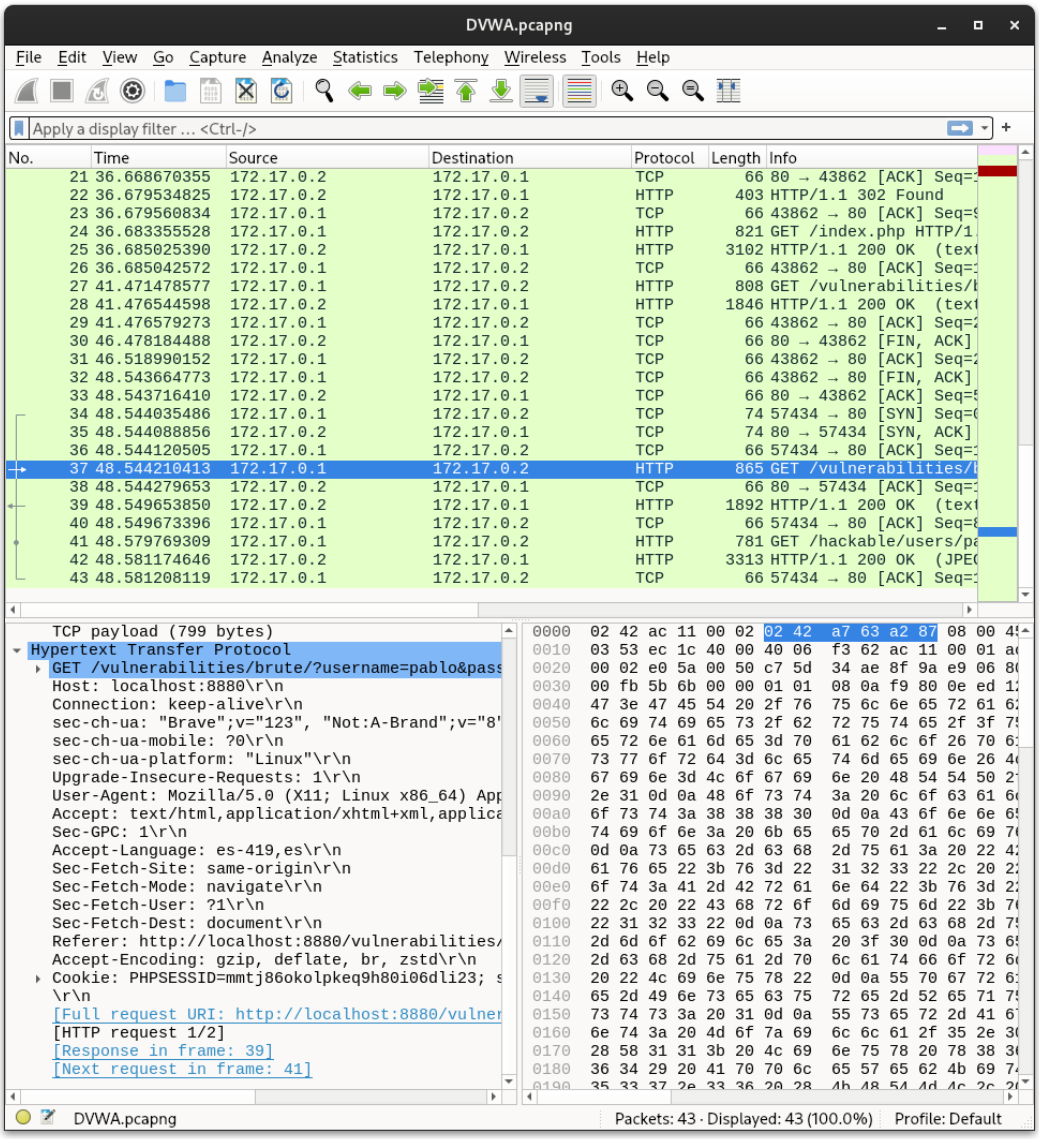

Figura 22: Captura de paquetes durante el uso de DVWA.

La Figura 22 muestra una sección de la captura de tráfico, que incluye paquetes HTTP y TCP. Dentro de estos paquetes, es posible identificar las cookies, el agente de usuario (User-Agent), las flags TCP y otros elementos. En los paquetes asociados con la autenticación, se revelan las credenciales ingresadas junto con una respuesta HTTP 200 OK, lo que confirma la validez de las credenciales y, desafortunadamente, también su exposición dentro del tráfico capturado.

Conclusiones y comentarios

En este informe se abordaron diversas herramientas, análisis y configuraciones, tales como Docker para levantar el entorno de DVWA, BurpSuite para la interceptación del tráfico y ataque por fuerza bruta, Hydra para realizar ataques por fuerza bruta atomatizador, cURL para simular solicitudes y Wireshark para capturar tráfico y posteriormente hacer el análisis de estos.

A través del levantamiento de Docker, se pudo poner en práctica las técnicas de ataque por fuerza bruta sin riesgos para estructuras u organizaciones reales. El uso de las herramientas descritas, fueron escenciales para el entendimiento del flujo de datos entre el cliente y el servidor, además, analizando la actividad, se resalta la importancia de tener contraseñas robustas para evitar el acceso no autorizado. En conclusión, la experimentación en un entorno controlado y el uso y estudio de estas herramientas, son primordiales para entender y mejorar las defensas en los sistemas frente a un ataque.