Informe Laboratorio 2

Sección 03

Anselmo Pacheco
e-mail: anselmo.pacheco@mail.udp.cl

13 de septiembre 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?
- Desarrolle un script en Python para realizar un ataque de fuerza bruta:
  - Utilice la librería requests para interactuar con el formulario ubicado en vulnerabilities/brute y desarrollar su propio script de fuerza bruta en Python. El script debe realizar intentos de inicio de sesión probando una lista de combinaciones de usuario/contraseña.
  - Identifique y explique la cabecera HTTP que empleará para realizar el ataque de fuerza bruta.
  - Muestre el código y los resultados obtenidos (al menos 2 combinaciones válidas de usuario/contraseña).
  - Compare el rendimiento de este script en Python con las herramientas Hydra, Burpsuite, y cURL en términos de velocidad y detección.
- Investigue y describa 4 métodos comunes para prevenir o mitigar ataques de fuerza bruta en aplicaciones web:
  - Para cada método, explique su funcionamiento, destacando en qué escenarios es más eficaz.

2. Desarrollo de actividades según criterio de rúbrica

2.1. Levantamiento de docker para correr DVWA (dvwa)

El proceso realizado para el levantamiento de docker para ejecutar DVWA en el navegador corresponde al siguiente comando que permite la ejecución de docker:
sudo docker run –rm -it -p 80:80 vulnerables/web-dvwa

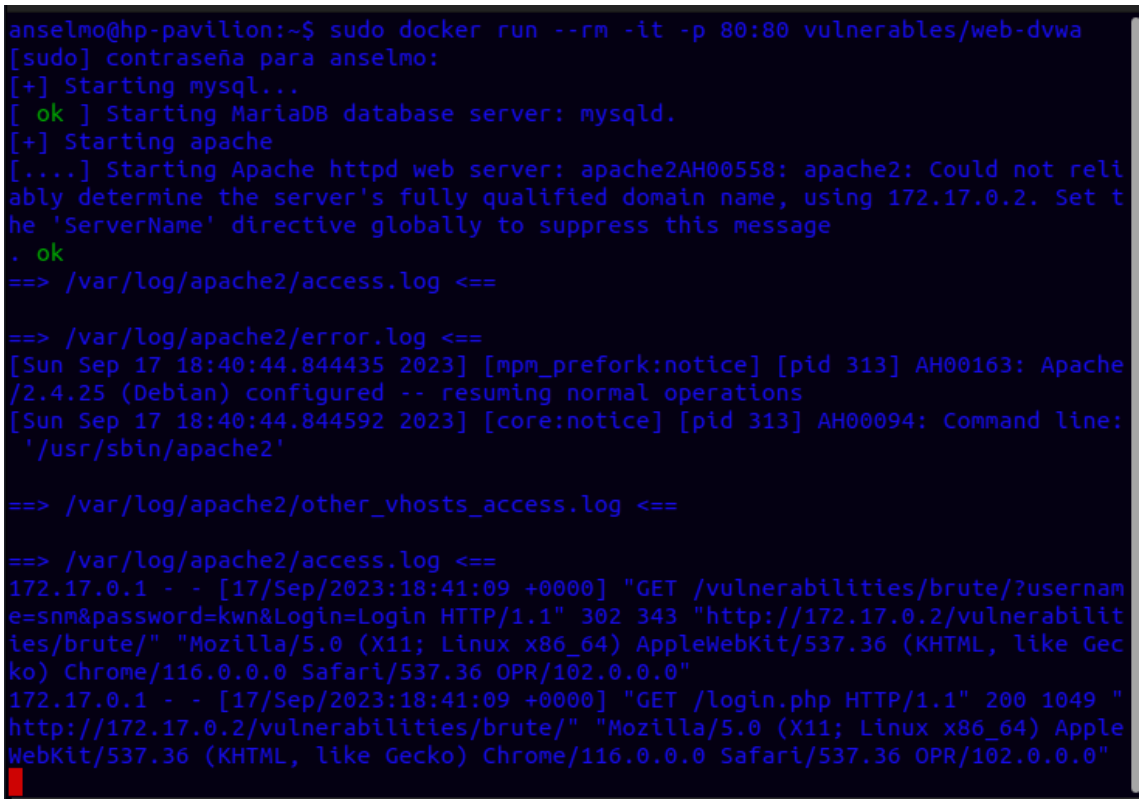

Figura 1: Levantamiento docker.

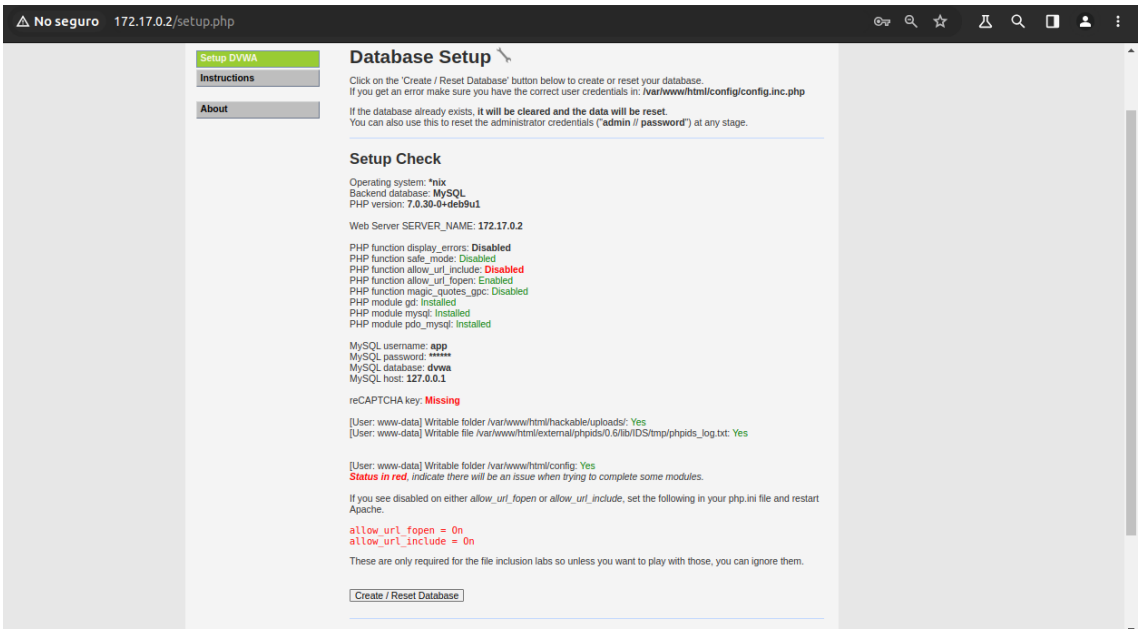

Figura 2: Ejecución comando.

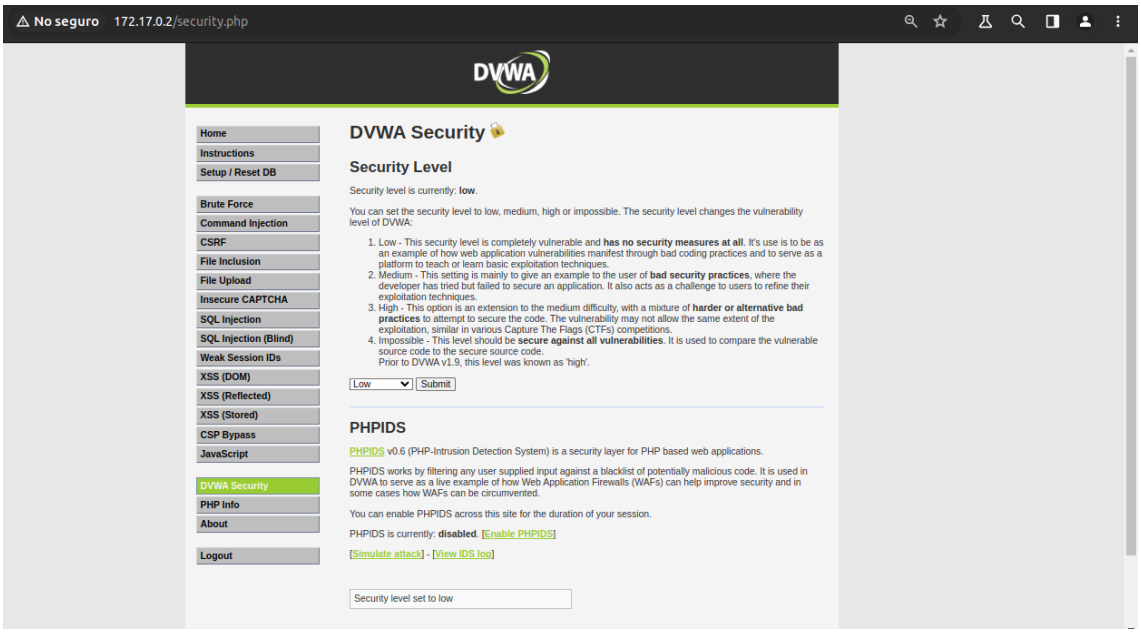

Figura 3: Configuración DVWA.

2.2. Redirección de puertos en docker (dvwa)

Lo que está realizando el comando sudo docker run –rm -it -p 80:80 vulnerables/web-dvwa es que se está creando un contenedor Docker utilizando la imagen vulnerables/web-

dvwa y mapeando el puerto 80 del contenedor al puerto 80 del host. Lo que permite que el tráfico dirigido al puerto 80 sea dirigido al puerto 80 del contenedor docker. Y de esta manera poder acceder a DVWA (Damn Vulnerable Web Application) deste el navegador web.
Para detallar de mejor manera lo que realiza el comando se procederá a explicar paso a paso lo que realiza cada sentencia:
sudo: Este comando permite ejecutar el comando que sigue con privilegios de administrador. Puede ser necesario para ejecutar comandos Docker, dependiendo de la configuración.

docker: Es el comando principal para interactuar con el sistema de contenedores Docker.
run: Indica que queremos ejecutar un nuevo contenedor a partir de una imagen.
–rm: Esta opción indica a Docker que elimine el contenedor automáticamente después de que se detenga. Esto es útil para evitar la acumulación de contenedores inactivos.
-it: Combina dos opciones:
-i: Permite la interactividad del contenedor, lo que significa que puedes interactuar con él desde tu terminal.
-t: Asigna un pseudo-terminal (TTY) al contenedor. Esto es necesario para la interacción interactiva.
-p 80:80: Esta opción establece un mapeo de puertos entre el sistema host y el contenedor. En este caso, el puerto 80 del host se mapea al puerto 80 del contenedor. Esto significa que cualquier tráfico dirigido al puerto 80 del host se redirige al puerto 80 del contenedor.
vulnerables/web-dvwa: Es el nombre de la imagen del contenedor que se utilizará para crear y ejecutar el contenedor. En este caso, se está utilizando una imagen llamada vulnerables/web-dvwa, que es Damn Vulnerable Web Application (DVWA), una aplicación web intencionadamente insegura para propósitos educativos y de pruebas de seguridad.

Figura 4: DVWA desde navegador web.

2.3. Obtención de consulta a replicar (burp)

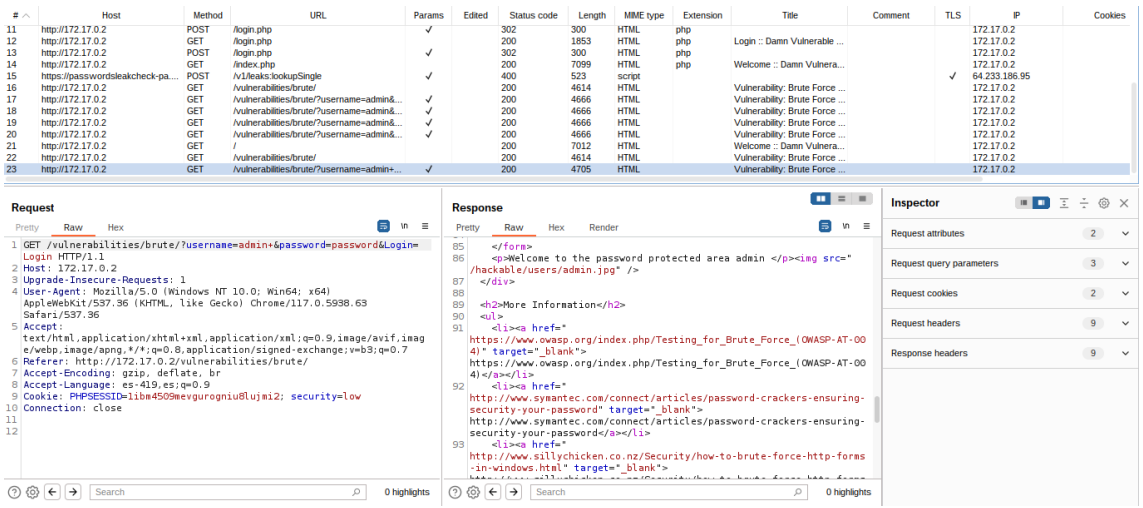

Figura 5: Consulta a replicar

Lo cual permitirá identificar los campos y secciones que se deben observar para el desarrollo del ataque por fuerza bruta.

2.4. Identificación de campos a modificar (burp)

En este punto se proceden a identificar los campos que deben marcarse para luego, al momento de realizar el intruder se permitan reemplazar estos campos seleccionados por el valor del diccionario creado para cada campo.
Los campos seleccionados son: admin y password.

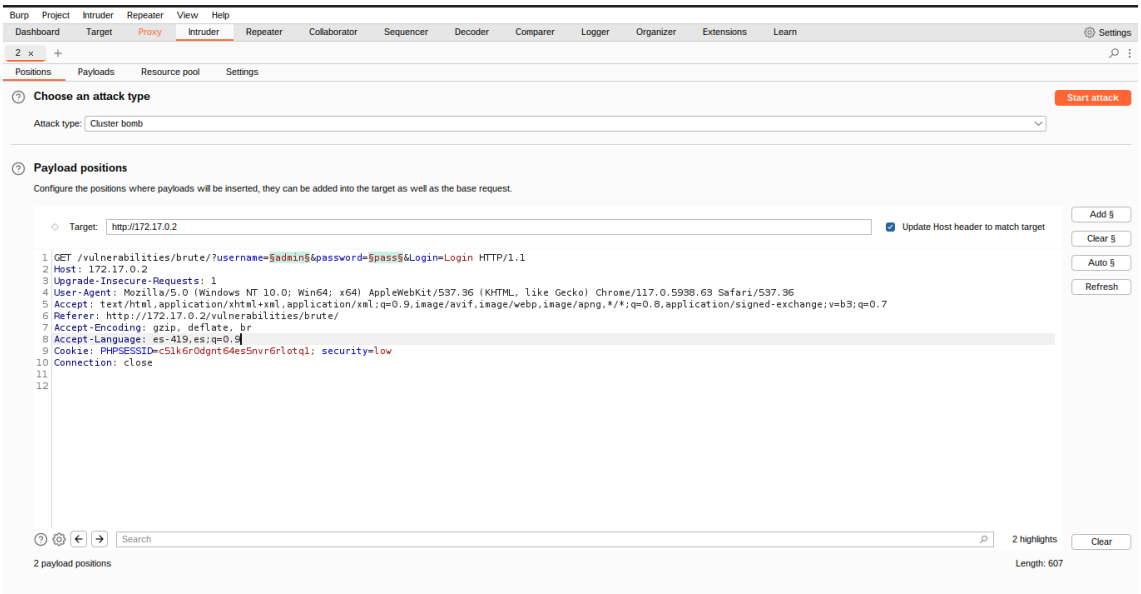

Figura 6: Selección de campos

2.5. Obtención de diccionarios para el ataque (burp)

En este punto se proceden a crear dos archivos que contengan posibles nombres de usuario y contraseña, en donde de esta manera se podrán crear diccionarios que serán utilizados para realizar el ataque de fuerza bruta.
Para realizar este ataque se procederán a reemplazar cada valor del diccionario Usuarios.txt para cada diccionario Contraseñas.txt
Los diccionarios creados son los siguientes:

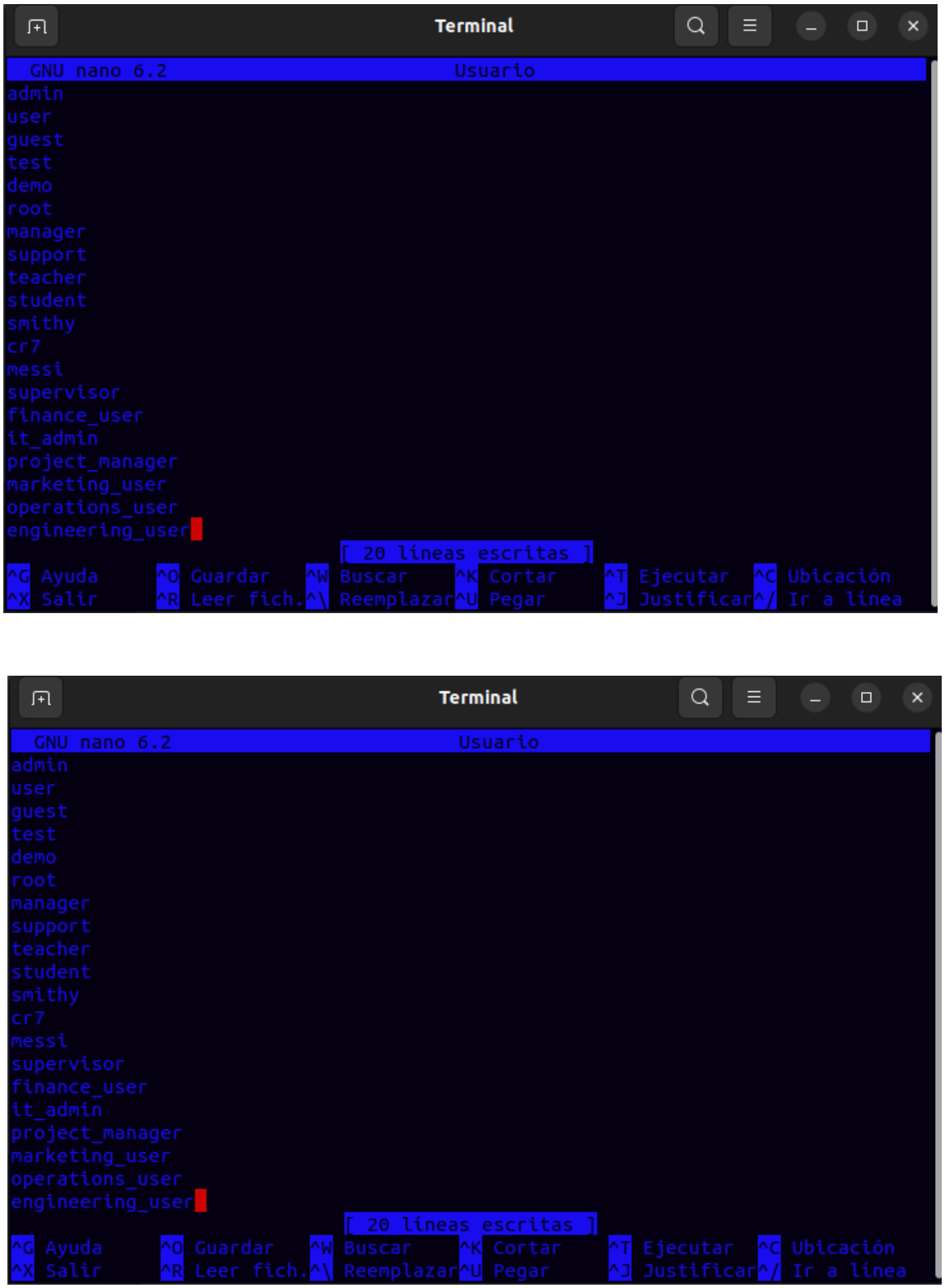

Figura 7: Diccionario Contraseñas

Luego de la creación de usuarios y contraseñas posibles se procederán a cargar los archivos en burp suite para realizar un ataque de fuerza bruta, el cuál será en este caso el ataque Çluster Bomb"que corresponde a una técnica de prueba de fuerza bruta que combina múltiples listas de palabras para generar una lista de contraseñas potenciales. Funciona tomando una lista de palabras y combinándola con otra lista de palabras, generando así todas las combinaciones posibles entre los elementos de ambas listas. Para cada payload se cargarán los datos de los diccionarios para ser reemplazados en los campos seleccionados anteriormente, quedando de la siguiente manera:

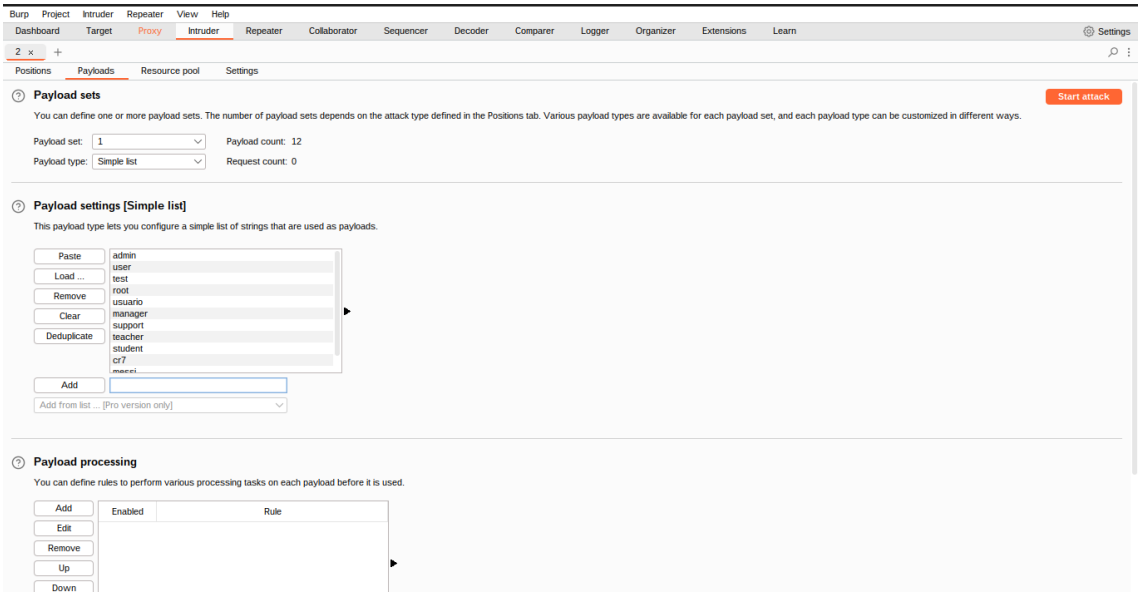

Figura 8: Reemplazo payload usuarios

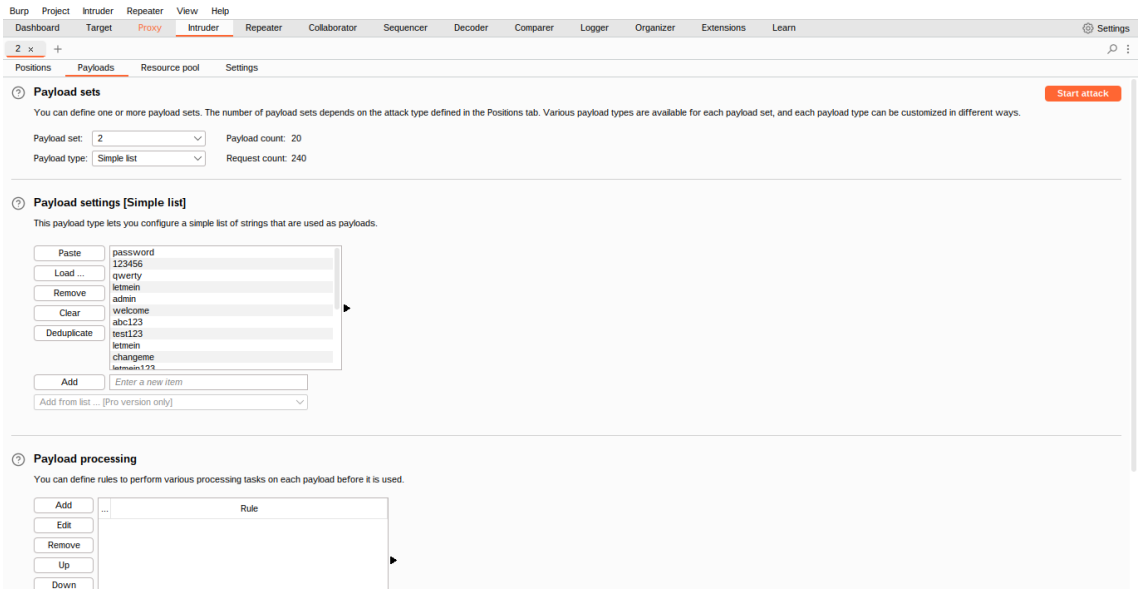

Figura 9: reemplazo payload contraseñas

Una vez realizado el ingreso de los archivos (diccionarios) creados se debe realizar el ataque por fuerza bruta.

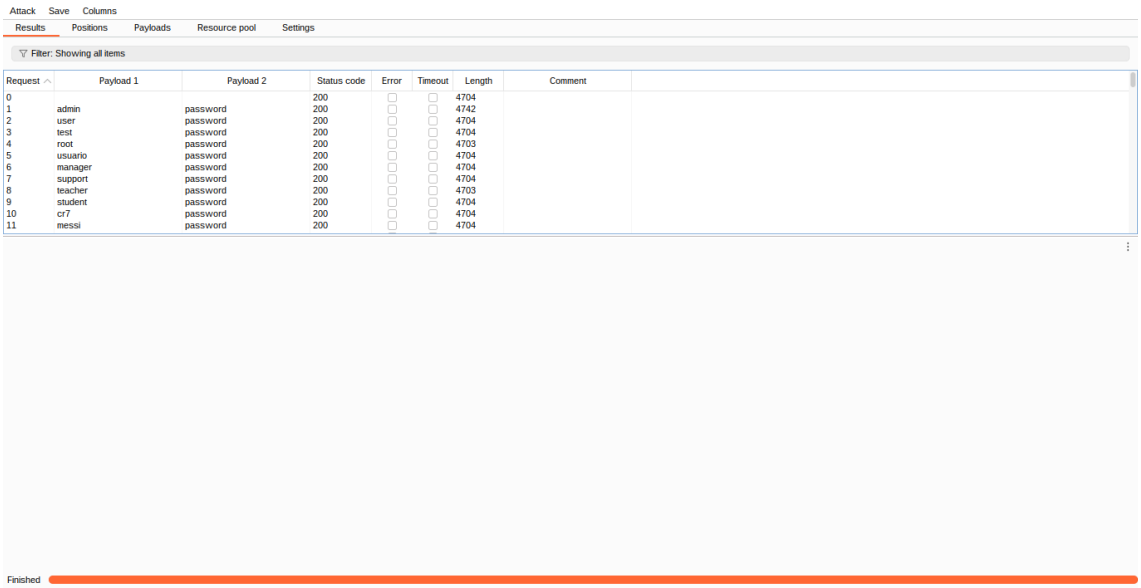

Figura 10: Ataque Cluster Bomb

2.6. Obtención de al menos 2 pares (burp)

Una vez finalizado el ataque se deben a comenzar a revisar parámetros que permitan determinar si el usuario y contraseña realizado en la combinación permite ingresar de manera correcta.
Para este caso se procede a realizar un filtro con la palabra "Welcome"que permite identificar que combinación es bienvenida en el sitio web, en donde se obtuvo lo siguiente:
Se proceden a adjuntar los pares obtenidos.

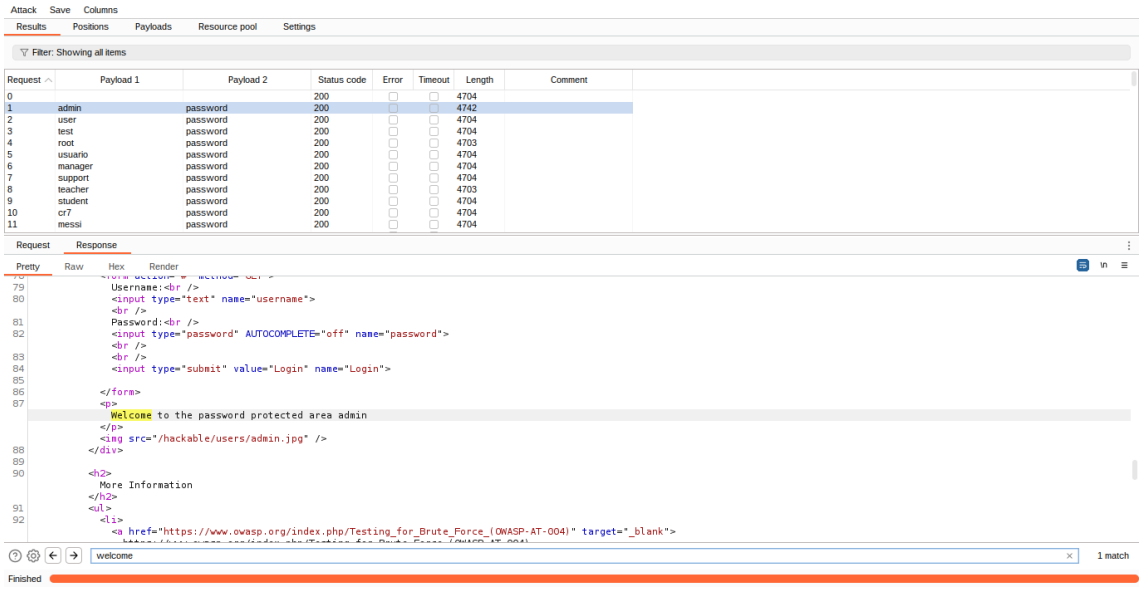

Figura 11: Par válido número 1

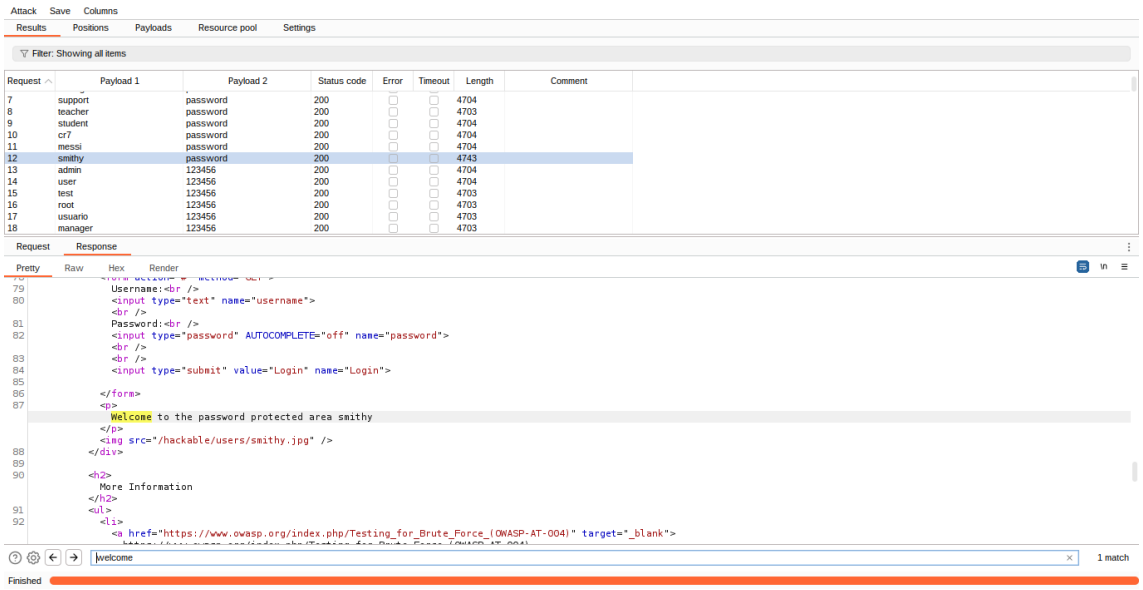

Figura 12: Par válido número 2

Para comprobar que los pares obtenidos son válidos se proceden a insertar en el login de la página y de esta manera comprobar su veracidad, de lo cual se obtuvo lo siguiente:

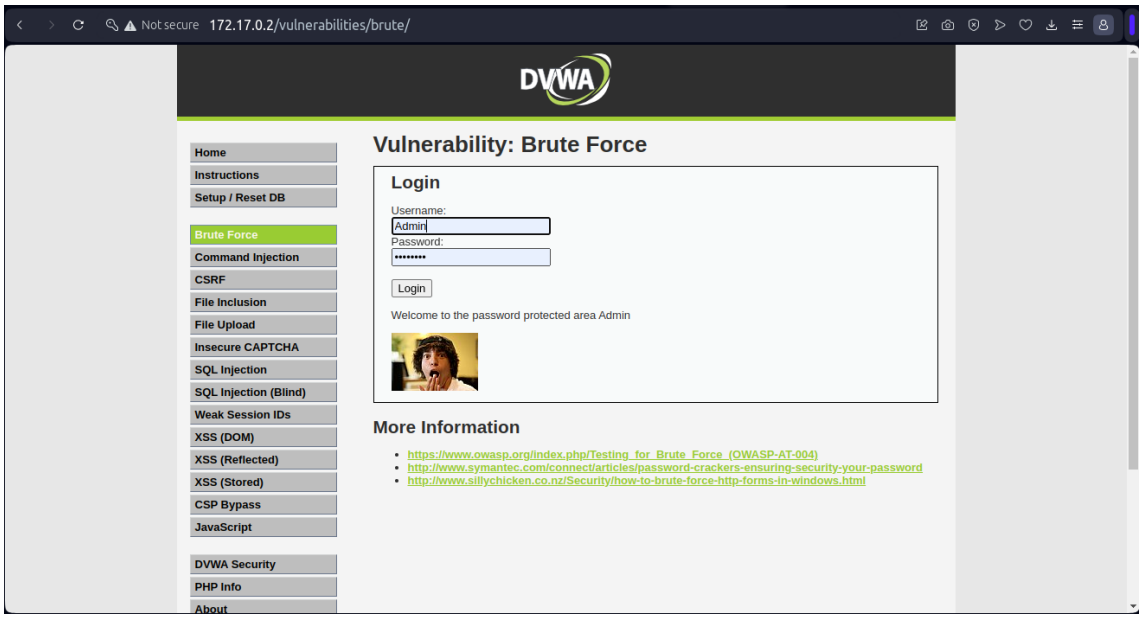

Figura 13: Comprobación número 1

Figura 14: Comprobación número 2

2.7. Obtención de código de inspect element (curl)

En este punto se debe realizar una inspección de elemento al sitio web con el objetivo de determinar el curl a utilizar, al analizar los elementos se logró identificar lo siguiente: Donde se debe realizar una copia de la dirección curl obtenida y de esta manera realizar la solicitud en consola de lo obtenido.

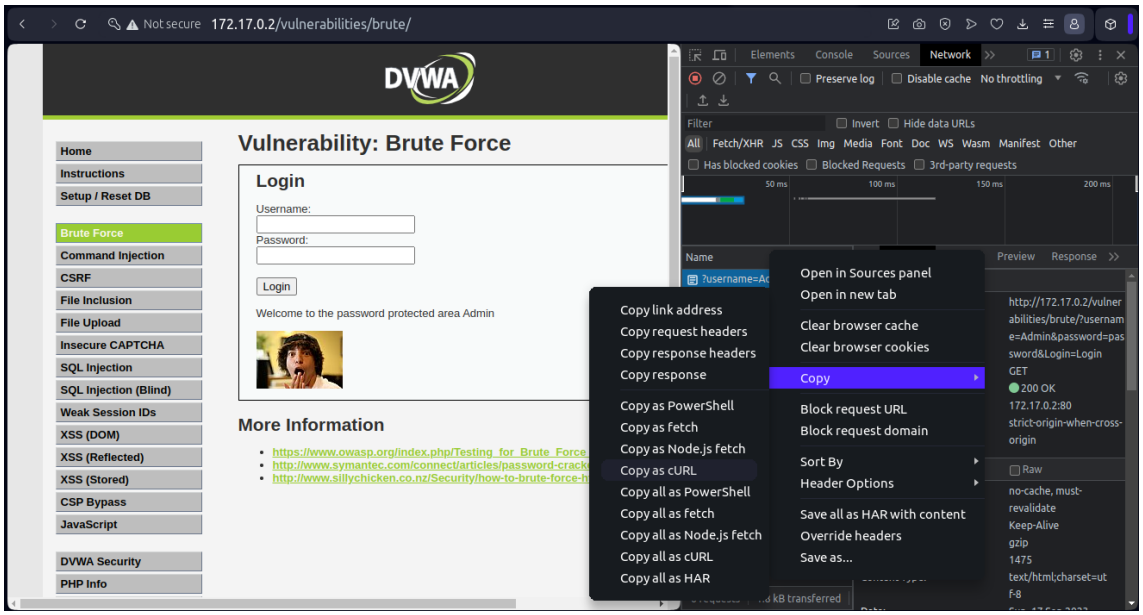

Figura 15: Obtención curl.

2.8. Utilización de curl por terminal (curl)

A partir de lo obtenido anteriormente se realizará la prueba para pares válidos e inválidos con el fin de determinar cuales son las diferencias en las respuestas que se obtienen, la manera en la cuál se realizará lo mencionado es modificar los valores de los campos username y password de manera manual antes de realizar la curl.
Como ya sabemos cuales son usuarios y contraseñas válidos e inválidos se procede a reemplazar, donde se obtuvo lo siguiente:

Figura 16: cURL válido.

Figura 17: cURL inválido.

2.9. Demuestra 4 diferencias (curl)

A continuación se deben mencionar 4 diferencias a partir de las cURL realizadas anteriormente.
Para comenzar se debe mencionar la principal diferencia, que corresponde a la respuesta obtenida al momento de realizar la acción por consola.

Figura 18: cURL aceptado

Figura 19: cURL no aceptado

Otra diferencia que se puede apreciar es el largo del contenido para cada cURL dependiendo su validez.

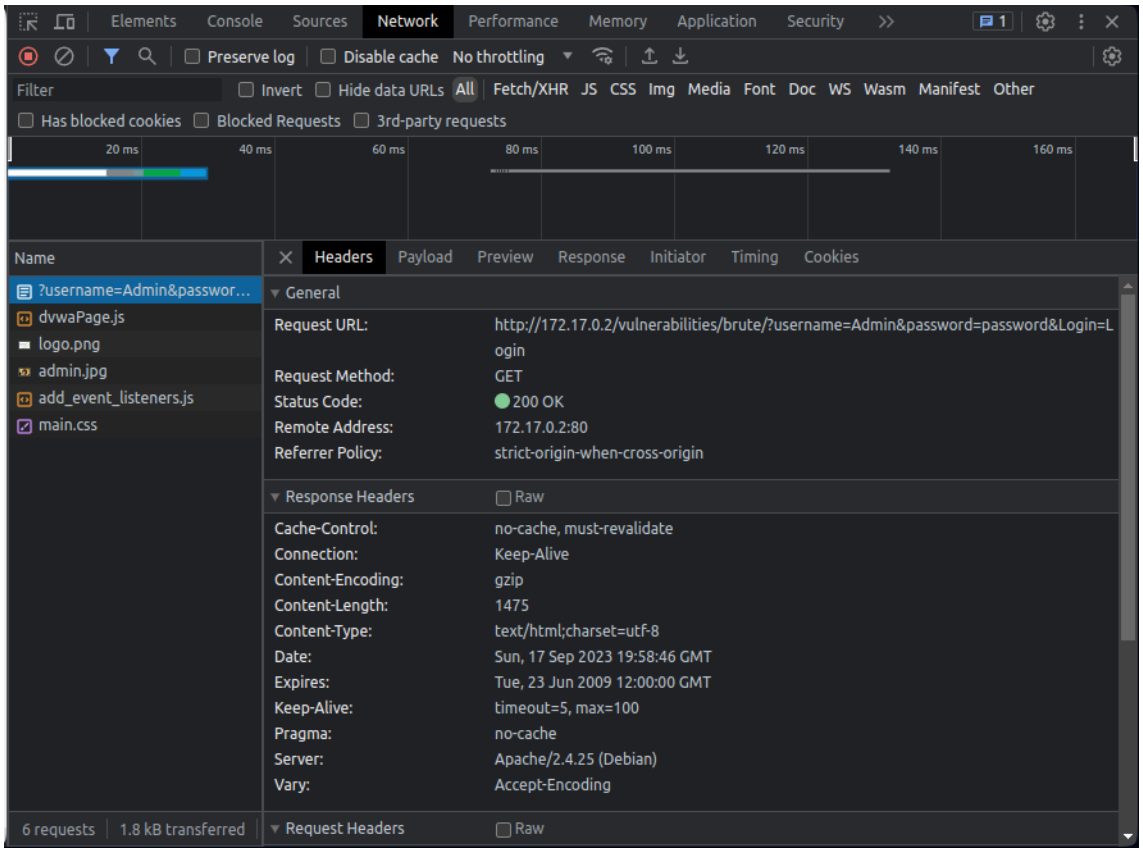

Figura 20: cURL aceptado

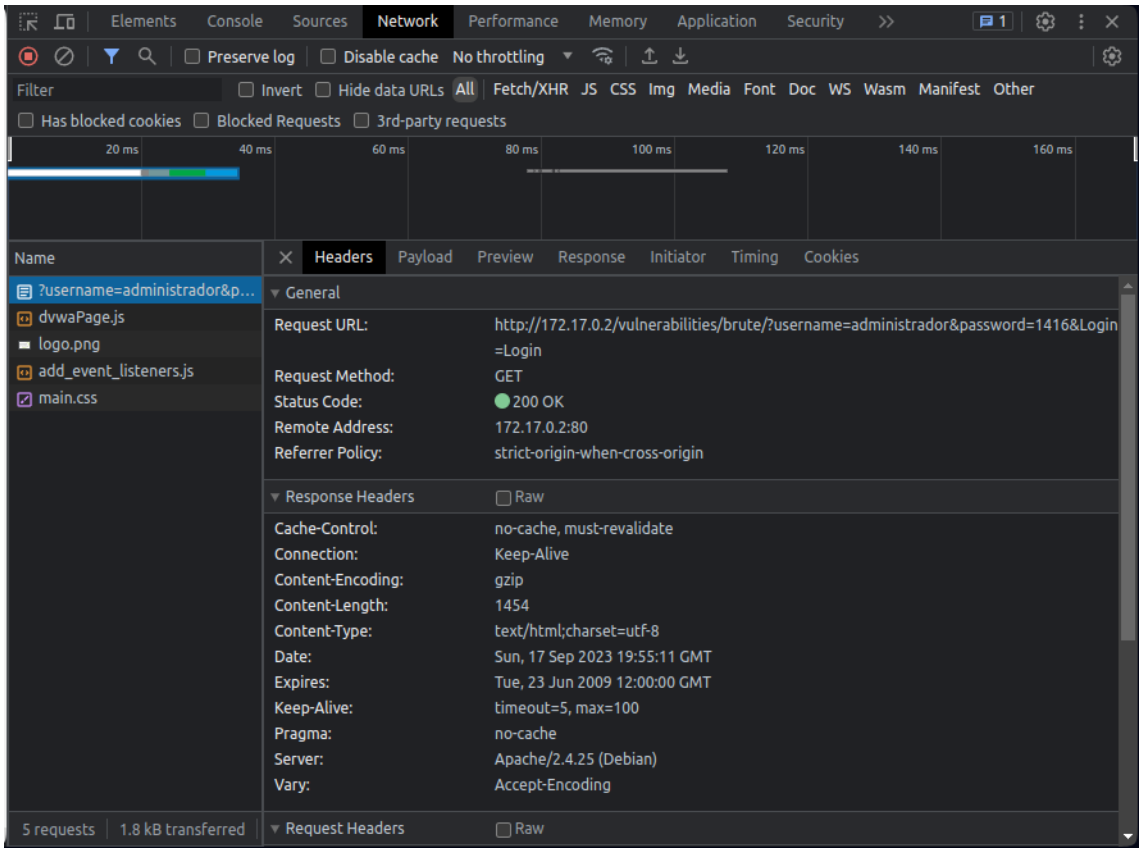

Figura 21: cURL no aceptado

En las imágenes adjuntas anteriormente se puede notar que si se trata de un usuario y contraseña válidos aparece la imagen admin.jpg en las cookies. Lo que permite evidenciar una diferencia entre ambas.

Otra diferencia que puede notarse es que al momento de redireccionar cuando se trata de un usuario es diferente a cuando no lo es. Tal como se aprecia a continuación:

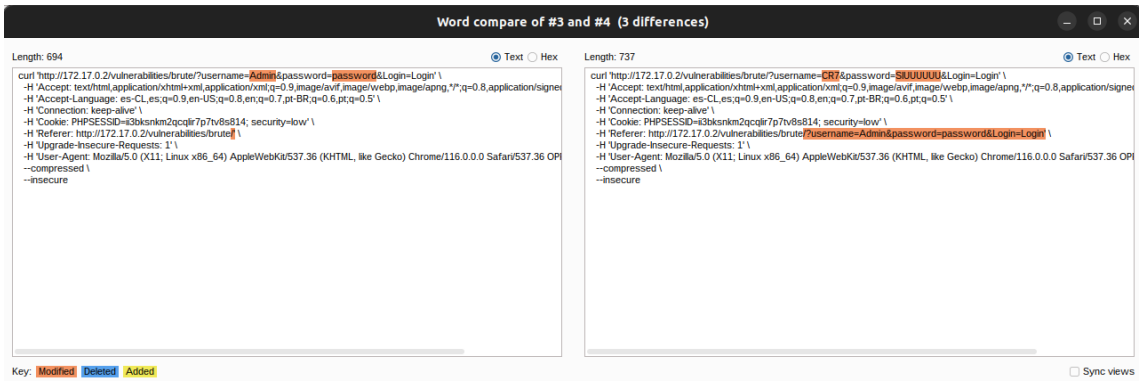

Figura 22: Comparación

Tal como se ve en la imagen se puede notar otra diferencia que sería el largo de caracteres de respuesta a partir de la validez del usuario.

2.10. Instalación y versión a utilizar (hydra)

En este apartado se debe realizar la instalación del software Hydra, la cual se realiza mediante el comando sudo apt install hydra.
Para determinar la versión utilizada se utiliza el comando hydra -version, el cual permitió determinar que la versión corresponde a Hydra v9.2 (c) 2021 by van Hauser/THC and David Maciejak.

2.11. Explicación de comando a utilizar (hydra)

A continuación se realizará una explicación de la función de cada parte del comando de acuerdo a sus funciones y luego se exolicará lo que realiza en conjunto esta herramienta de fuerza bruta.
hydra: Este es el comando principal para ejecutar la herramienta Hydra, que es una herramienta de prueba de fuerza bruta.

172.17.0.2: Esta es la dirección IP del objetivo que se va a atacar.

-m /vulnerabilities/brute/http-get-form.php: Esta opción -m especifica el módulo de autenticación a utilizar. En este caso, se está utilizando un módulo HTTP GET para la autenticación.
http-form-get /vulnerabilities/brute/:username=USER&password=PASS&Login=Login:incorrect
PHPSESSID=ii3bksnkm2qcqlir7p7tv8s814; security=low": Esta es la cadena que describe la forma en que se va a realizar el ataque. Se indica cómo debe ser la solicitud

HTTP para autenticarse. Explicado detalladamente:

http-form-get: Indica que el método HTTP que se utilizará para la autenticación es GET.
/vulnerabilities/brute/:username=USER&password=PASS&Login= Login:incorrect: H=Cookie: PHPSESSID=ii3bksnkm2qcqlir7p7tv8s814; security=low": Esta es la cadena de solicitud HTTP que se enviará. Es una URL con los campos username y password, y un campo Login que indica que se está realizando un intento de inicio de sesión.
También especifica qué respuesta indica un intento fallido (:incorrect:).
Por último, se incluye una cabecera Cookie que establece ciertas variables de sesión, como PHPSESSID y security.
-L Usuarios.txt: Especifica el archivo Usuarios.txt que contiene la lista de nombres de usuario que se probarán.

-P Contraseñas.txt: Especifica el archivo Contraseñas.txt que contiene la lista de contraseñas que se probarán.

A grandes rasgos este comando se utiliza para realizar un ataque de fuerza bruta utilizando una solicitud HTTP GET contra un objetivo específico (172.17.0.2) que utiliza un formulario de inicio de sesión. Se prueba una lista de nombres de usuario (Usuarios.txt) y una lista de contraseñas (Contraseñas.txt). Se define una solicitud HTTP personalizada y se especifica cómo interpretar las respuestas para determinar si un intento de inicio de sesión fue exitoso o no.

2.12. Obtención de al menos 2 pares (hydra)

Al momento de realizar la ejecución del comando mencionado anteriormente se obtiene la siguiente respuesta, que corresponde a la búsqueda de usuarios y contraseña válidos encontrados en los diccionarios proporcionados para realizar el inicio utilizando fuerza bruta.

```
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-09-17 17:40:
11
[DATA] max 16 tasks per 1 server, overall 16 tasks, 400 login tries (l:20/p:20),
 ~25 tries per task
[DATA] attacking http-get-form://172.17.0.2:80/vulnerabilities/brute/:username=^
USER^&password=^PASS^&Login=Login:incorrect:H=Cookie: PHPSESSID=ii3bksnkm2qcqlir
7p7tv8s814; security=low
[80][http-get-form] host: 172.17.0.2   login: admin   password: password
[80][http-get-form] host: 172.17.0.2   login: smithy   password: password
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-17 17:40:
17
anselmo@hp-pavilion:~/Descargas$
```

Figura 23: Usuarios registrados

2.13. Explicación paquete curl (tráfico)

Su tráfico puede variar según el comando y la URL que se esté utilizando. No está diseñado específicamente para pruebas de fuerza bruta, por lo que su tráfico puede ser más bajo en comparación con Hydra.

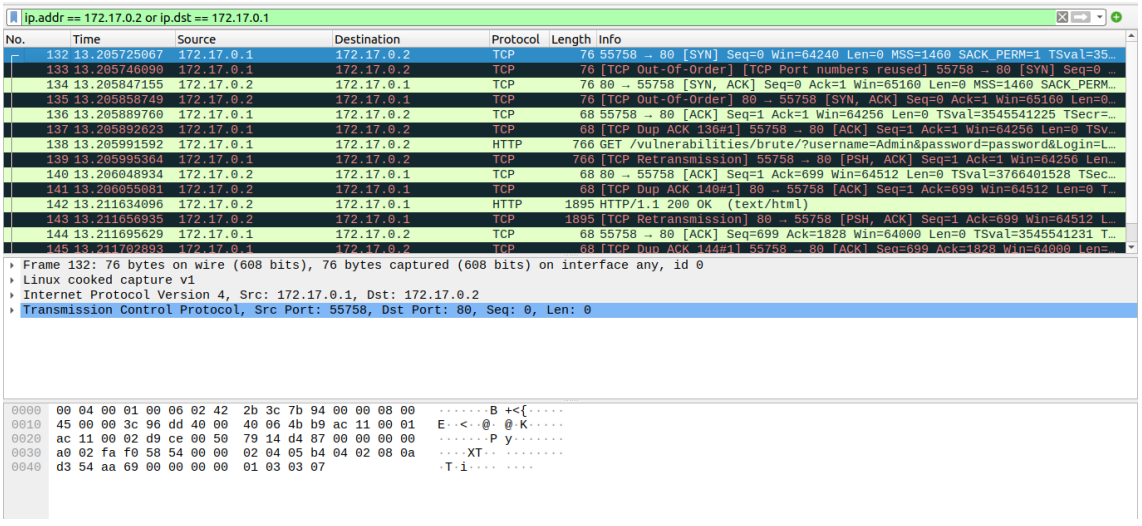

Figura 24: Tráfico cURL

2.14. Explicación paquete burp (tráfico)

Con esta herramienta se genera tráfico a partir de las solicitudes que se van realizando y a partir de esto se van determinando las respuestas, es decir, para este caso se puede notar que cuando se trata de un usuario y/o contraseña incorrecto se envían paquetes tcp y al ser un par válido se envía la solicitud HTTP.

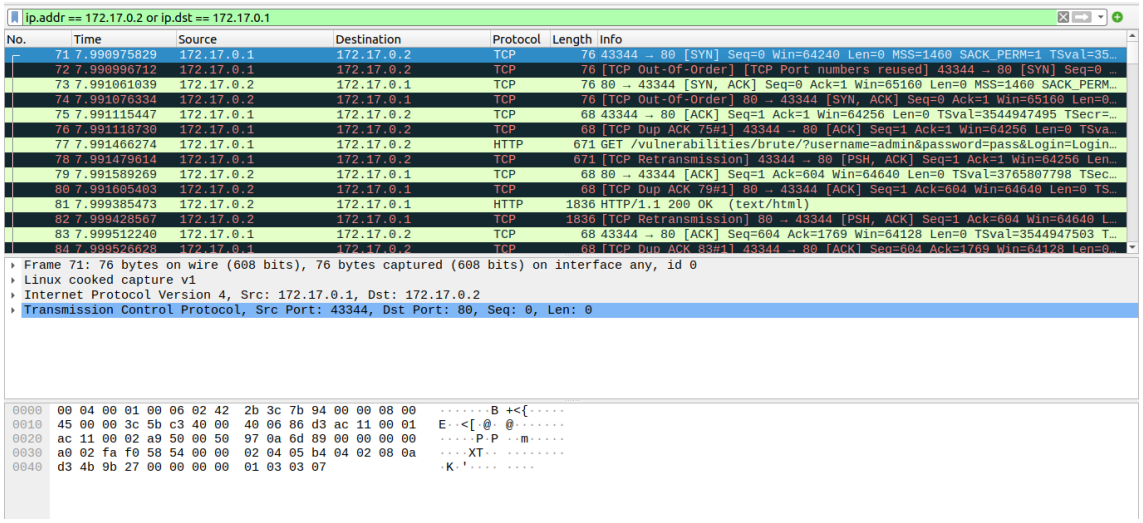

Figura 25: Tráfico burp

2.15. Explicación paquete hydra (tráfico)

En esta captura de tráfico se puede apreciar una gran cantidad de tráfico correspondiente al protocolo tcp, que pudiera darse debido a los ataques de fuerza bruta. Es decir, se intentará iniciar sesión repetidamente con diferentes combinaciones de nombres de usuario y contraseñas.

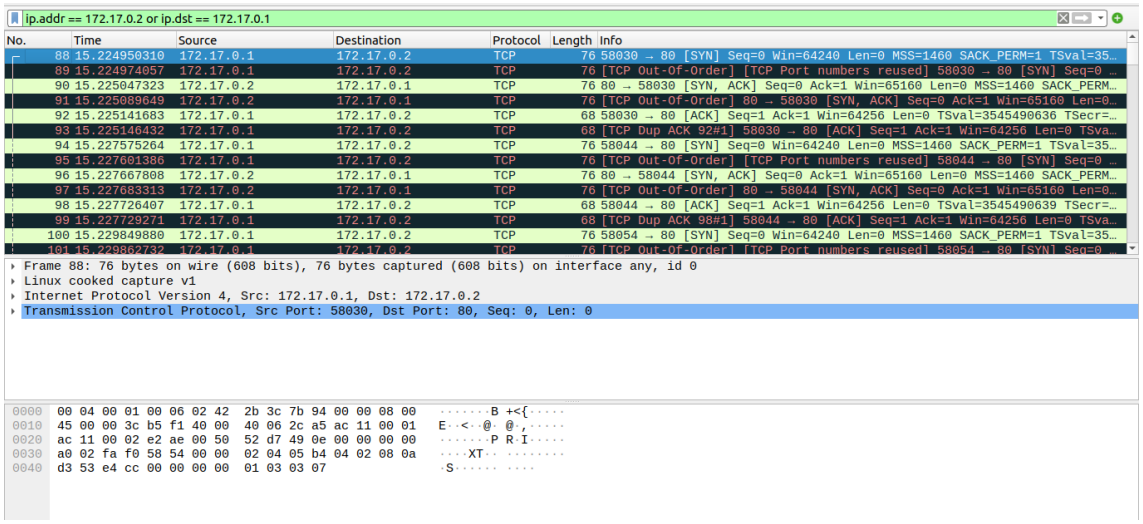

Figura 26: Tráfico hydra

2.16. Mención de las diferencias (tráfico)

Las diferencia que se pudieron notar entre las herramientas es la cantidad de paquetes generados a partir de la ejecución de las actividades realizadas, debido a que en hydra se generaron una mayor cantidad de paquetes en comparación a las otras herramientas, principalmente del protocolo TCP.
Lo que puede producir diferencias entre estas herramientas son las configuraciones y opciones específicas que se utilicen en cada herramienta, ya que estas pueden influir en la cantidad y el tipo de tráfico generado.

2.17. Detección de SW (tráfico)

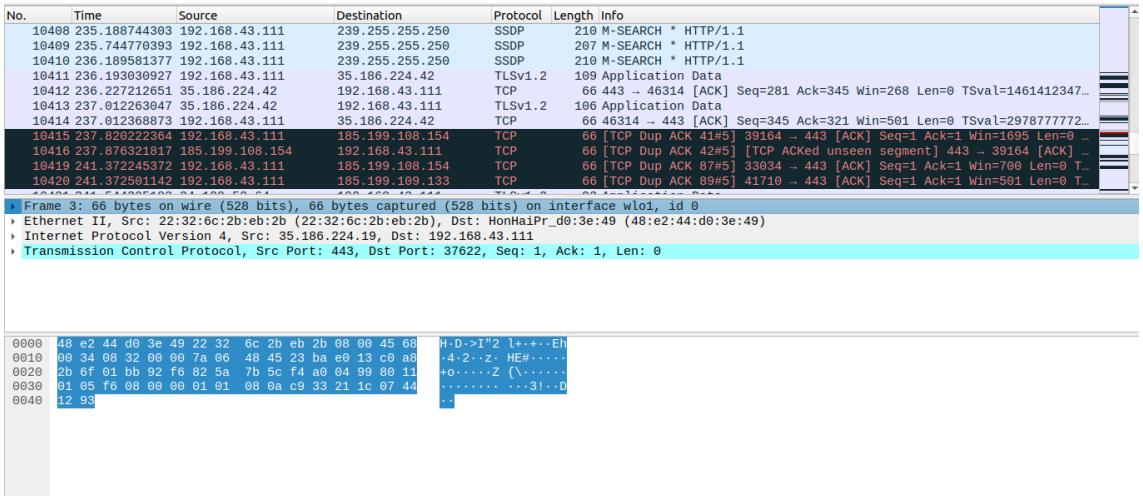

Figura 27: Tráfico SW

2.18. Interacción con el formulario (python)

2.19. Cabeceras HTTP (python)

2.20. Obtención de al menos 2 pares (python)

2.21. Comparación de rendimiento con Hydra, Burpsuite, y cURL (python)

2.22. Demuestra 4 métodos de mitigación (investigación)

- **Limitación de intentos de inicio de sesión**: Este método consiste en restringir el número de intentos fallidos de inicio de sesión permitidos en un período de tiempo. Si un usuario excede un número determinado de intentos fallidos (por ejemplo, 5 intentos), la cuenta puede ser bloqueada temporalmente o se requiere verificación adicional (como un CAPTCHA o autenticación por correo electrónico). Esto dificulta que los atacantes puedan realizar un gran número de intentos en poco tiempo.
- **Autenticación multifactor (MFA)**: La autenticación multifactor requiere que los usuarios verifiquen su identidad mediante múltiples formas de autenticación (algo que saben, como una contraseña, y algo que tienen, como un token de autenticación o un código enviado por SMS). Esto agrega una capa adicional de seguridad, ya que incluso si un atacante logra obtener las credenciales, necesitaría el segundo factor para acceder a la cuenta.
- **CAPTCHA**: Los CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) se utilizan para verificar que el intento de acceso está siendo realizado por un humano y no por un bot automatizado. Se pueden implementar después de un cierto número de intentos fallidos de inicio de sesión para prevenir ataques automatizados de fuerza bruta. Los CAPTCHA pueden ser tradicionales (con imágenes o texto) o invisibles (detectar comportamiento humano en el sitio).
- **Hashing y salting de contraseñas**: Almacenar contraseñas de manera segura es crucial para prevenir ataques de fuerza bruta offline. Las contraseñas deben ser *hasheadas* usando algoritmos robustos como bcrypt o Argon2, y acompañadas de un *sal* (un valor aleatorio añadido a la contraseña antes de hashearla). Esto hace que incluso si un atacante obtiene acceso a la base de datos, las contraseñas en texto claro no sean accesibles, y un ataque de fuerza bruta sea menos efectivo debido al costo computacional que implica hashear cada posible combinación.

Conclusiones y comentarios

De esta actividad se puede concluir que cada herramienta genera tráfico de manera diferente debido a su funcionalidad y propósito. Por ejemplo, Hydra generará tráfico de fuerza bruta intenso, mientras que cURL y Burp Suite están diseñados para propósitos más específicos. Como es el caso de burp suite que se utiliza para protocolo Http.
Por otra parte se puede notar que las técnicas de fuerza bruta son poderosas. En este caso fueron útiles para probar la robustez de contraseñas y que los HTTP se pueden vulnerar de manera sencilla utilizando las herramientas y técnicas empleadas en esta experiencia.