Informe Laboratorio 2

Sección 02

Ignacio Guajardo
e-mail: ignacio.guajardo_m@mail.udp.cl

Septiembre de 2023

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

2.1. Levantamiento de docker para correr DVWA (dvwa)

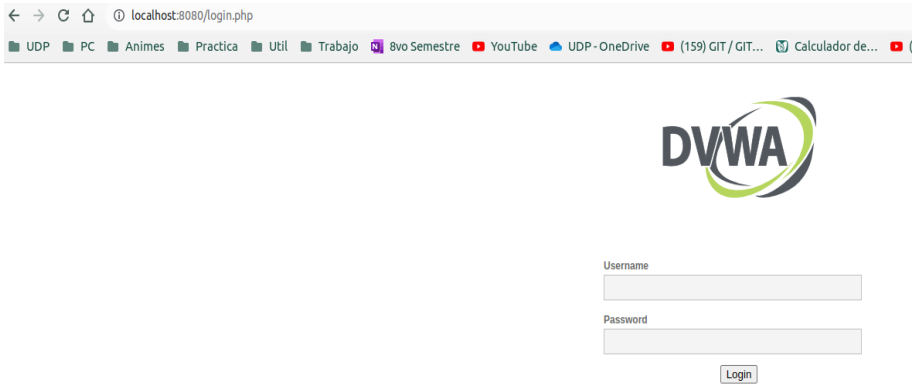
Para comenzar la actividad, en primer lugar, se llevó a cabo la instalación de Docker y la configuración de la imagen de DVWA, siguiendo detenidamente las indicaciones proporcionadas aqui.
Al ejecutar dicho comando en la terminal, se obtiene lo siguiente:

Lo cuál indica cómo se inicializa la base de datos Mysql y el servidor apache, lo cuál permite entrar a la aplicación web. También indica la dirección IP que usará la aplicación web, esta IP será útil posteriormente.

2.2. Redirección de puertos en docker (dvwa)

El comando previo se encarga de iniciar la imagen y configurar la redirección de puertos. La opción '-p' se utiliza para este propósito, y en el comando se especifica '-p 8080:80', lo que implica que estamos enlazando el puerto 8080 de nuestra máquina con el puerto 80 del contenedor.

Una vez que la imagen está en funcionamiento, simplemente debemos abrir el navegador y acceder a ese puerto para visualizar la aplicación.



2.3. Obtención de consulta a replicar (burp)

Para obtener la solicitud que deseamos replicar en Burp Suite, el primer paso es acceder a la página a través del navegador integrado en la aplicación. Para hacerlo, debemos activar la función de interceptación en la aplicación.

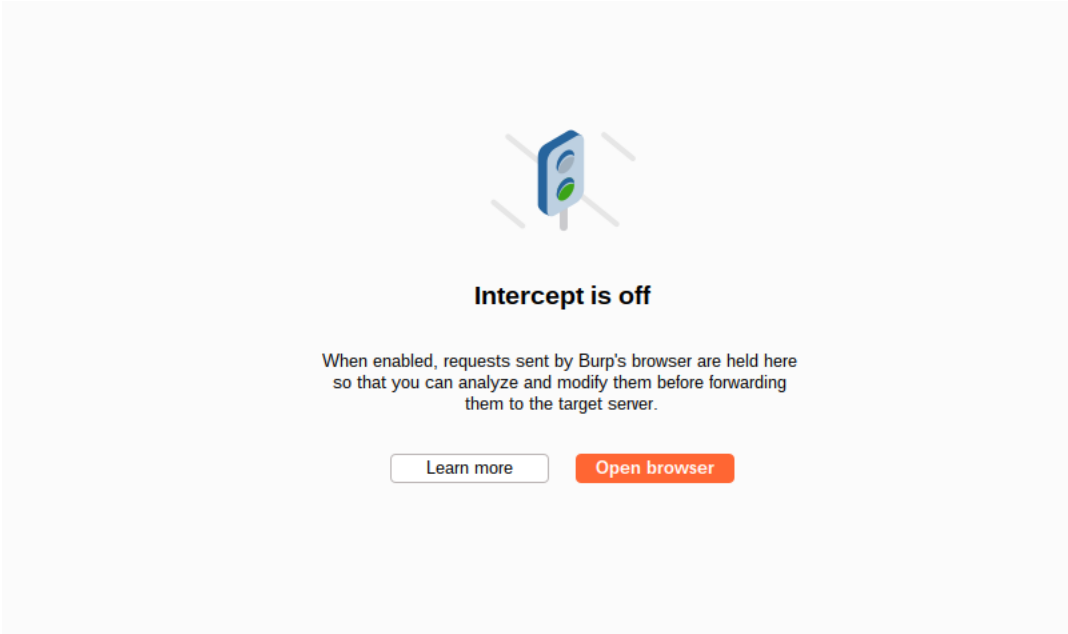
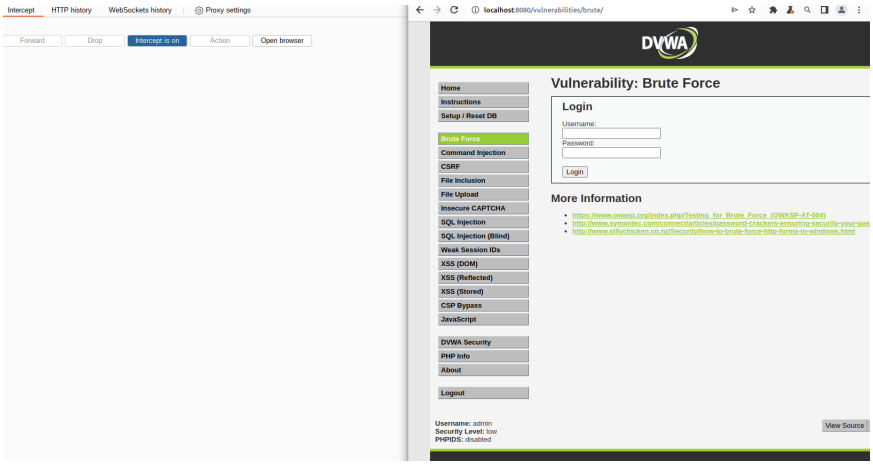


Figura 1: Se inicializa la aplicación burpsuite

Cuando activamos esta opción, se abrirá un navegador dentro de la aplicación, y a través de este navegador, debemos visitar la página que deseamos interceptar. Luego, realizamos una consulta similar a la que planeamos atacar, pero de manera manual.



Una vez que hayamos completado estos pasos, la solicitud GET que hemos realizado aparecerá en el menú 'Target' de Burp Suite. Una vez identificada, debemos seleccionar ese elemento y luego elegir la opción 'Send to Intruder'.

2.4. Identificación de campos a modificar (burp)

Una vez que hayamos completado los pasos previos, el siguiente paso implica la identificación de los campos que deseamos ajustar en cada intento. En este escenario, si pretendemos probar con diversos usuarios y contraseñas, debemos identificar y señalar específicamente estos dos campos en la solicitud. De esta manera, Burp Suite los considerará en los ataques de fuerza bruta o en las pruebas de intrusión.

2.5. Obtención de diccionarios para el ataque (burp)

Después de haber identificado los campos que queremos modificar en la solicitud, el siguiente paso consiste en adquirir dos diccionarios: uno con nombres de usuario y otro con contraseñas. Estos diccionarios serán empleados como parte del payload en nuestro ataque. La decisión de combinar dos diccionarios se fundamenta en la idea de que esta estrategia podría aumentar la eficacia de nuestro ataque.

Continuando el ítem anterior, se deben tener algunas listas de valores para estas listas simples, y poder realizar el ataque. Para esto se probará, en primera instancia, con credenciales comunes que se otorgan para los servicios por defecto (ambos archivos se obtuvieron de Github):

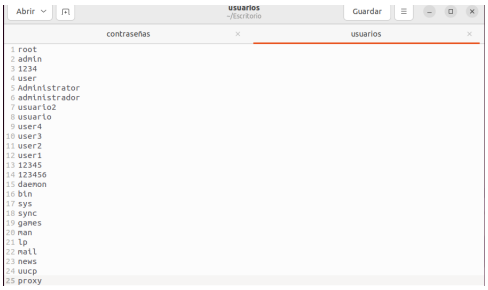
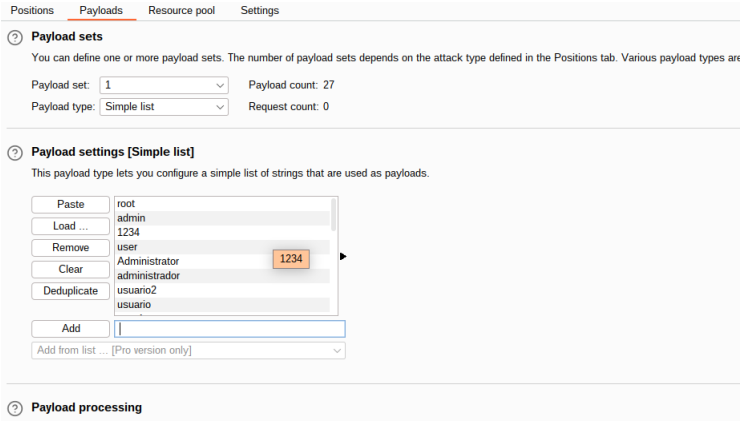


Figura 2: Usuarios mas comunes

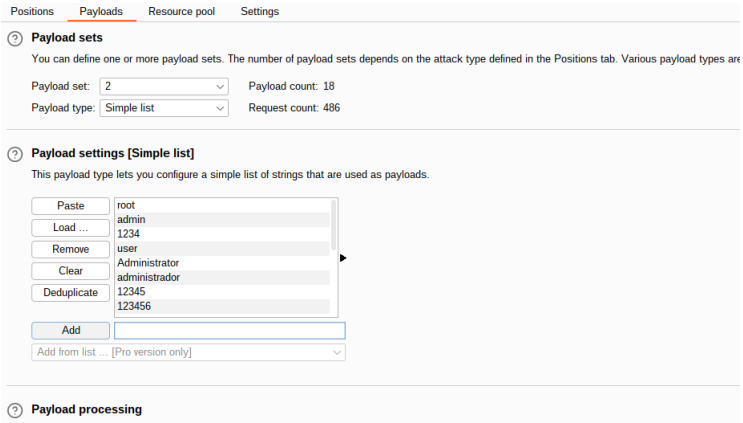
Figura 3: Contraseñas mas comunes

2.6. Obtención de al menos 2 pares (burp)

Una vez que hayamos elegido los diccionarios, el siguiente paso es seleccionar el tipo de ataque 'Cluster Bomb' y cargar los diccionarios en sus campos correspondientes, en este caso el de usuarios:



Luego se cargan en el segundo paylod, que es el de las contraseñas



Después de realizar estos pasos, iniciamos el ataque y observamos las múltiples solicitudes realizadas. Este proceso puede llevar mucho tiempo debido al gran número de solicitudes y al límite establecido en Burp Suite. Una vez que el ataque ha finalizado, podemos identificar dos combinaciones de usuario y contraseña válidas gracias a la longitud de sus respuestas:

Figura 4: Primer login correcto

Figura 5: Segundo login correcto

Al observar el render de cada uno de los logins, se observa lo siguiente:
Primer render:

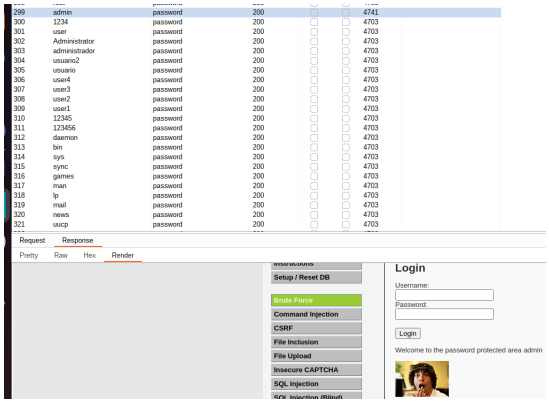


Figura 6: Primer render del login correcto

Segundo render:

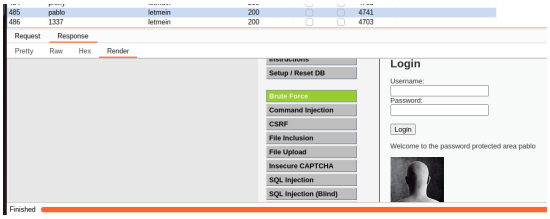


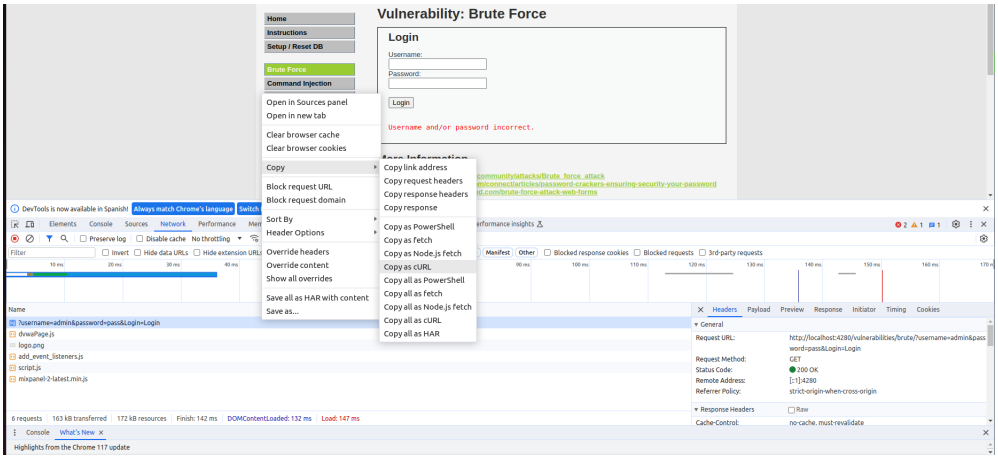
Figura 7: Segundo render del login correcto

Podemos observar que aunque ambas respuestas son válidas, difieren en su contenido y muestran imágenes diferentes según el usuario ingresado. Como regla general, las respuestas de acceso válido tienen una longitud de 4741, mientras que las demás consultas tienen una longitud de 4703. No se encontraron más combinaciones de usuario y contraseña válidas aparte de estas dos. Las respuestas restantes produjeron el mismo resultado que la vista en la imagen durante el ingreso manual.

2.7. Obtención de código de inspect element (curl)

Después de obtener los usuarios válidos, se requería emplear el comando 'curl' para llevar a cabo dos solicitudes. Para hacerlo, era necesario obtener el código de solicitud específico que se usaría en la terminal. Para este propósito, se recurrió al navegador Chrome y se aprovecharon las herramientas de desarrollo disponibles.

Se repitió el procedimiento de efectuar una consulta manual con el propósito de registrarla en el navegador. Luego, se abrieron las herramientas de desarrollo, donde se ubicó la solicitud y se copió en el formato adecuado para utilizarla con el comando 'curl'.



2.8. Utilización de curl por terminal (curl)

Una vez que se copió el código, se pegó en la terminal y se editaron los campos 'username' y 'password' con los valores que se deseaba ingresar. Para las credenciales de acceso válido, se probó con 'admin' y 'password', mientras que para el acceso inválido se utilizaron 'esteno' y 'estesiqueno', respectivamente.

```
root@ignacio-Modern-14-B5M:/home/ignacio# curl 'http://localhost:4280/vulnerabilities/brute/?username=admin&password=pass&Login=Login' \/home/ign
  -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' \cation/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7' \/apng,*/*;q=0  -  -
  -H 'Accept-Language: en,es-CL;q=0.9,es;q=0.8' \Accept-Language: en,es-CL;q=0.9,es;q=0.8' \n,es-CL  -H 'Accept-Language: en,es-CL;q=0.9,es  -H  -H
  -H 'Connection: keep-alive' \
  -H 'Cookie: PHPSESSID=1784e575a4a569f8157c709cfdd5c2b7; security=low' \
  -H 'Referer: http://localhost:4280/vulnerabilities/brute/?username=admin&password=password&Login=Login' \
  -H 'Sec-Fetch-Dest: document' \280/vulnerabilities/brute/?username=admin&password=password&Login=Login' \oc  -H 'Sec-Fe  -H 'Sec-Fetc  -H 'Sec-
  -H 'Sec-Fetch-Mode: navigate' \tch-Dest:  -H 'Sec-  -H 'Sec-Fetch-De  -H 'Sec-Fetch-Site: sa  -H 'Sec-Fetch-Mode: naviga  -H 'Sec-Fetch-Mode
  -H 'Sec-Fetch-Site: same-origin' \
  -H 'Sec-Fetch-User: ?1' \origin' \
  -H 'Upgrade-Insecure-Requests: 1' \
  -H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36' \ -H 'User-Age  -H 'User
  -H 'sec-ch-ua: "Google Chrome";v="117", "Not;A=Brand";v="8", "Chromium";v="117"' \117.0.0.0 Safari/537.36' \", "Not;A=Bra  -H 'sec-ch-ua: "Go
  -H 'sec-ch-ua-mobile: ?0' \d";v="8", "Chromium";v="117"' \ \me";v="  -H 'sec-ch-ua: "Google Chrome";v="117",  -H 'sec-ch-ua-mobile: ?0' \m  -  -
  -H 'sec-ch-ua-platform: "Linux"' \8", "Ch  -H 'sec-ch-ua-mobile: ?0' \
  --compressed
```

Figura 8: Comando curl en terminal

Figura 9: Comando curl en terminal con datos invalidos

Figura 10: Comando curl en terminal con datos validos

Lo que resulta intrigante en este proceso es que después de ejecutar el comando 'curl' que hemos copiado desde la terminal, la página 'DVWA' proporcionará una respuesta que reflejará el resultado de la solicitud. En este contexto, es lógico esperar que si tanto el nombre de usuario como la contraseña son correctos, la respuesta indicará un inicio de sesión exitoso. Por otro lado, si ingresamos credenciales inválidas, la respuesta debería mostrar un inicio de sesión incorrecto.

2.9. Demuestra 4 diferencias (curl)

En cuanto a 'curl', solo se pudo identificar una diferencia notable en la respuesta. Al comparar ambos textos, resultaron ser idénticos, a excepción de la sección que indicaba si el inicio de sesión había sido exitoso o no:

```
</form>
<p>Welcome to the password protected area admin</p><img src="/hackable/users/admin.jpg" />
```

```
    </form>
    <pre><br />Username and/or password incorrect.</pre>
</v>
```

Salvo por lo mencionado, las respuestas eran idénticas hasta el más mínimo detalle.

2.10. Instalación y versión a utilizar (hydra)

Finalmente, se nos pidió repetir el mismo ataque que habíamos ejecutado previamente en Burp Suite, pero esta vez utilizando la herramienta Hydra. Para llevar a cabo esta tarea, tuvimos que proceder con la instalación del software Hydra, y esta instalación se realizó a través de la línea de comandos en la terminal:

Figura 11: Instalación Hydra

Hydra es una herramienta de lınea de comandos que permite automatizar la prueba de credenciales en diversos servicios, como servicios web, servidores FTP, bases de datos, entre otros. Al proporcionar una lista de nombres de usuario y una lista de contraseñas, Hydra intenta iniciar sesión en el servicio objetivo probando todas las combinaciones posibles hasta encontrar un nombre de usuario y una contraseña validos o hasta que se agoten las combinaciones.

Utilizando el comando 'hydra', se verificó la versión del software que íbamos a emplear, la cual resultó ser la 9.2.

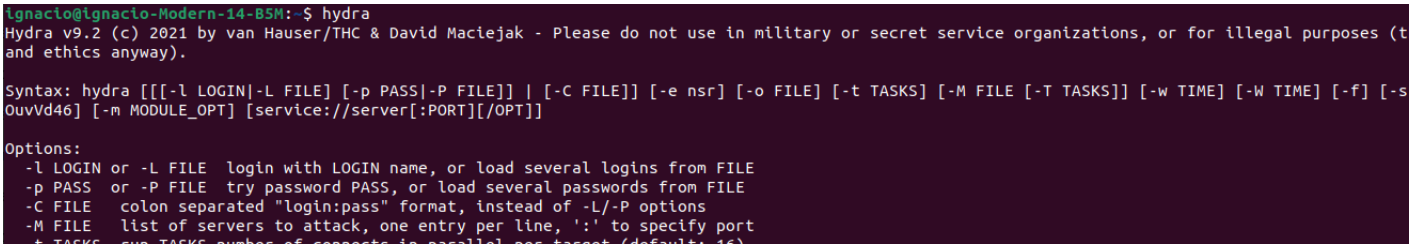


Figura 12: Instalación Hydra

2.11. Explicación de comando a utilizar (hydra)

Hemos empleado la herramienta Hydra para llevar a cabo un ataque de fuerza bruta contra un formulario web en un servidor local. El comando utilizado es el siguiente:

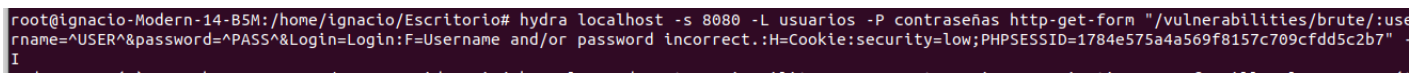


Figura 13: Comando Hydra

Este comando consta de varias partes clave. En primer lugar, especificamos el objetivo del ataque, que es un servidor local en el puerto 8080. Luego, definimos las listas de usuarios y contraseñas que serán utilizadas en el ataque, indicando los archivos 'usuarios.txt' y 'contraseñas.txt' respectivamente.

El comando también incluye la configuración del formulario web que se va a atacar. Esto se logra mediante la indicación del método HTTP, en este caso 'http-get-form', y la URL del

formulario objetivo, que se encuentra en '/vulnerabilities/brute/'. Además, especificamos los parámetros del formulario, donde 'ÛSER' y 'PASS' son marcadores que serán reemplazados por los nombres de usuario y contraseñas de nuestras listas.

Asimismo, definimos los mensajes que Hydra debe buscar para determinar si una combinación de usuario/contraseña es válida. El mensaje de éxito, 'Username and/or password incorrect. ', se utiliza para identificar un acceso fallido, mientras que la opción ' -I' se emplea para que Hydra finalice la búsqueda después de encontrar una combinación válida.

2.12. Obtención de al menos 2 pares (hydra)

Después de haber configurado y definido el comando necesario, procedimos a dar inicio al ataque correspondiente, lo que nos llevó a obtener la siguiente respuesta:

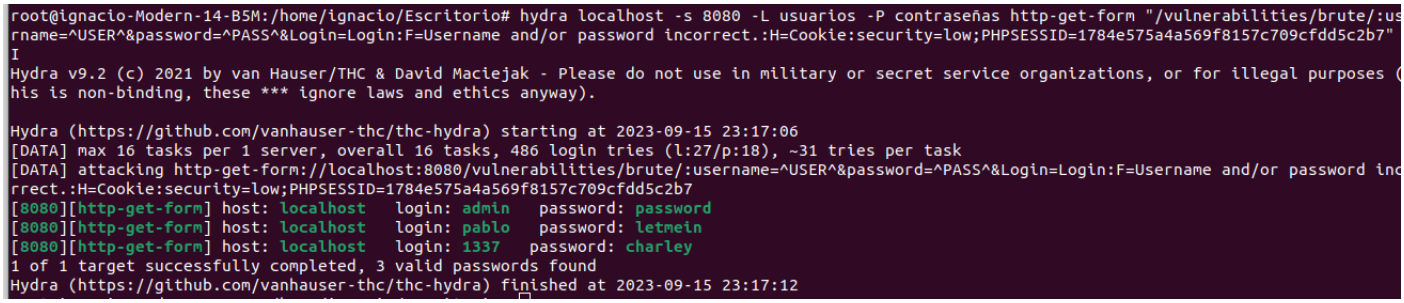


Figura 14: Resultado de comando Hydra

Como se puede apreciar, se lograron obtener tres conjuntos de credenciales válidas, y estos coinciden con los que se obtuvieron previamente a través de Burp Suite. Este resultado confirma la efectividad y precisión de la consulta realizada.

2.13. Explicación paquete curl (tráfico)

Se solicitó llevar a cabo un análisis del tráfico generado por cada programa, para lo cual se procedió a capturar una solicitud de prueba de cada programa utilizando las credenciales 'admin' y 'password' utilizando Wireshark.

Para curl se obtuvo lo siguiente:

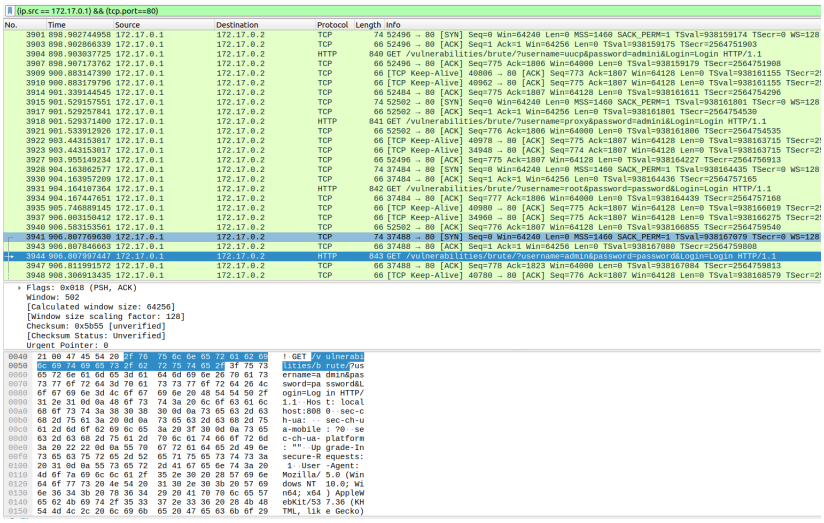


Figura 15: Resultado de paquetes curl

Como era de esperar, se puede observar que el contenido del paquete, incluyendo la información del método GET, es legible en texto plano cuando se utiliza Wireshark. Esto se debe a que se está empleando el protocolo HTTP en lugar de HTTPS, lo que significa que los datos no están cifrados y son visibles en la red.

Cuando se examina con mayor detalle el contenido del paquete, se puede notar que contiene una gran cantidad de información sobre la entidad que realiza la solicitud. Esto incluye datos como el navegador utilizado, las preferencias de lenguaje, la cookie de la sesión actual, la indicación de mantener la conexión abierta (keep-alive) y una serie de encabezados Sec-fetch que proporcionan información sobre el tipo de recurso solicitado y otros detalles relacionados. Estos datos son valiosos para entender la solicitud y el contexto en el que se está realizando.

2.14. Explicación paquete burp (tráfico)

Realizamos el mismo proceso para el paquete de Burp Suite, enviando las mismas credenciales.

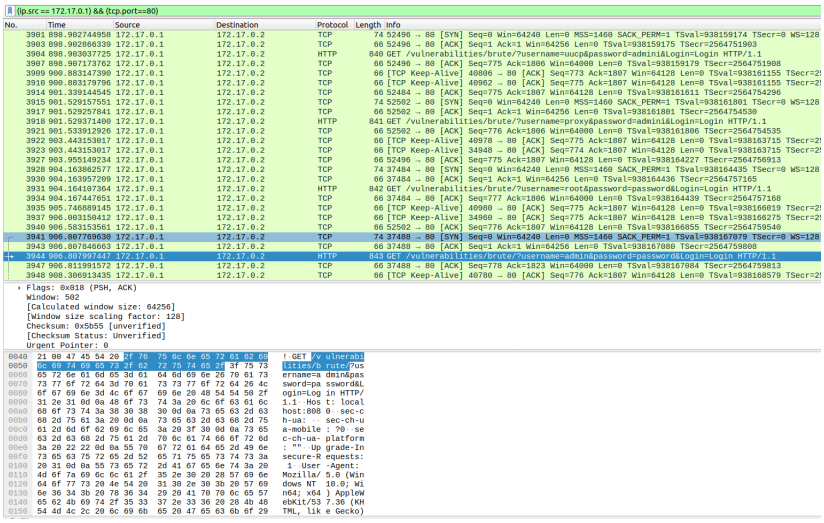


Figura 16: Resultado de paquetes burp

Podemos notar que el paquete generado por Burp Suite es similar al enviado por ‘curl‘, ya que ambos utilizan el mismo protocolo y contienen información del navegador, cookies y los mismos encabezados Sec-fetch. Sin embargo, el tamaño del paquete generado por Burp Suite es ligeramente mayor que el de ‘curl‘. Esto podría deberse a diferencias en los contenidos de los campos, como por ejemplo, Burp Suite incluye ’Chrome’ y ’Safari’ dentro de sus user-agent, mientras que ‘curl‘ no lo hace. En general, ambos paquetes mantienen una estructura similar debido a que están realizando solicitudes similares.

2.15. Explicación paquete Hydra (tráfico)

Repetimos el proceso por última vez para Hydra, con las misma credenciales:

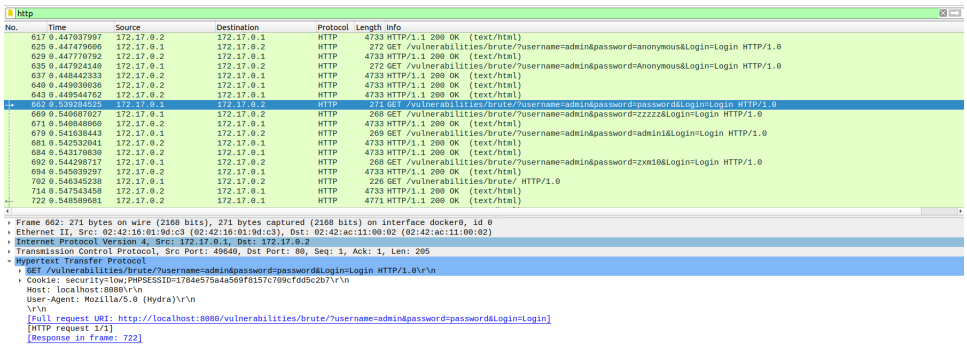


Figura 17: Resultado de paquetes burp

En el caso de Hydra, notamos que el paquete resultante es considerablemente más compacto, ya que incluye únicamente los elementos esenciales necesarios para una solicitud GET. Esta diferencia podría explicarse por la orientación de Hydra hacia ataques de fuerza bruta, lo que podría llevarlo a simplificar las peticiones para agilizar el proceso. A diferencia de las solicitudes generadas por Burp Suite y 'curl', Hydra no incorpora detalles del navegador ni los campos Sec-fetch, ya que su enfoque se centra en realizar intentos de acceso de manera eficiente y directa.

2.16. Mención de las diferencias (tráfico)

Como se pudo notar anteriormente, los paquetes enviados se distinguen principalmente por su tamaño y los campos que contienen, aunque todos siguen la estructura básica de una solicitud GET. En este sentido, Curl y Burp Suite presentan estructuras prácticamente idénticas, diferenciándose principalmente en los contenidos específicos. En contraste, Hydra envía un paquete que se destaca fácilmente debido a su tamaño notablemente diferente, lo que lo hace fácilmente identificable en comparación con los otros dos.

En general:
Curl:

- Nuevamente, el trafico comienza con un paquete SYN para iniciar una nueva conexión TCP.
- Curl realiza una solicitud HTTP GET con un nombre de usuario y una contraseña específicos para simular un intento de inicio de sesión.
- La respuesta HTTP muestra si el intento de inicio de sesión fue exitoso o no.

Burp Suite:

- El trafico también comienza con un paquete SYN para iniciar una nueva conexión TCP.
- La respuesta HTTP muestra si el intento de inicio de sesión fue exitoso o no.
- Los paquetes FIN se utilizan para cerrar la conexión TCP después de la comunicación.

Hydra:

- El trafico comienza con un paquete SYN (Sincronización) indicando el inicio de una nueva conexión TCP
- La respuesta HTTP varia dependiendo de si el intento de inicio de sesión fue exitoso o no.
- Se utilizan paquetes FIN (Finalización) para cerrar la conexión TCP después de cada intento de inicio de sesión.

2.17. Detección de SW (tráfico)

Al analizar los paquetes, se hace evidente que el campo más sencillo de detectar en cada uno de ellos es el 'User-Agent', lo que podría facilitar su identificación si previamente se ha tenido contacto con este tráfico. En el caso de Curl, el campo 'User-Agent' incluye referencias a Mozilla y Gecko, mientras que en Burp Suite, se mencionan Mozilla, Apple Web Kit, Chrome y Safari, lo que brinda pistas para distinguir los paquetes generados por estos dos programas. No obstante, la detección más evidente se encuentra en Hydra, ya que se lista directamente en este campo, lo que permite su identificación de manera inmediata.

Conclusiones y comentarios

Durante esta experiencia, se pudo adentrar en el mundo de la ciberseguridad y fortalecer la comprensión de los ataques de fuerza bruta. Se profundizaron conocimientos esenciales en esta área, como el uso de diccionarios y la comprensión de distintos tipos de ataques. Además, se tuvo la oportunidad de practicar con herramientas nuevas, lo que amplió el conjunto de habilidades y recursos disponibles para abordar desafíos futuros relacionados con la seguridad cibernética. Esta experiencia no solo contribuyó al conocimiento teórico, sino que también brindó una valiosa práctica en el campo, lo que podría resultar extremadamente útil en futuras situaciones relacionadas con la seguridad de la información.

Es fundamental resaltar que estas actividades demandaron un tiempo considerable y un enfoque meticuloso para asimilar y aplicar los conceptos involucrados. El análisis de paquetes y la identificación de disparidades entre las herramientas proporcionaron una comprensión más profunda sobre cómo se ejecutan los ataques y cómo pueden ser detectados. Este proceso de aprendizaje no solo incrementó el conocimiento teórico, sino que también cultivó habilidades prácticas esenciales en el ámbito de la ciberseguridad, lo que resulta crucial en la defensa contra posibles amenazas y ataques en el futuro.

En resumen, se lograron alcanzar los objetivos planteados en esta experiencia y se considera que fue productiva en términos de aprendizaje y adquisición de habilidades en ciberseguridad. No obstante, es importante señalar que la duración de la experiencia se percibió como un poco extensa en relación con el plazo disponible para completarla. A pesar de este desafío temporal, los conocimientos y la experiencia adquiridos resultaron valiosos y beneficiosos para el desarrollo de habilidades en esta área crucial.