Informe Laboratorio 2

Sección 2

Cristóbal León
e-mail: cristobal.leon1@mail.udp.cl

14 de Septiembre de 2023

Índice

1. Descripción de actividades **2**

2. Desarrollo de actividades según criterio de rúbrica **2**
2.1. Levantamiento de docker para correr DVWA (dvwa) 2
2.2. Redirección de puertos en docker (dvwa) 3
2.3. Obtención de consulta a replicar (burp) 3
2.4. Identificación de campos a modificar (burp) 3
2.5. Obtención de diccionarios para el ataque (burp) 3
2.6. Obtención de al menos 2 pares (burp) 4
2.7. Obtención de código de inspect element (curl) 4
2.8. Utilización de curl por terminal (curl) 4
2.9. Demuestra 5 diferencias (curl) 4
2.10. Instalación y versión a utilizar (hydra) 7
2.11. Explicación de comando a utilizar (hydra) 8
2.12. Obtención de al menos 2 pares (hydra) 8
2.13. Explicación paquete curl (tráfico) 8
2.14. Explicación paquete burp (tráfico) 8
2.15. Explicación paquete hydra (tráfico) 9
2.16. Mención de las diferencias (tráfico) 9
2.17. Detección de SW (tráfico) 9

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 llevar a cabo los experimentos y pruebas de seguridad, se optó por utilizar la Damn Vulnerable Web Application (DVWA), una plataforma especialmente diseñada para probar la seguridad en aplicaciones web de manera ética.

Proceso de Despliegue Descarga de Docker: Se inició el proceso descargando e instalando Docker en el sistema operativo, que servirá como el entorno de ejecución para DVWA.

Pull de la Imagen de DVWA: Posteriormente, se hizo un "pull"de la imagen de DVWA previamente creada y disponible en el repositorio de Docker Hub. Este paso es crucial, ya que la imagen contiene todos los archivos y configuraciones necesarios para ejecutar DVWA. El comando utilizado para este fin fue:

docker pull vulnerables/web-dvwa

imagen 1

Este enfoque permitió un despliegue rápido y efectivo de DVWA, al no requerir configuraciones adicionales y asegurar un ambiente estandarizado para las pruebas de seguridad.

```
(base) admin@iMac-de-Cristobal DVWA % sudo docker pull vulnerables/web-dvwa

Password:
Using default tag: latest
2023/09/13 20:41:51 must use ASL logging (which requires CGO) if running as root
latest: Pulling from vulnerables/web-dvwa
3e17c6eae66c: Pull complete
0c57df616dbf: Pull complete
eb05d18be401: Pull complete
e9968e5981d2: Pull complete
2cd72dba8257: Pull complete
6cff5f35147f: Pull complete
098cffd43466: Pull complete
b3d64a33242d: Pull complete
Digest: sha256:dae203fe11646a86937bf04db0079adef295f426da68a92b40e3b181f337daa7
Status: Downloaded newer image for vulnerables/web-dvwa:latest
docker.io/vulnerables/web-dvwa:latest
```

Figura 1: Imagen 1

2.2. Redirección de puertos en docker (dvwa)

Luego se hace push a la imagen de Docker que ya venía creada y se despliega en el puerto 8080.

imagen 2

```
(base) admin@iMac-de-Cristobal DVWA % sudo docker run -d -p 8081:80 vulnerables/
web-dvwa

812188ddb16b05c76494b8f281421eda0c310385c86b4e999cc3dde534fe477d
(base) admin@iMac-de-Cristobal DVWA % sudo docker pull vulnerables/web-dvwa
```

Figura 2: Imagen 2

2.3. Obtención de consulta a replicar (burp)

Después, en Burpsuite se intercepta el HTML del login del sitio desplegado y con la base de datos ya creada.

imagen 3

2.4. Identificación de campos a modificar (burp)

Los campos a modificar son Username y Password agregandole un § antes y despues

2.5. Obtención de diccionarios para el ataque (burp)

Luego, se utiliza el diccionario facilitado por el laboratorio para atacar la página.

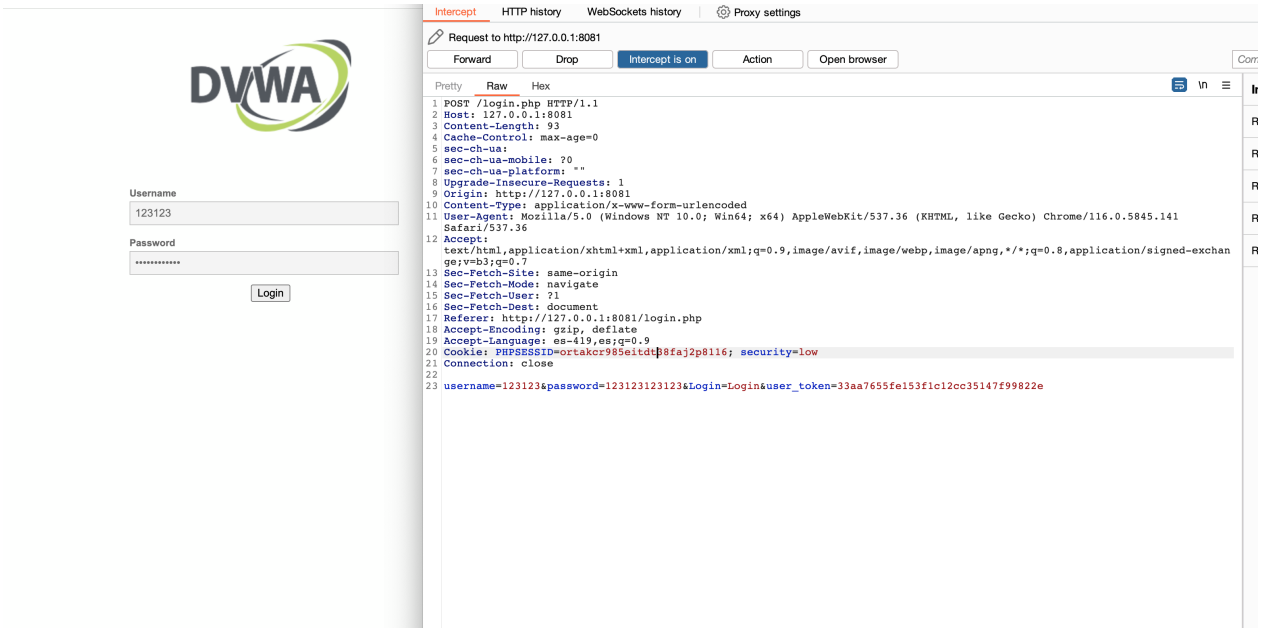

Figura 3: Imagen 3

imagen 4

2.6. Obtención de al menos 2 pares (burp)

Primero se obtiene el siguiente par de contraseñas.
imagen 5
Luego, se obtuvo el par admin/123123

2.7. Obtención de código de inspect element (curl)

En el sitio web desplegado por Docker se le hace inspect element al Login.
imagen 6

2.8. Utilización de curl por terminal (curl)

Luego, se utilizo curl en al terminal entregando el siguiente resultado:
imagen 7

2.9. Demuestra 5 diferencias (curl)

Aquí hay cinco diferencias que se puden encontrar entre los dos archivos HTML:

- Título de la página: En "bueno2 -1.html", el título es "Login :: Damn Vulnerable Web Application (DVWA) v1.10 Development". Este título no está presente en "malo1.html".

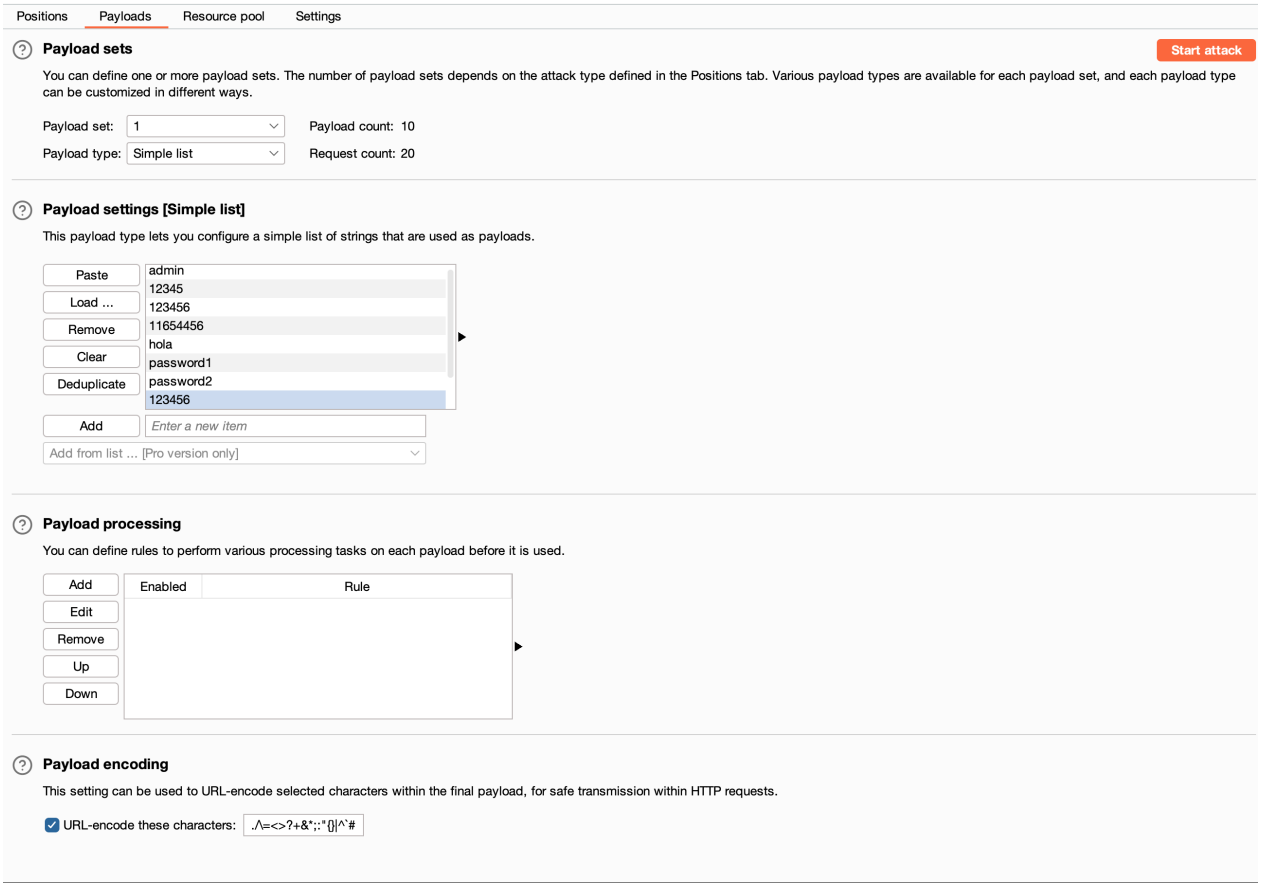

Figura 4: Imagen 4

```
POST /login.php HTTP/1.1
Host: 127.0.0.1:8081
Content-Length: 93
Cache-Control: max-age=0
sec-ch-ua:
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: ""
Upgrade-Insecure-Requests: 1
Origin: http://127.0.0.1:8081
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.141
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchan
ge;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://127.0.0.1:8081/login.php
Accept-Encoding: gzip, deflate
Accept-Language: es-419,es;q=0.9
Cookie: PHPSESSID=ortakcr985eitdt38faj2p8116; security=low
Connection: close

username=123123&password=123123123123&Login=Login&user_token=33aa7655fe153f1c12cc35147f99822e
```

Figura 5: Imagen 5

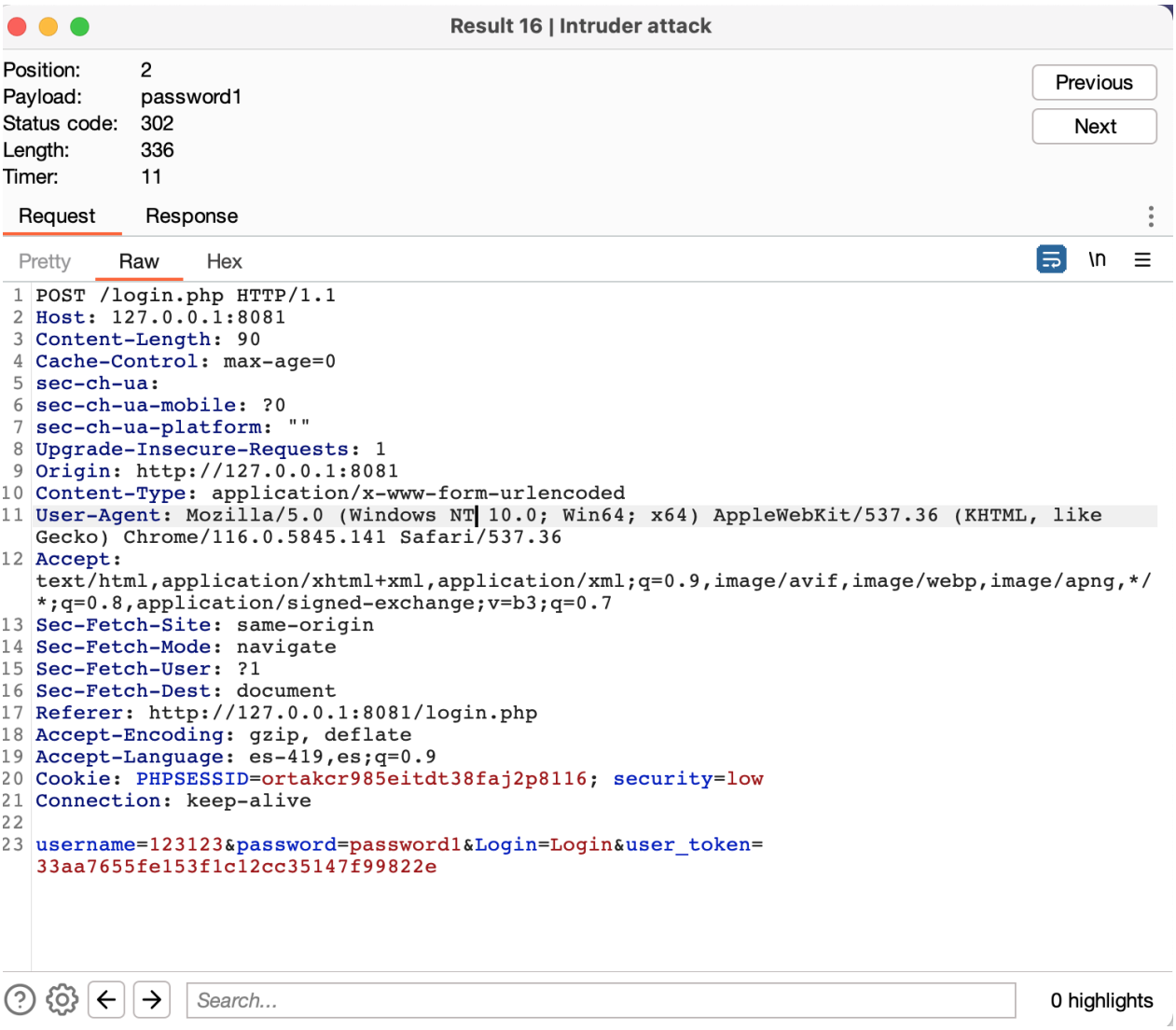

Figura 6: Imagen 6

  - Mensaje de error: En "malo1.html", hay un mensaje de error que dice Üsername and/or password incorrect.". Este mensaje no está presente en "bueno2 -1.html".
  - Token CSRF: En "bueno2 -1.html", hay un token CSRF (üser$_{t}oken$")$queseestableceenel formulari$

- Comentarios HTML: "bueno2 -1.html"tiene comentarios HTML como ¡!– 2017-07-06 19:50:59 CEST [1.00] 1.111, 0.100, 0.000 –¿que no se encuentran en "malo1.html".
- Metadatos: El archivo "bueno2 -1.htmlïncluye una etiqueta meta para la codificación de caracteres (¡meta http-equiv=Çontent-Typeçontent="text/html; charset=UTF-8») que no está presente en "malo1.html".

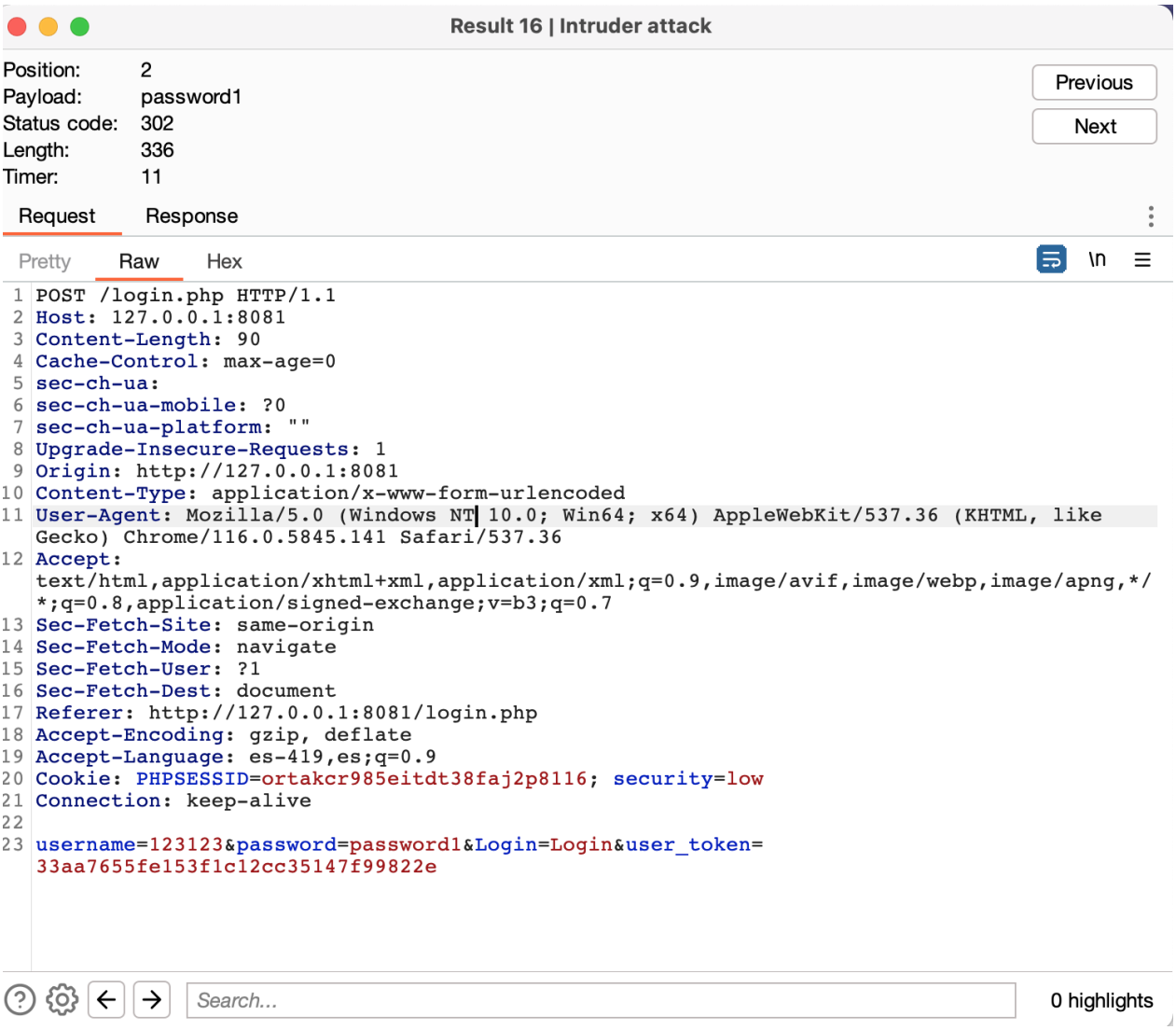

Figura 7: Imagen 7

2.10. Instalación y versión a utilizar (hydra)

Luego, se procede a instalar y validar la versión a utilizar de Hydra:
imagen 8

```
(base) admin@iMac-de-Cristobal Lab2 %
(base) admin@iMac-de-Cristobal Lab2 % brew install hydra
Running `brew update --auto-update`...
==> Auto-updated Homebrew!
Updated 2 taps (homebrew/core and homebrew/cask).
==> New Formulae
apprise            checkdmarc         gotestwaf          recoverpy          surelog
==> New Casks
akuity                 meld-studio            wetype                 wiso-steuer-2023
floorp                 proxy-audio-device     wiso-steuer-2022

You have 17 outdated formulae installed.

hydra 9.4 is already installed but outdated (so it will be upgraded).
==> Downloading https://ghcr.io/v2/homebrew/core/hydra/manifests/9.5_2
######################################################################################## 100.0%
==> Fetching dependencies for hydra: ca-certificates, openssl@3, libssh, xz and mysql-client
==> Downloading https://ghcr.io/v2/homebrew/core/ca-certificates/manifests/2023-08-22
######################################################################################## 100.0%
==> Fetching ca-certificates
==> Downloading https://ghcr.io/v2/homebrew/core/ca-certificates/blobs/sha256:a331e92e7a75957
######################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/openssl/3/manifests/3.1.2-1
######################################################################################## 100.0%
==> Fetching openssl@3
==> Downloading https://ghcr.io/v2/homebrew/core/openssl/3/blobs/sha256:23b87e2e71c62a0128d17
######################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/libssh/manifests/0.10.5_1
######################################################################################## 100.0%
==> Fetching libssh
==> Downloading https://ghcr.io/v2/homebrew/core/libssh/blobs/sha256:9caf4d93dc7fd37ba7b9acb0
```

Figura 8: Imagen 8

2.11. Explicación de comando a utilizar (hydra)

2.12. Obtención de al menos 2 pares (hydra)

2.13. Explicación paquete curl (tráfico)

Cuando se utiliza cURL para hacer una solicitud HTTP o HTTPS, los paquetes generados son bastante estándar. Se crea una conexión TCP con el servidor y se envía una petición HTTP que puede incluir diferentes métodos como GET o POST, además de encabezados y datos si es necesario. A pesar de su versatilidad, el tráfico de red generado por cURL suele ser fácil de identificar.

2.14. Explicación paquete burp (tráfico)

Burp Suite es una herramienta de pruebas de seguridad que puede generar una variedad de tráficos de red. Dependiendo de la prueba que se esté realizando, el tráfico puede variar desde simples peticiones HTTP hasta secuencias de tráfico más complejas que pueden incluir múltiples tipos de ataques y encabezados personalizados.

2.15. Explicación paquete hydra (tráfico)

Hydra es una herramienta diseñada para ataques de fuerza bruta. Produce un tráfico de red que es altamente repetitivo, con la variabilidad principal en los campos de datos que contienen las credenciales a probar. Al cambiar rápidamente estos campos, Hydra intenta encontrar combinaciones válidas de nombre de usuario y contraseña.

2.16. Mención de las diferencias (tráfico)

1. Patrones de Tráfico: cURL es más adecuado para tareas simples, Burp Suite para pruebas de seguridad más extensas, y Hydra para ataques de fuerza bruta.

2. Encabezados HTTP: cURL y Burp Suite permiten una gran personalización de los encabezados, mientras que Hydra se centra más en cambiar los datos del cuerpo de la solicitud.

3. Concurrencia: Hydra tiene la capacidad de generar un gran número de solicitudes en paralelo, algo que no es común en cURL o Burp Suite.

4. Versatilidad: Burp Suite puede realizar una variedad de ataques, a diferencia de cURL y Hydra, que son más especializados en tareas específicas.

5. Complejidad: Los paquetes generados por Burp Suite pueden ser más complejos debido a las múltiples funciones que ofrece para pruebas de seguridad.

2.17. Detección de SW (tráfico)

Para determinar qué herramienta está generando un cierto tipo de tráfico, se pueden observar varios indicadores. Estos pueden ir desde patrones específicos en el tráfico hasta la revisión de encabezados de usuario y la frecuencia de las solicitudes. Estas métricas ofrecen una idea general pero no definitiva del origen del tráfico.

Conclusiones y comentarios

En la experiencia detallada en el informe, se observa un enfoque meticuloso en el uso de múltiples herramientas para evaluar la seguridad de las aplicaciones web. Desde el despliegue inicial de DVWA en un entorno Docker hasta la utilización de herramientas especializadas como Burp Suite, cURL y Hydra, cada paso se explica con claridad.

El informe no solo se centra en la ejecución de las pruebas, sino que también ofrece un análisis comparativo del tráfico de red generado por las diferentes herramientas. Este nivel de detalle permite una comprensión más profunda de cómo cada herramienta opera y qué tipo de información se puede extraer de su uso, contribuyendo así a una evaluación de seguridad más completa y enriquecedora.

https://github.com/cristobal-leon1/lab2-cripto.git