

Informe Laboratorio 2

Sección 02

Alumno Jorge Toro Macías
e-mail: jorge.toro1@mail.udp.cl

Abril de 2024

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)	4
2.3. Obtención de consulta a replicar (burp)	5
2.4. Identificación de campos a modificar (burp)	7
2.5. Obtención de diccionarios para el ataque (burp)	8
2.6. Obtención de al menos 2 pares (burp)	10
2.7. Obtención de código de inspect element (curl)	12
2.8. Utilización de curl por terminal (curl)	12
2.9. Demuestra 5 diferencias (curl)	12
2.10. Instalación y versión a utilizar (hydra)	12
2.11. Explicación de comando a utilizar (hydra)	13
2.12. Obtención de al menos 2 pares (hydra)	14
2.13. Explicación paquete curl (tráfico)	15
2.14. Explicación paquete burp (tráfico)	15
2.15. Explicación paquete hydra (tráfico)	16
2.16. Mención de las diferencias (tráfico)	17
2.17. Detección de SW (tráfico)	17

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)

```
informatica@informatica-14:~$ git clone https://github.com/digininja/DVWA.git
Cloning into 'DVWA'...
remote: Enumerating objects: 4500, done.
remote: Counting objects: 100% (50/50), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 4500 (delta 17), reused 34 (delta 8), pack-reused 4450
Receiving objects: 100% (4500/4500), 2.30 MiB | 6.28 MiB/s, done.
Resolving deltas: 100% (2112/2112), done.
informatica@informatica-14:~$ ls
cualesmiip.html  Documents  DVWA       Pictures   snap       Videos
Desktop         Downloads  Music     Public    Templates
informatica@informatica-14:~$ cd DVWA
```

Figura 1: Clonación de repositorio para obtención de directorio.

En la figura 1 se muestra cómo se clona un repositorio de github para la obtención de la carpeta de DVWA donde se encuentran los archivos importantes tales como la imagen del docker que levantará la web de DVWA en el localhost.

2.1 Levantamiento del Docker de DVWA

```
informatica@informatica-14:~/DVWA$ docker compose up -d
permission denied while trying to connect to the Docker daemon socket at unix:///var/run
docker.compose.config-hash%22%3Atrue%2C%22com.docker.compose.project%3Ddvwa%22%3Atrue%7D
informatica@informatica-14:~/DVWA$ sudo docker compose up -d
[+] Running 27/19
✓ dvwa 17 layers [████████████████████████████████████████] 0B/0B Pulled 26.8s
✓ db 8 layers [████████████████████████████████████████] 0B/0B Pulled 26.7s
```

Figura 2: Levantamiento del docker de DVWA.

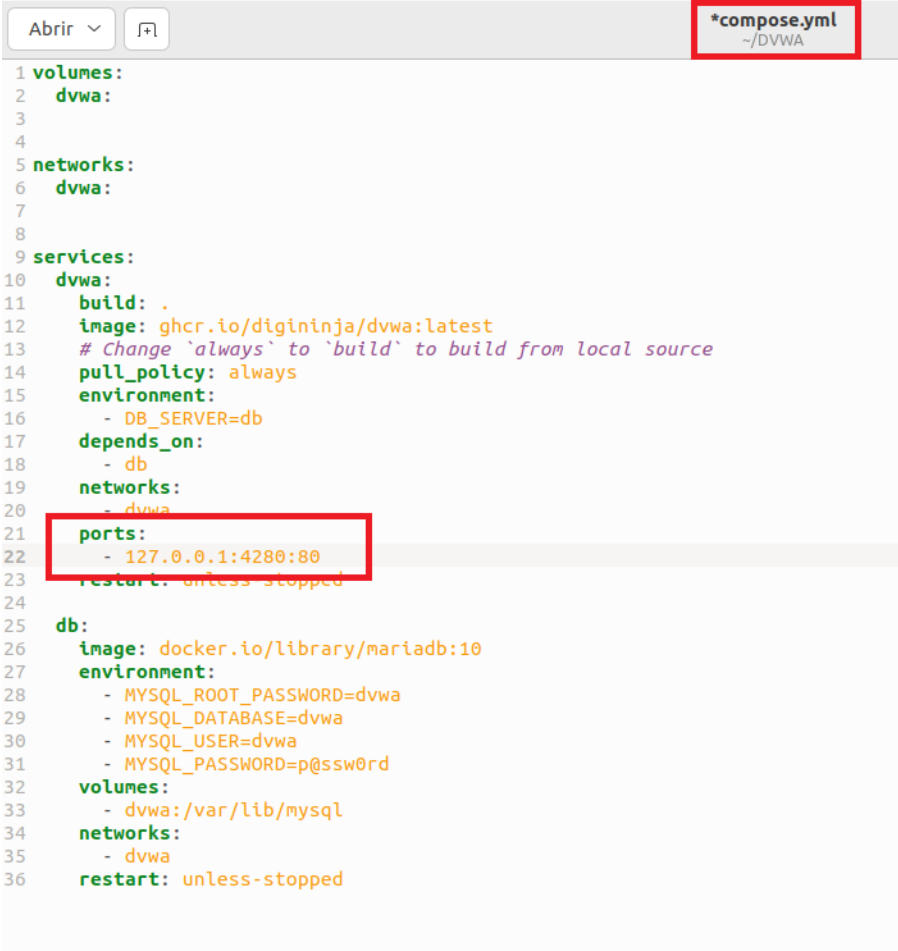
En la figura 2 se observa el levantamiento del docker de DVWA desde la carpeta que se clonó desde el github en el paso anterior (figura 1).

```
[+] Running 2/4
  ⚙ Network dvwa_dvwa      Created          9.4s
  ⚙ Volume "dvwa_dvwa"     Created          9.2s
  ✓ Container dvwa-db-1    Started          7.5s
  ✓ Container dvwa-dvwa-1  Started          3.2s
```

Figura 3: Docker de DVWA corriendo.

Finalmente, en la figura 3 se evidencia la ejecución exitosa de DVWA.

2.2. Redirección de puertos en docker (dvwa)



```
1 volumes:
2   dvwa:
3
4
5 networks:
6   dvwa:
7
8
9 services:
10  dvwa:
11    build: .
12    image: ghcr.io/digininja/dvwa:latest
13    # Change 'always' to 'build' to build from local source
14    pull_policy: always
15    environment:
16      - DB_SERVER=db
17    depends_on:
18      - db
19    networks:
20      - dvwa
21    ports:
22      - 127.0.0.1:4280:80
23    restart: unless-stopped
24
25  db:
26    image: docker.io/library/mariadb:10
27    environment:
28      - MYSQL_ROOT_PASSWORD=dvwa
29      - MYSQL_DATABASE=dvwa
30      - MYSQL_USER=dvwa
31      - MYSQL_PASSWORD=p@ssw0rd
32    volumes:
33      - dvwa:/var/lib/mysql
34    networks:
35      - dvwa
36    restart: unless-stopped
```

Figura 4: Configuración inicial del compose.

En la figura 4 se observa la configuración inicial del compose de DVWA. En esta está definido el puerto como el **4280**. Esto es porque usualmente el puerto **80** se utiliza para otros fines HTTP, o bien lo utilizan otras aplicaciones.



```
20      - dvwa
21    ports:
22      - 127.0.0.1:8080:80
23    restart: unless-stopped
24
```

Figura 5: Redireccionamiento de puerto.

2.3 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Se logra la redirección de puerto al **8080** editando los parámetros del compose, tal cual se observa en la figura 5.

2.3. Obtención de consulta a replicar (burp)

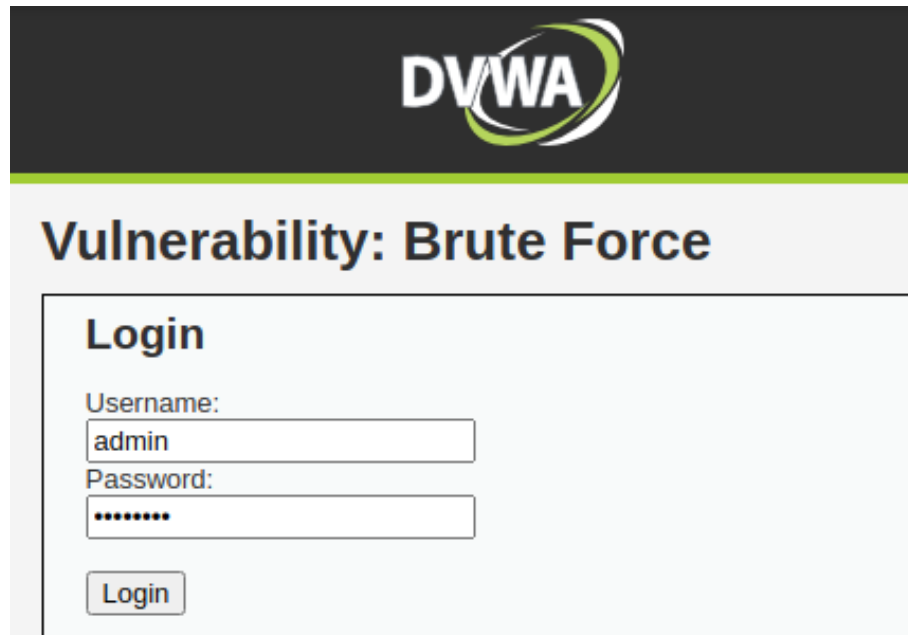


Figura 6: Menú de login de DVWA para fuerza bruta.

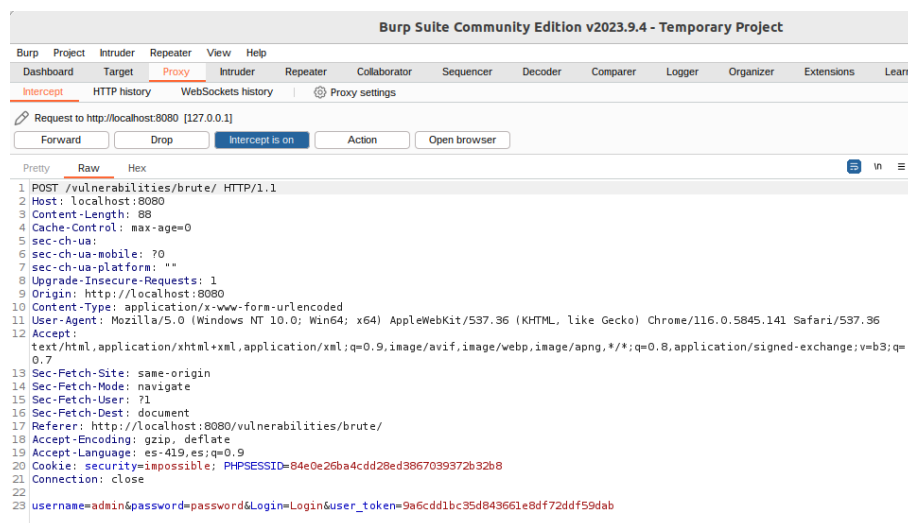


Figura 7: Captura del request en navegador Chromium de Burpsuite.

2.3 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

En las figuras 6 y 7 se muestra cómo se captura en Burpsuite el request de login que es enviado en DVWA. Esto con el fin de obtener los parámetros de interés para su posterior estudio.



Figura 8: Login exitoso con credenciales registradas.

2.4 Identificación de campos a modificar (burp)

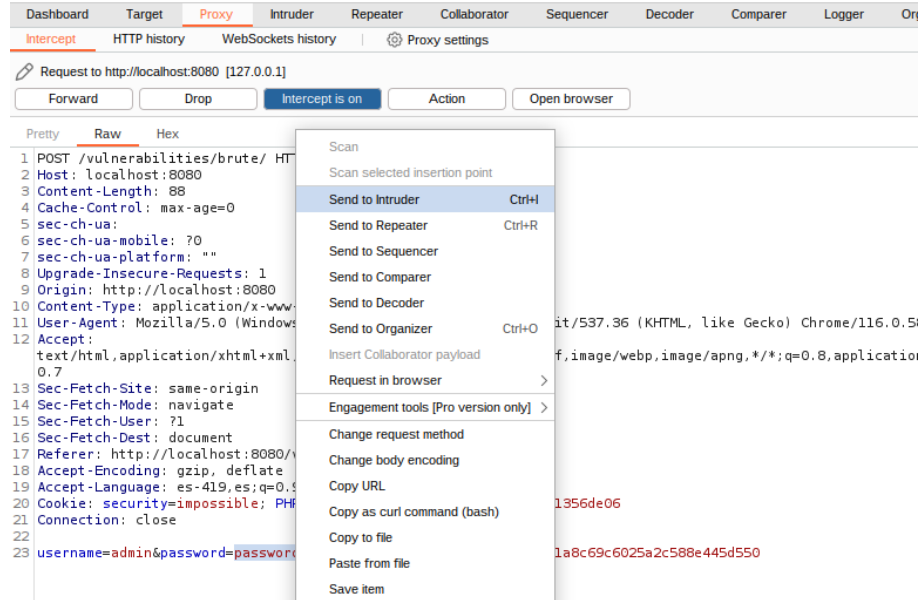


Figura 9: Envío de la consulta al intruder.

Luego, cuando ya se ha enviado la solicitud (figura 8) se procede a pasar la consulta al Intruder de Burpsuite para planear el ataque. Este paso se observa en la figura 9.

2.4. Identificación de campos a modificar (burp)



Figura 10: Elección del tipo de ataque.

Cuando ya se ha pasado la consulta al intruder se procede a elegir el tipo de ataque. En este caso el elegido es el de tipo **Cluster bomb** (figura 10), el cual permite atacar con más de un par de datos. Adecuado para un login.

```
22 |
23 | username=$admin$&password=$password$&
```

Figura 11: Identificación de parámetros para el ataque.

Dentro de la consulta que fue enviada al intruder se identifican los parámetros que se utilizarán en el ataque; en este caso las variables **username** y **password**. Lo que se probará serán valores de estos, así que se selecciona no la variable, si no el valor, como se puede observar en la figura 11.

2.5. Obtención de diccionarios para el ataque (burp)

The screenshot shows the Burp Suite Intruder interface. The 'Intruder' tab is selected, and the 'Payloads' sub-tab is active. Under 'Payload sets', there is a description: 'You can define one or more payload sets. The number of payload sets depends on the different ways.' Below this, 'Payload set' is set to '1' and 'Payload count' is '7'. 'Payload type' is set to 'Simple list' and 'Request count' is '0'. Under 'Payload settings [Simple list]', there is a description: 'This payload type lets you configure a simple list of strings that are used as payloads.' On the left, there are buttons: 'Paste', 'Load ...', 'Remove', 'Clear', and 'Deduplicate'. In the center, a list of strings is shown: 'admin', 'gordonb', 'smithy', '1337', 'pablo', 'jorge', and 'user'. On the right, there is an 'Add' button and a text input field. At the bottom, there is a dropdown menu labeled 'Add from list ... [Pro version only]'.

Figura 12: Diccionarios para usuario.

The screenshot shows the Burp Suite Intruder tab with the 'Payloads' sub-tab selected. The 'Payload sets' section indicates 2 payload sets and a payload count of 6. The 'Payload settings [Simple list]' section shows a list of payloads: password, letmein, abc123, charley, luna, and jaja. There are buttons for Paste, Load, Remove, Clear, Deduplicate, and Add. An 'Add from list ... [Pro version only]' dropdown is also visible.

Figura 13: Diccionarios para contraseña.

En las figuras 12 y 13 se observa cómo se añaden valores a probar para cada parámetro o payload, usuario y contraseña, respectivamente. Con anterioridad se obtuvo las credenciales correctas consultando en sitios externos, por lo que se agregaron todas para la prueba y un par de valores incorrectos.

2.6 Obtención de las actividades según criterio de rúbrica

2.6. Obtención de al menos 2 pares (burp)

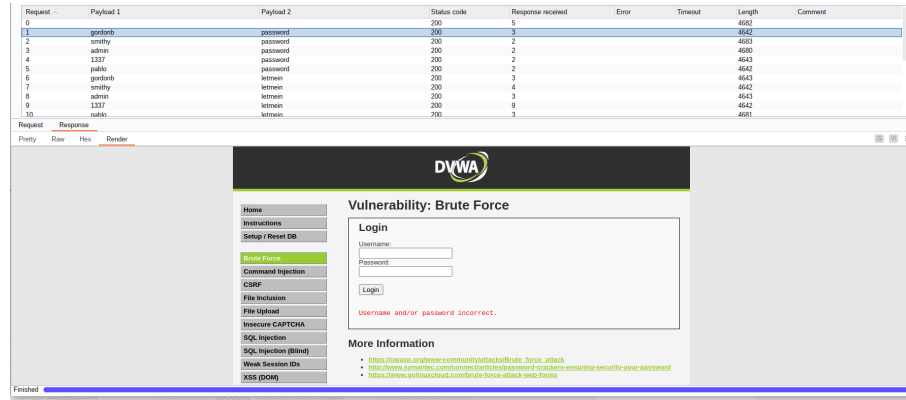


Figura 14: Caso 1. Credenciales incorrectas.

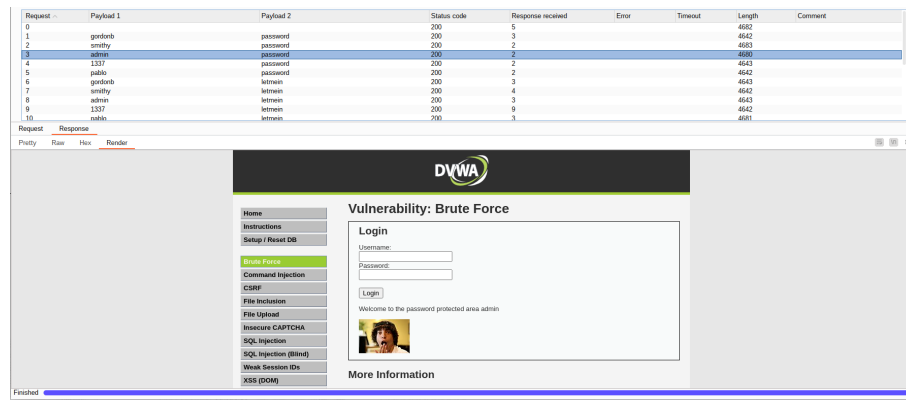


Figura 15: Caso 2. Credenciales correctas.

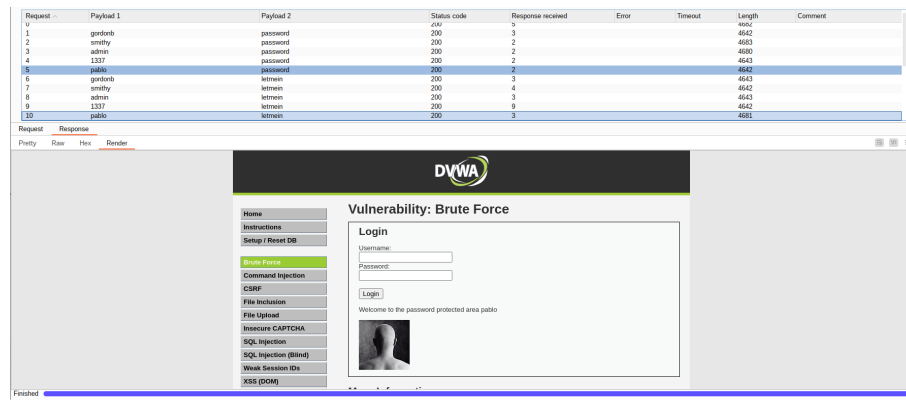


Figura 16: Caso 3. Credenciales correctas.

2.6 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
9	1337	letmein	200	5			4642	
10	public	letmein	200	3			4681	
11	gordonb	abc123	200	2			4684	
12	sm00th	abc123	200	2			4643	
13	admin	abc123	200	2			4642	
14	1337	abc123	200	2			4643	
15	public	abc123	200	2			4642	
16	gordonb	charley	200	3			4643	
17	sm00th	charley	200	2			4642	
18	admin	charley	200	2			4643	

Request	Response
Pretty	Raw
View	Render

Figura 17: Caso 4. Credenciales correctas.

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
11	gordonb	admin	200	2			4684	
12	sm00th	abc123	200	2			4643	
13	admin	abc123	200	2			4642	
14	1337	abc123	200	2			4643	
15	public	abc123	200	2			4642	
16	gordonb	charley	200	3			4643	
17	sm00th	charley	200	2			4642	
18	admin	charley	200	2			4643	
19	1337	charley	200	2			4642	
20	public	charley	200	3			4643	

Request	Response
Pretty	Raw
View	Render

Figura 18: Caso 5. Credenciales correctas.

Finalmente, en la figura 14 tenemos como primer caso uno donde las credenciales no coincidían con los registros de la base de datos, por lo que retornó un mensaje de 'usuario o contraseña incorrectos'.

Por otra parte, podemos observar casos de pares correctos en las figuras 15, 16, 17 y 18, donde se obtuvo como respuesta una imagen distinta para cada caso de pares exitosos.

2.7. Obtención de código de inspect element (curl)

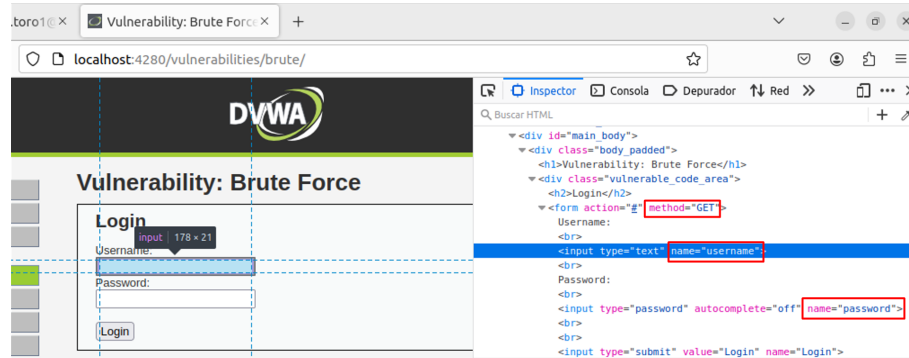


Figura 19: Código de inspect element.

En la figura 19 se muestra el código obtenido al inspeccionar la página de DVWA Brute Force.

Se logran identificar parámetros tales como el tipo de método que tiene definido: **GET**, el cual indica que espera datos de entrada. Y luego se observan tales datos, que son el username y la password. Todos los parámetros de interés están encerrados en una casilla roja.

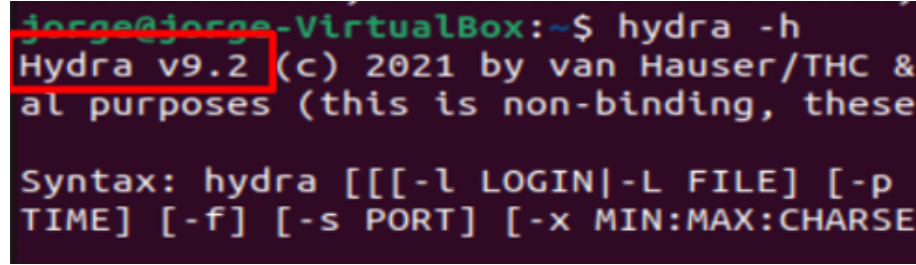
2.8. Utilización de curl por terminal (curl)

2.9. Demuestra 5 diferencias (curl)

2.10. Instalación y versión a utilizar (hydra)

```
jorge@jorge-VirtualBox:~$ sudo apt install hydra
[sudo] contraseña para jorge:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
hydra ya está en su versión más reciente (9.2-1ubuntu1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 123 no actualizados.
```

Figura 20: Instalación de Hydra.



```
jorge@jorge-VirtualBox:~$ hydra -h
Hydra v9.2 (c) 2021 by van Hauser/THC &
al purposes (this is non-binding, these

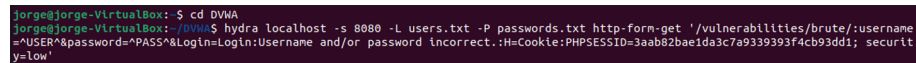
Syntax: hydra [[-l LOGIN|-L FILE] [-p
TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSE
```

Figura 21: Versión instalada de Hydra.

En la figura 20 se muestra el comando a utilizar para la instalación de Hydra. Simplemente se ejecutó el comando de **sudo apt install hydra**. Instalación que ya se había concretado previamente.

En la figura 21 se observa la versión instalada de Hydra, la cual corresponde a la versión **9.2**.

2.11. Explicación de comando a utilizar (hydra)



```
jorge@jorge-VirtualBox:~$ cd DVWA
jorge@jorge-VirtualBox:~/DVWA$ hydra localhost -s 8080 -L users.txt -P passwords.txt http-form-get '/vulnerabilities/brute/:username
=admin&password=^PASS^&Login=Login:Username and/or password incorrect.:H=Cookie:PHPSESSID=3aab82bae1da3c7a9339393f4cb93dd1; securit
y=low'
```

Figura 22: Utilización de Hydra.

En la figura 22 se observa el comando a utilizar de Hydra. Primero se localiza la carpeta de DVWA donde se encuentran los archivos de diccionarios a utilizar. Luego, se hace uso directo del comando Hydra. El primer campo es la dirección objetivo, en este caso el **localhost**. Luego se señala el puerto a actuar con el flag **'-s'**, que en este caso es el **8080**. Después se señalan los usuarios a probar mediante el flag **'-L'**, lo que corresponde a un archivo de texto, al igual que para las contraseñas, las cuales se identifican mediante un flag **'-P'**. Luego se indica el tipo de método el cual corresponde al GET, mediante el parámetro **http-form-get**. Finalmente se indica el 'sitio' donde actuará, el cual es en este caso la página del DVWA Brute Force, de directorio **'vulnerabilities/brute/'**, y se indican los valores a probar, que son identificados como **USER** y **PASS** en el **Login**. Se entrega de igual manera el PHPSESSID de la sesión.

2.12 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA



Figura 23: Diccionario de usuarios para Hydra.



Figura 24: Diccionario de contraseñas para Hydra.

En las figuras 23 y 24 se observan los archivos **users.txt** y **passwords.txt** que fueron entregados a Hydra como diccionarios de usuarios y contraseñas, respectivamente. **9.2.**

2.12. Obtención de al menos 2 pares (hydra)

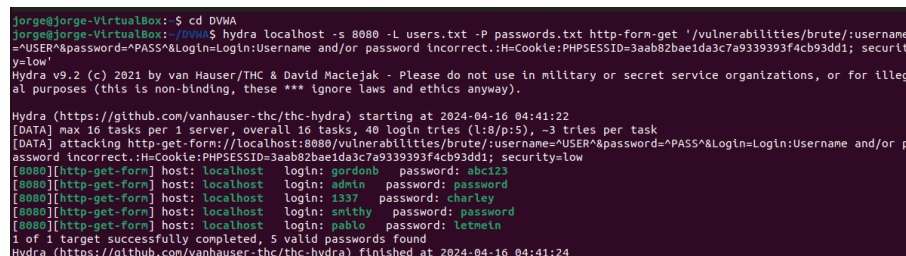


Figura 25: Obtención de pares válidos.

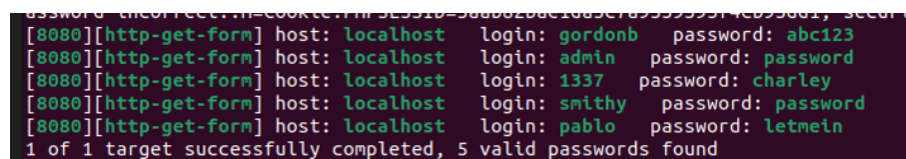


Figura 26: Obtención de pares válidos.

2.13 Explicación de la **DESARROLLO** (Orde) de ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Finalmente, en las figura 25 y 26 se logra observar la obtención de los pares correctos de usuario y contraseña. Siendo así exitoso el ataque de fuerza bruta mediante Hydra.

2.13. Explicación paquete curl (tráfico)

2.14. Explicación paquete burp (tráfico)

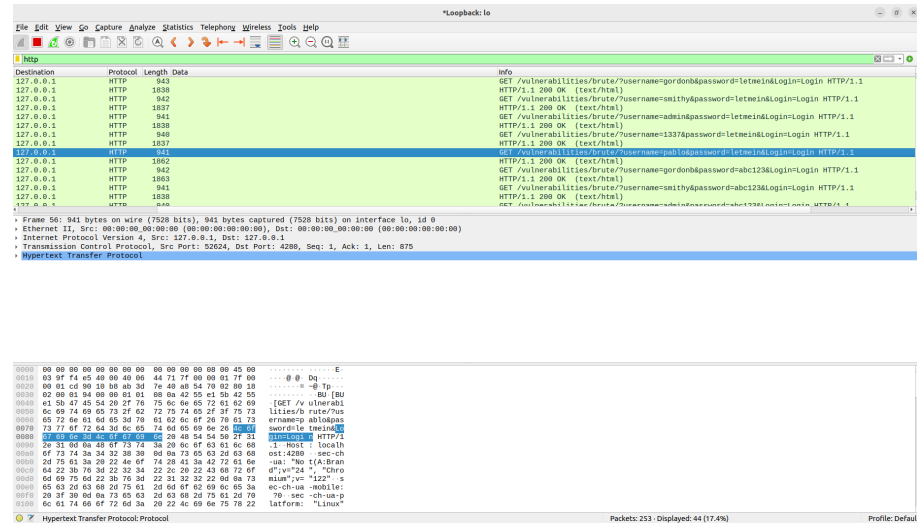


Figura 27: Tráfico de paquetes Burp.

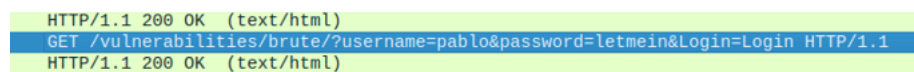


Figura 28: Tráfico de paquetes Burp. Campos.

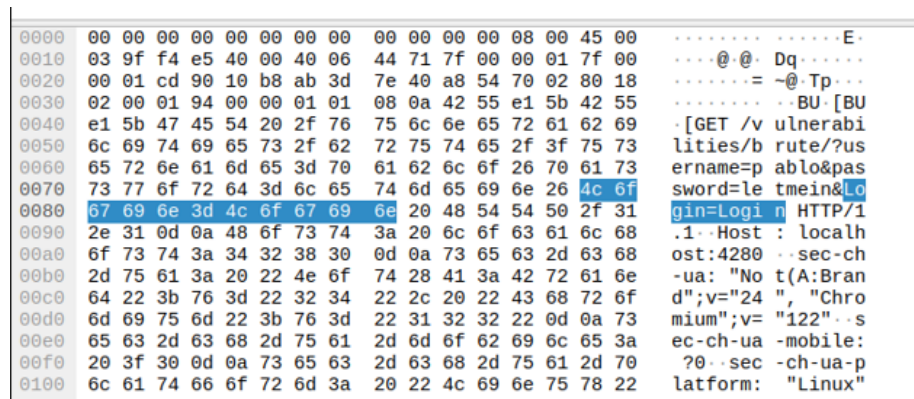


Figura 29: Tráfico de paquetes Burp. Data.

2.15 Explicación de paquetes Hydra (DVWA) ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

En la figura 27 se observa la captura de Wireshark del tráfico generado por el ataque Cluster bomb realizado por medio de Burpsuit. En las figuras 28 y 29 se pueden observar los distintos campos de los paquetes HTTP generados, donde se evidencia la nula seguridad de DVWA ya que la información de las credenciales es mostrada en texto plano en la info y en la data de los paquetes. Se puede observar también que fueron enviados como texto html y recibidos mediante un método GET dentro de la instancia del localhost, siendo la correspondiente dirección ip 127.0.0.1 tanto el origen como el destino.

2.15. Explicación paquete hydra (tráfico)

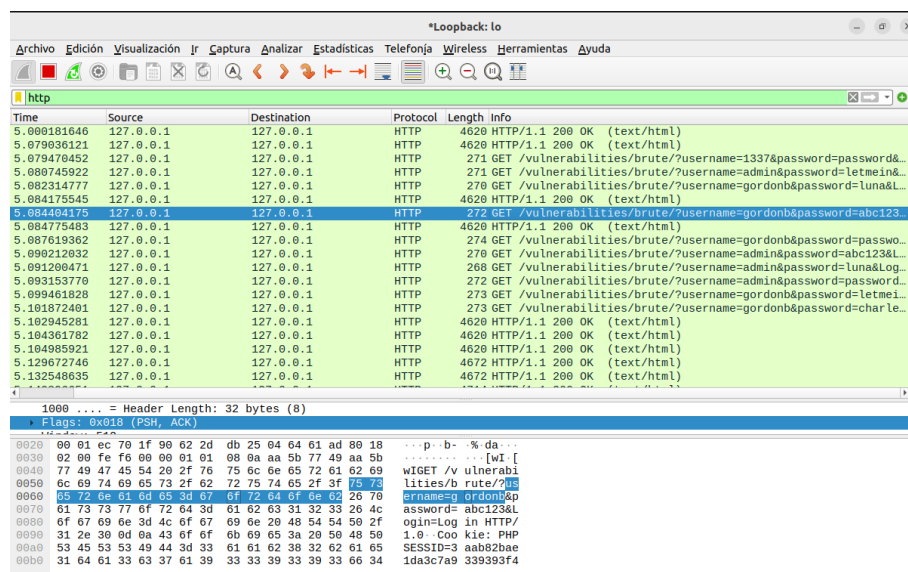


Figura 30: Tráfico de paquetes Hydra

HTTP	4620	HTTP/1.1	200	OK	(text/html)
HTTP	272	GET	/vulnerabilities/brute/?username=gordonb&password=abc123...		
HTTP	4620	HTTP/1.1	200	OK	(text/html)

Figura 31: Tráfico de paquetes Hydra. Campos.

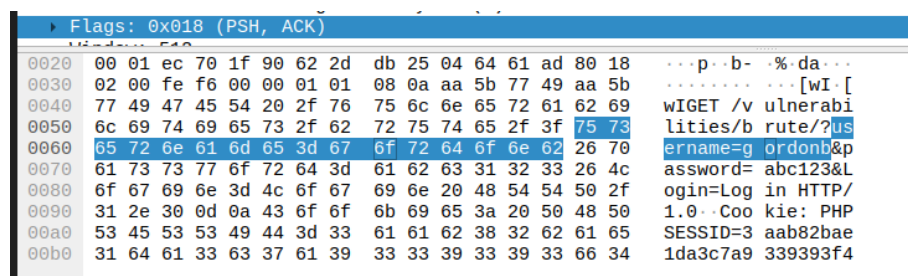


Figura 32: Tráfico de paquetes Hydra. Data.

Por último, en la figura 30 se muestra el tráfico generado por el ataque de fuerza bruta de Hydra. Nótese en las figuras 31 y 32 que los paquetes son muy similares a los que fueron generados por el ataque de Burpsuite, pues la información es enviada en formato texto html y es recibida mediante un método GET, donde una vez más se evidencia la nula seguridad, pues en los campos de Info y Data de los paquetes generados por Hydra se revelan todas las credenciales utilizadas en texto plano.

El tráfico es capturado a través de la interfaz **lo** de Wireshark, al ser transmitido dentro del loopback / localhost, con la misma ip de origen y destino, 127.0.0.1.

2.16. Mención de las diferencias (tráfico)

2.17. Detección de SW (tráfico)

Conclusiones y comentarios

Como comentario no se ha logrado realizar el completo estudio del método CURL, pero se puede concluir que los ataques por fuerza bruta de Burpsuite e Hydra son muy similares, pues se hizo el estudio de ambos casos y se obtuvieron resultados ejemplares tanto en el tráfico como en los casos exitosos. Se logró descubrir las credenciales válidas utilizando ambos métodos basados en la fuerza bruta. Gracias a ello hemos sido capaces de identificar parámetros que son influyentes en la seguridad de cada sitio y el cómo manipularlos para burlar la seguridad de una red o sistema. Esto es de utilidad ya que teniendo el conocimiento base de estas técnicas se es capaz de mejorar la seguridad de un sitio.