M1.870 - Fundamentos de ciberseguridad.

José Enrique Rodríguez González.

PRACTICA 2.

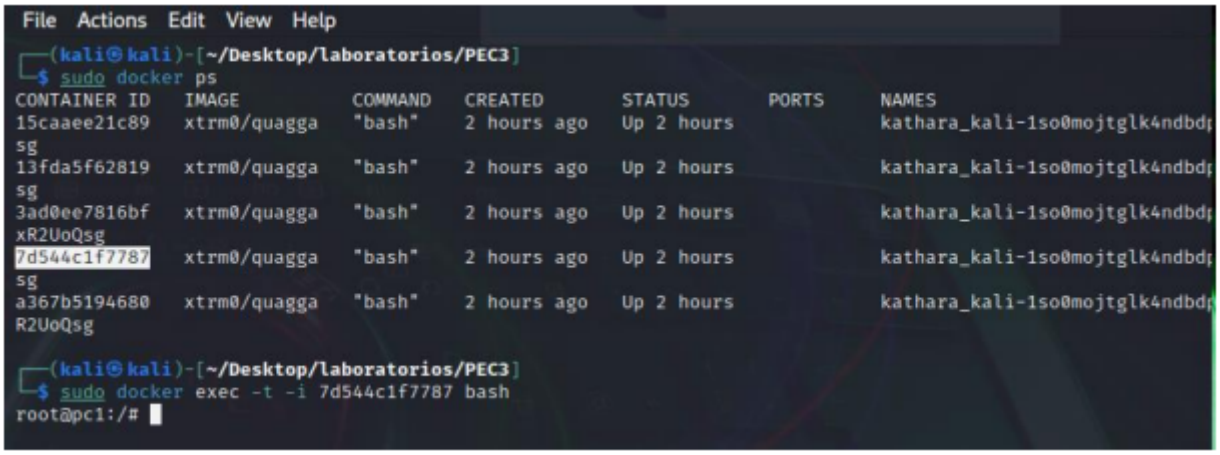
Indice.

Enunciado.

En esta segunda y última práctica usaremos también el mismo entorno de Kathará de la PEC 3 en algunos de los ejercicios (no en todos). Antes de continuar debemos de realizar algunas puntualizaciones:

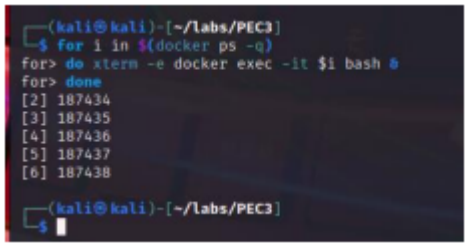
- Para alguna de las actividades posteriores será necesario iniciar el entorno de Kathará en modo privilegiado. Por este motivo se pidió en la PEC3 anterior como pregunta. Se puede realizar de la siguiente manera:

```
sudo kathara lstart --privileged

sudo docker ps

sudo docker exec -t -i identificador_container_ bash
```



También es posible abrir todos los terminales directamente con el bucle siguiente escrito en Bash:

```
for i in $(docker ps -q)

do xterm -e docker exec -it $i bash &

done
```



- Comentar que Kathará no tiene persistencia así que recomendamos no cerrar el laboratorio y guardar el estado de la máquina virtual.

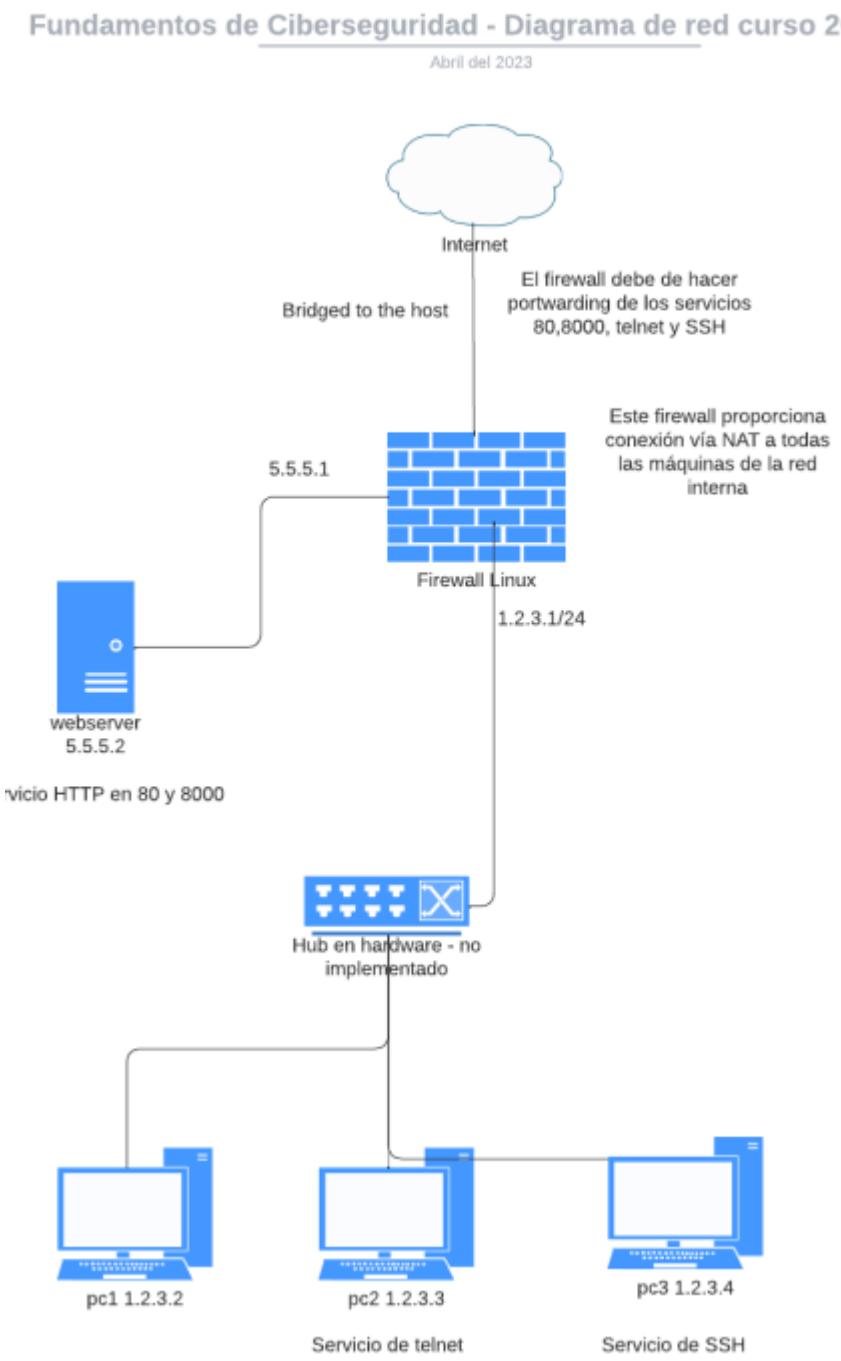
- Otra puntualización es que, para que las máquinas internas de la red puedan acceder correctamente a Internet deberemos de suprimir la regla siguiente en la máquina firewall[1].

> [1]: Este es el primer semestre que usamos Kathará y nos estamos encontrando con situaciones diversas. Estamos en contacto con los desarrolladores para profundizar un poco más. Por ejemplo: https://github.com/KatharaFramework/Kathara/issues/211.

```
iptables -t nat -D PREROUTING -p tcp --dport 80 -j DNAT --to 5.5.5.2
```

Esta regla eliminará la redirección del puerto 80 desde la máquina Kali. Si deseamos acceder desde la máquina Kali a este puerto podemos añadirla de nuevo con:

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to 5.5.5.2
```



> Añadimos nuevamente el diagrama de red implementado a modo de recordatorio. Es el mismo que el de la PEC3.

Volver al indice.

Actividad.

Demostrad con capturas de pantalla de vuestro entorno las operaciones siguientes y responded a todas las preguntas.

Nota: aseguraros que aparezca en la shell vuestro nombre de usuario de la UOC en todas las capturas de pantalla. Podéis utilizar el comando:

```
export PS1=(gfarrasb)$PS1
```

Adicionalmente proporcionad una explicación para justificar qué realiza cada captura:

Volver al indice.

Pregunta 1.

El administrador de red, conjuntamente con la Dirección de la empresa, han considerado que sería una buena idea realizar una acción de concienciación de phishing para todos trabajadores y empleados. Implementad una campaña de concienciación de phishing con el programa `Gophish`. Esta actividad está fuera de Kathará y, de hecho, podéis hacerla en el entorno que queráis. Demostrad que al menos un trabajador cae en el ataque de phishing y podéis ver su usuario y contraseña **(1.5 puntos)**.

Volver al indice.

Pregunta 2.

El administrador de red quiere investigar si es posible lanzar un ataque de persistencia de clave SSH mediante `metasploit`[2] contra PC3 (se trata, en concreto, del módulo llamado SSH Key persistence - This module will add an SSH key to a specified user to allow remote login via SSH at any time) Aseguraros que, aunque user3 cambie la contraseña, el administrador podrá acceder igual a la máquina mediante la clave generada vía este módulo de metasploit **(1.5 puntos)**.

> [2]: Metasploit es una herramienta incorporada ya en Kali Linux que permite ejecutar exploits conocidos contra una máquina remota. No es necesario instalar nada ya que Kali la lleva por defecto.

Volver al indice.

Pregunta 3.

El administrador de red ha descubierto que el servidor web de main.py contiene un bug que permite un Directory Transversal Attack. Se os pide arreglar el código de este servidor web escrito en Python para solucionarlo. El servidor web debería de poder entregar solamente ficheros de la carpeta de donde reside. **(1 punto)**.

Volver al indice.

Pregunta 4.

Ahora resulta que el usuario de PC2 está enfadado contra el usuario de PC3 así que decide atacarlo. Ejecutará un ataque de DNS Spoofing contra una de las webs preferidas del usuario de PC3. Implementad un ataque de DNS spoofing usando la herramienta de terminal `ettercap` contra PC3 que use el dominio que consideréis. **(2 puntos)**

Volver al indice.

Pregunta 5.

El usuario de PC3 nota que algo pasa en su ordenador y empieza a defenderse. Aplicará las siguientes protecciones:

1. Quiere evaluar el uso de unidades cifradas. Demostrad que podéis crear una disco cifrado mediante la herramienta `zulucrypt`. Esta actividad está fuera de Kathará e incluso de PC3. Implementadlo en la Kali nativa proporcionada. **(1 punto)**.

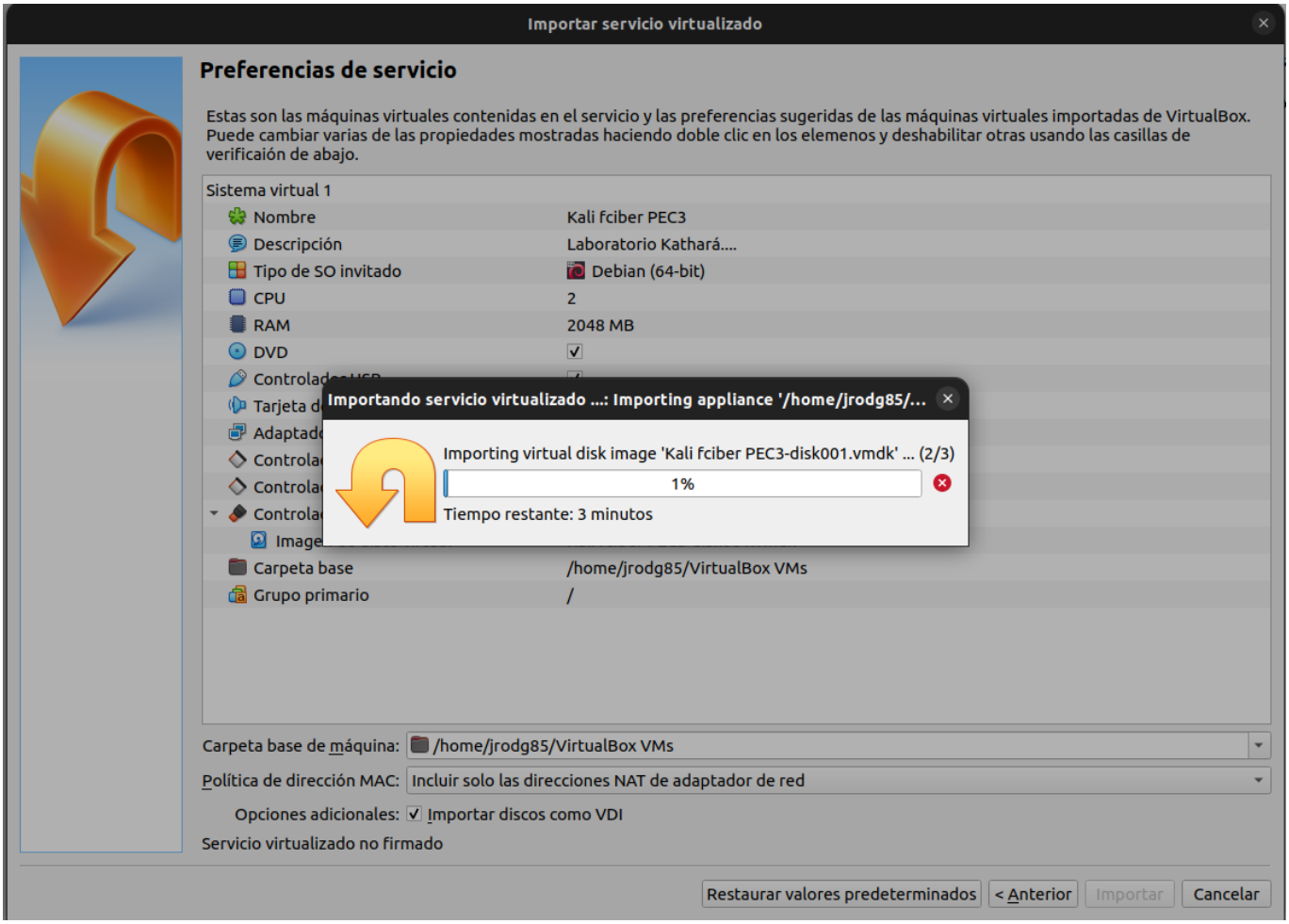
2. Quiere detectar mediante snort todas las conexiones SYN que entren en su máquina y ver una alerta por pantalla.**(1 punto)**.

3. Como ya duda de sus compañeros aplicará una regla en el cortafuegos (concretamente, usará la herramienta de terminal iptables) que solamente permita conexiones SSH desde fuera, no desde las máquinas internas (ni desde PC1 ni desde PC2, solamente desde firewall). **(1 punto)**.

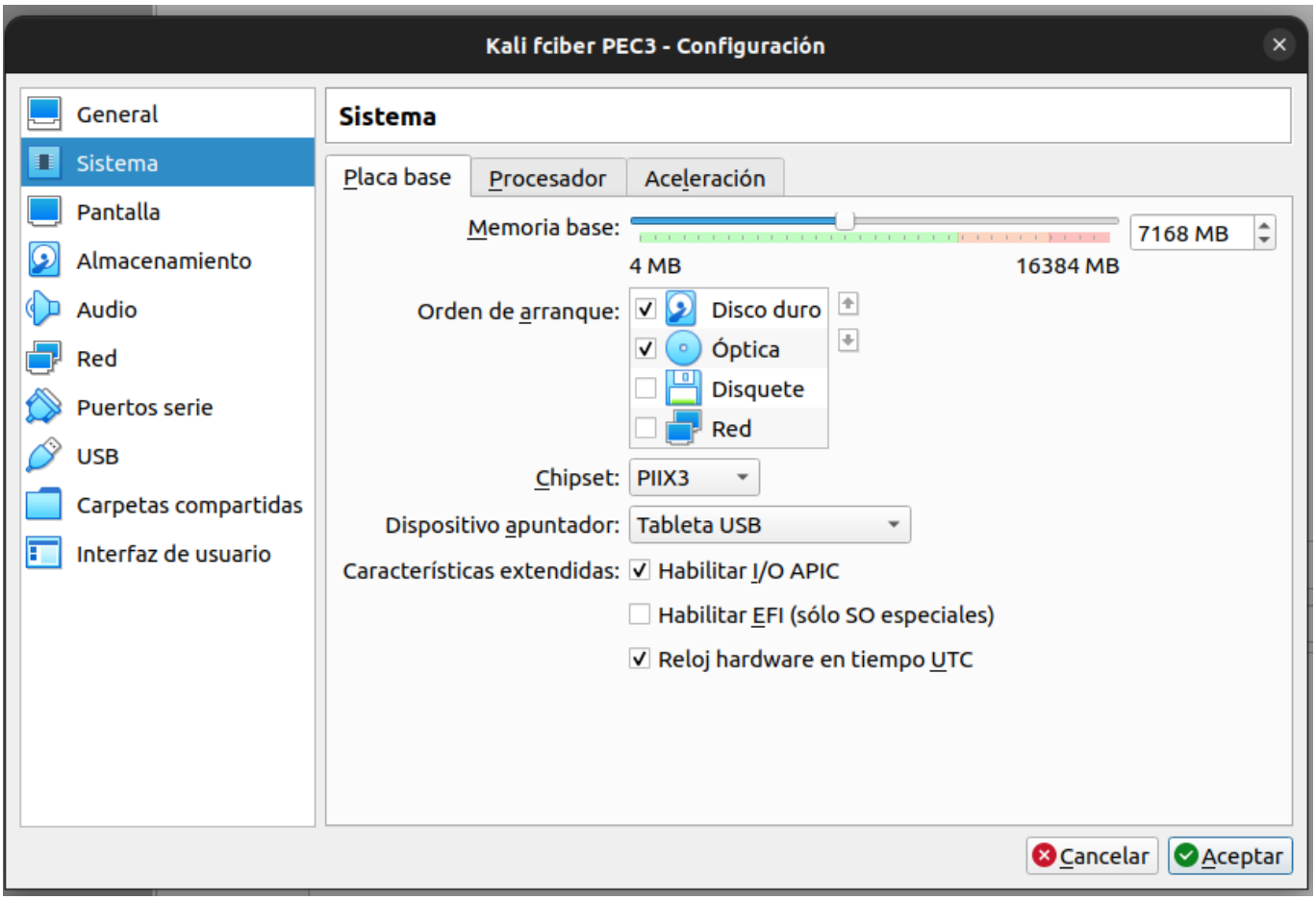
Volver al indice.

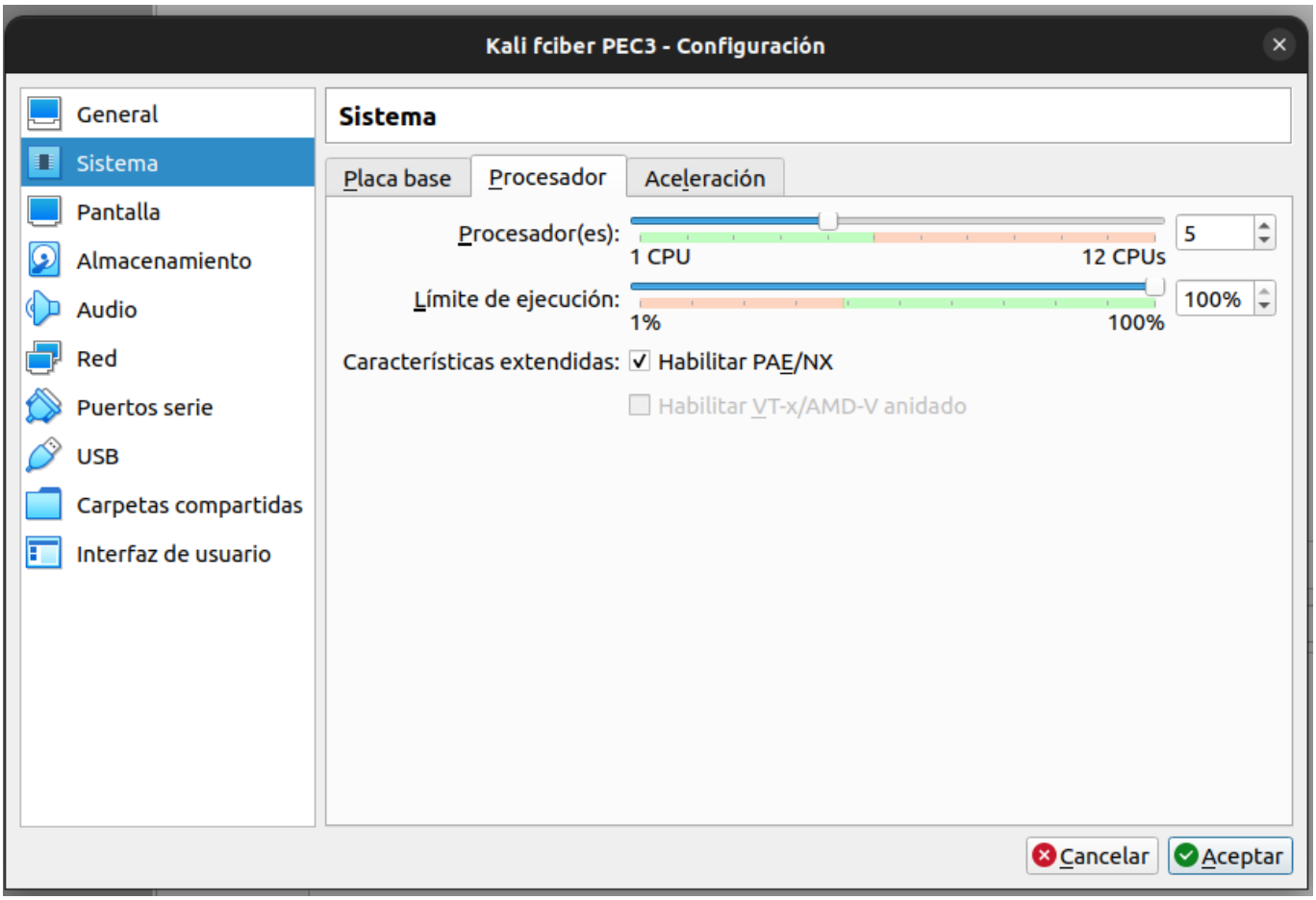
Preparación del enunciado y de la actividad.

Procedemos a importar la VM de la PEC 3.



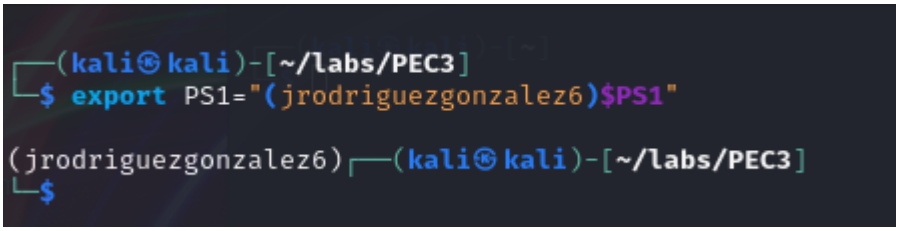
Una vez importado, ya que la maquina huésped que dispongo tiene bastante potencia, procedo a aprovecharla ampliando RAM y uso de CPU's





Procedo a abrir la terminal desde la carpeta de la PEC 3 y ejecuto el siguiente comando que se indica en la preparación de la actividad, de modo que los pantallazos son personalizados.

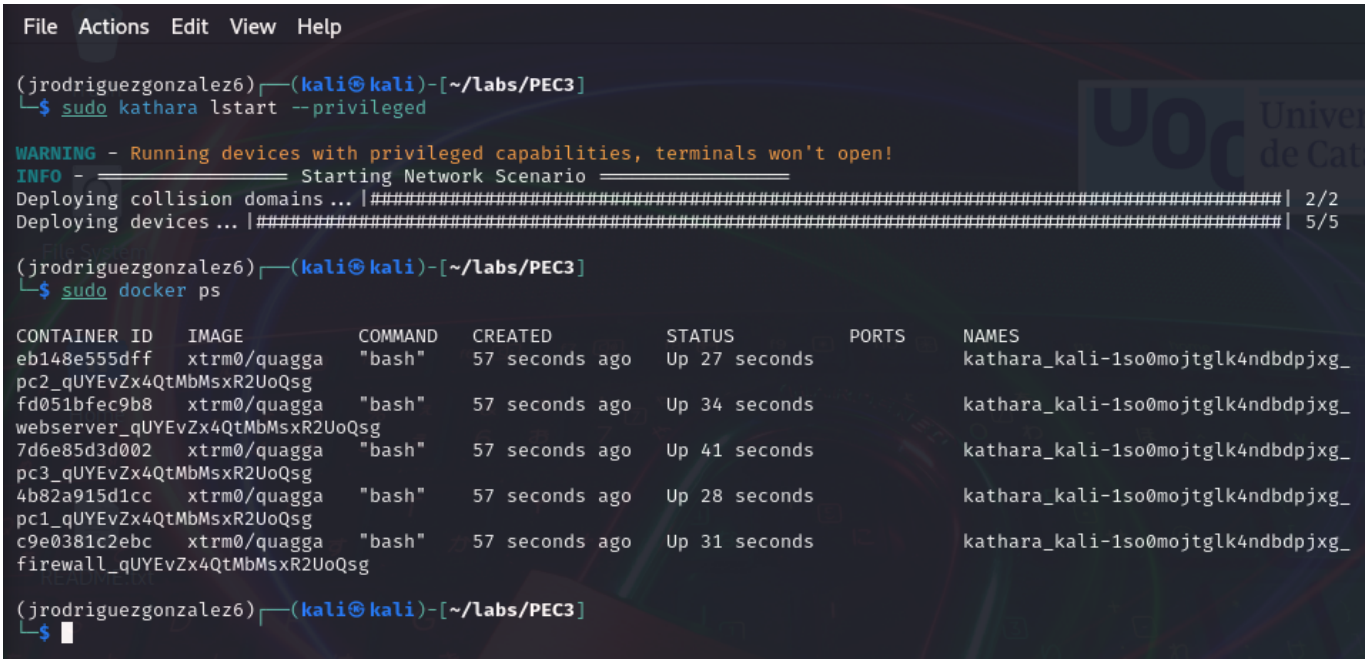
```
export PS1="(jrodriguezgonzalez6)$PS1"
```



Procedemos a iniciar el entorno de Kathará en modo privilegiado, ejecutando en terminal los siguientes comandos:

```
sudo kathara lstart --privileged

sudo docker ps
```

El tercer comando lo considero innecesario realizarlo porque debajo del container id dice quien es cada uno.



- En el container id `eb148e555dff`podemos ver la siguiente etiqueta de names kathara_kali-1so0mojtglk4ndbdpjxg_**pc2**_qUYEvZx4QtMbMsxR2UoQsg
- En el container id `fd051bfec9b8`podemos ver la siguiente etiqueta de names kathara_kali-1so0mojtglk4ndbdpjxg_**webserver**_qUYEvZx4QtMbMsxR2UoQsg
- En el container id `7d6e85d3d002`podemos ver la siguiente etiqueta de names kathara_kali-1so0mojtglk4ndbdpjxg_**pc3**_qUYEvZx4QtMbMsxR2UoQsg
- En el container id `4b82a915d1cc`podemos ver la siguiente etiqueta de names kathara_kali-1so0mojtglk4ndbdpjxg_**pc1**_qUYEvZx4QtMbMsxR2UoQsg

- En el container id `c9e0381c2ebc`podemos ver la siguiente etiqueta de names kathara_kali-1so0mojtglk4ndbdpjxg_**firewall**_qUYEvZx4QtMbMsxR2UoQsg

De este modo, quedan correctamente identificados, para entrar en la consola de cada uno de ellos procederemos según lo indicado en el enunciado.

Volver al indice.

Respuesta a la pregunta 1.

Introducción.

Para cumplimentar este ejercicio, debo implementar una campaña de concienciación de phishing utilizando el programa Gophish. Nos aseguraremos de seguir las políticas y regulaciones éticas y legales cuando realices este tipo de actividades, como en este caso, estamos realizando una acción conjunta con la Dirección de la empresa, presumimos que tenemos la venia para realizarlo.

En esta pregunta desarrollaré los siguientes pasos:

1. Instalación de Gophish.
2. Configuración de la campaña.
3. Ejecución de la campaña y recolección de datos

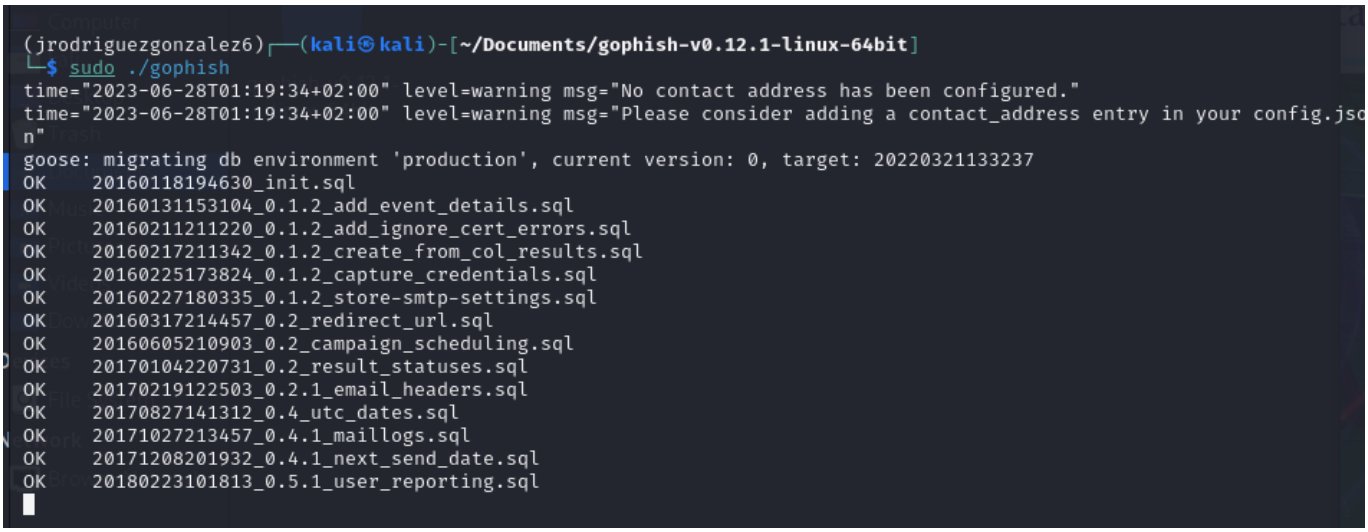
Instalación de Gophish.

- Instalaremos en la VM de kali, `Gophish` desde la web oficial de Gophish

- Hacemos click en descargar y nos redirige al repositorio github de Gophish

- Descomprimimos y copiamos la carpeta en Documentos, para este paso, nos he procedido a buscar en YouTube un tutorial relativo a la instalación de `Gophish` usando el siguiente enlace de YouTube.
- Damos permiso para poder ejecutar el binario con `sudo chmod +x gophish`

- Por ultimo ejecutamos el binario

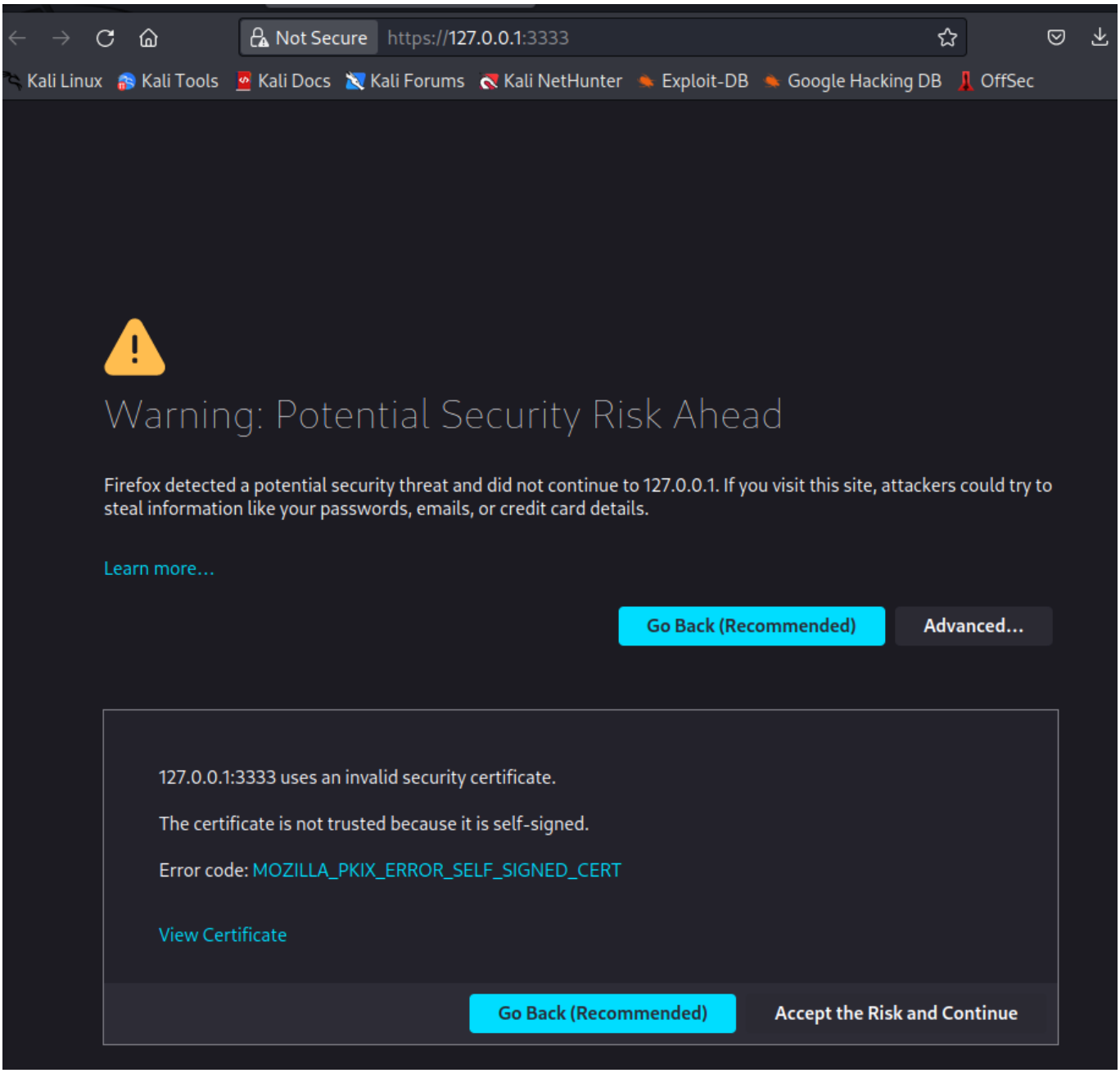


En el momento que nos aparezca en la consola `level=info msg="Starting admin server at https://127.0.0.1:3333"`, significa que `Gophish` se está ejecutando correctamente.

Configuración de la campaña.

Para estas acciones, seguiremos usando como fuente el video de YouTube anteriormente indicado.

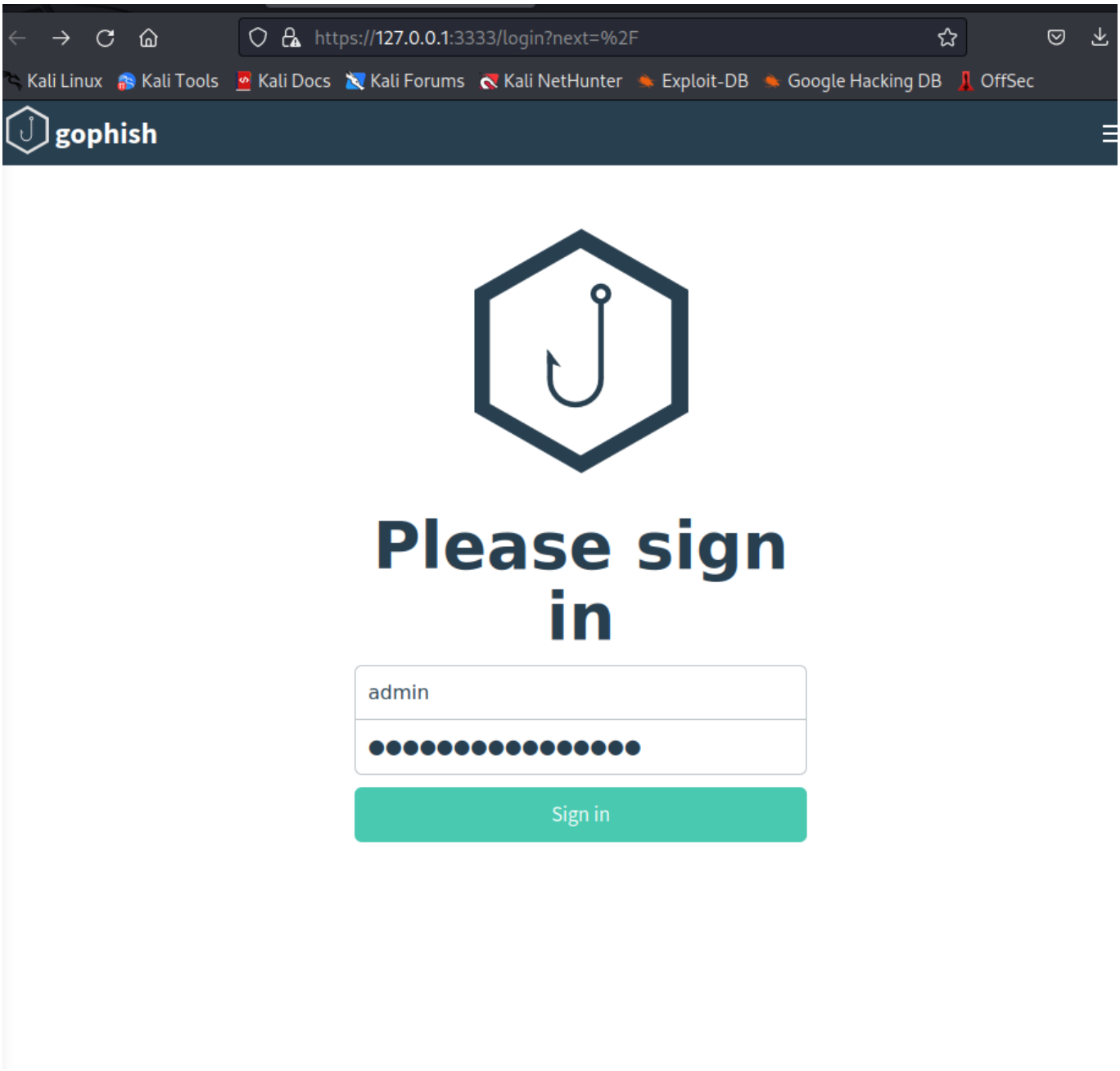
Una vez instalado el programa procederemos con la campaña, para ello primero de ello accederemos a la web `https://127.0.0.1:3333`, aceptaremos los riesgos que indican en la pantalla.



A continuación entraremos en el login del la aplicación, para saber el admin y pass nos iremos a unos de los mensajes info que tenemos en la consola con la que hemos ejecutado `Gophish` que nos dice lo siguiente:

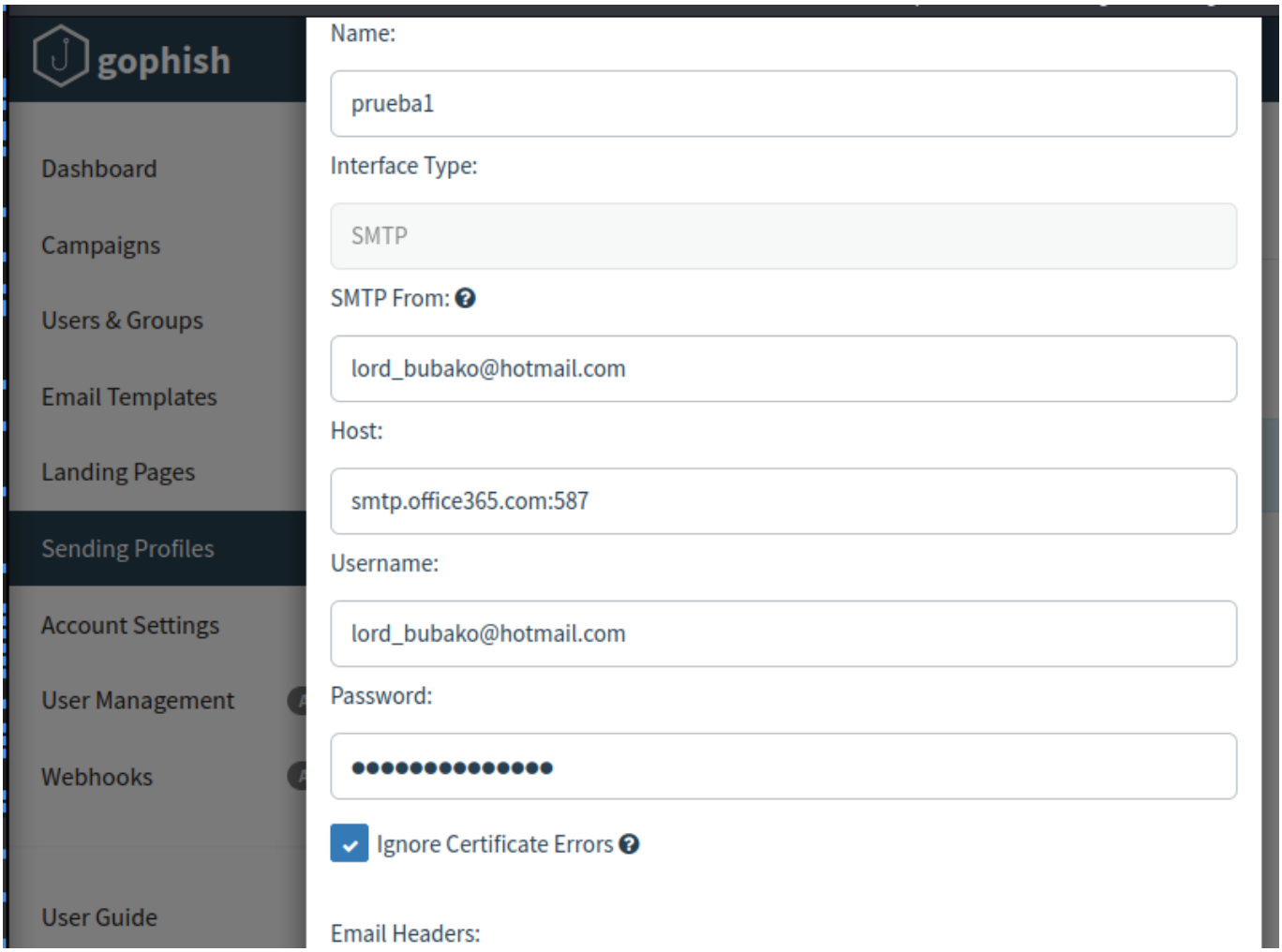
```
level=info msg="Please login with the username admin and the password
a866395e00f77bcd"
```

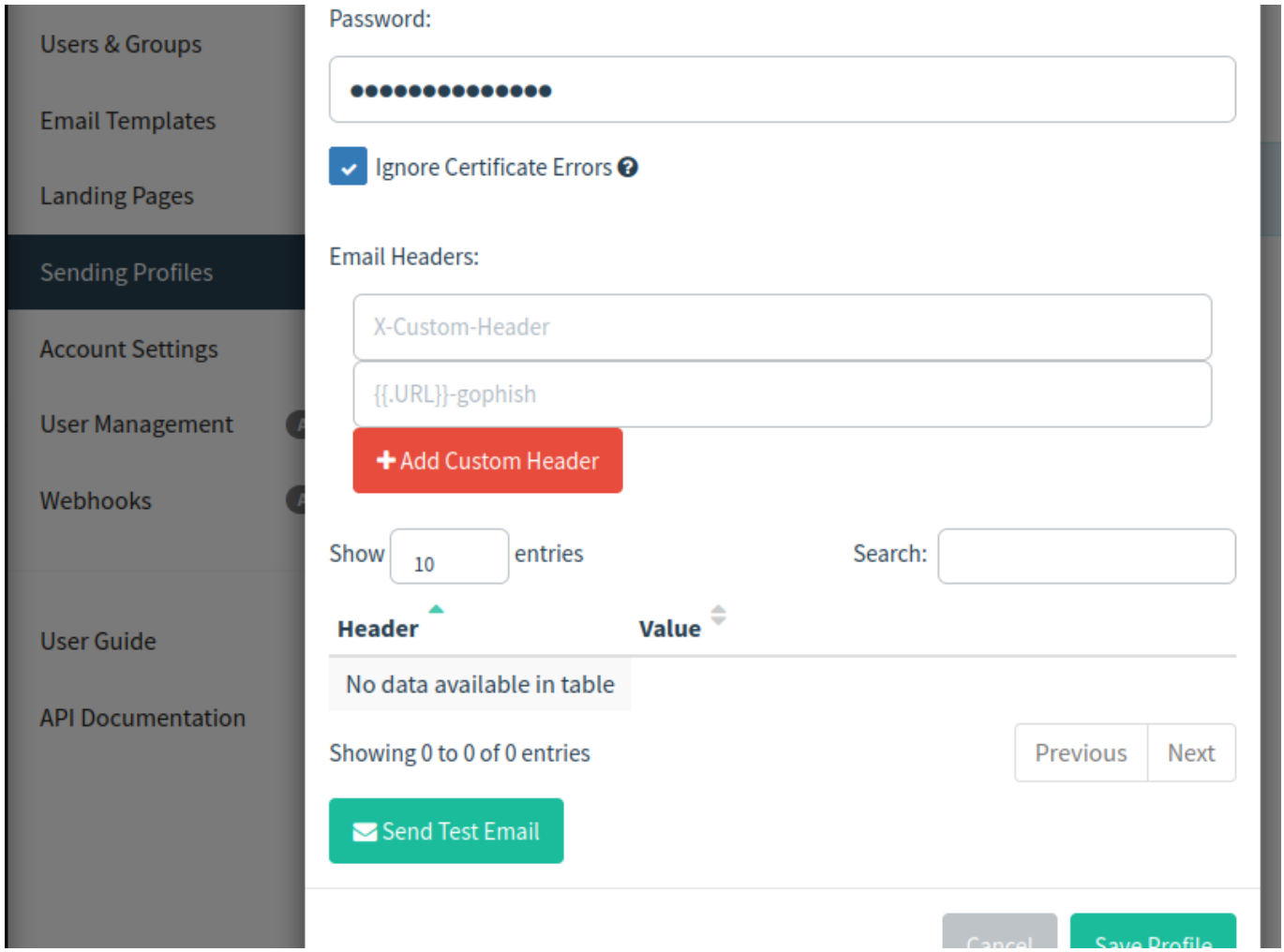
- Procedemos a ingresar user y pass.



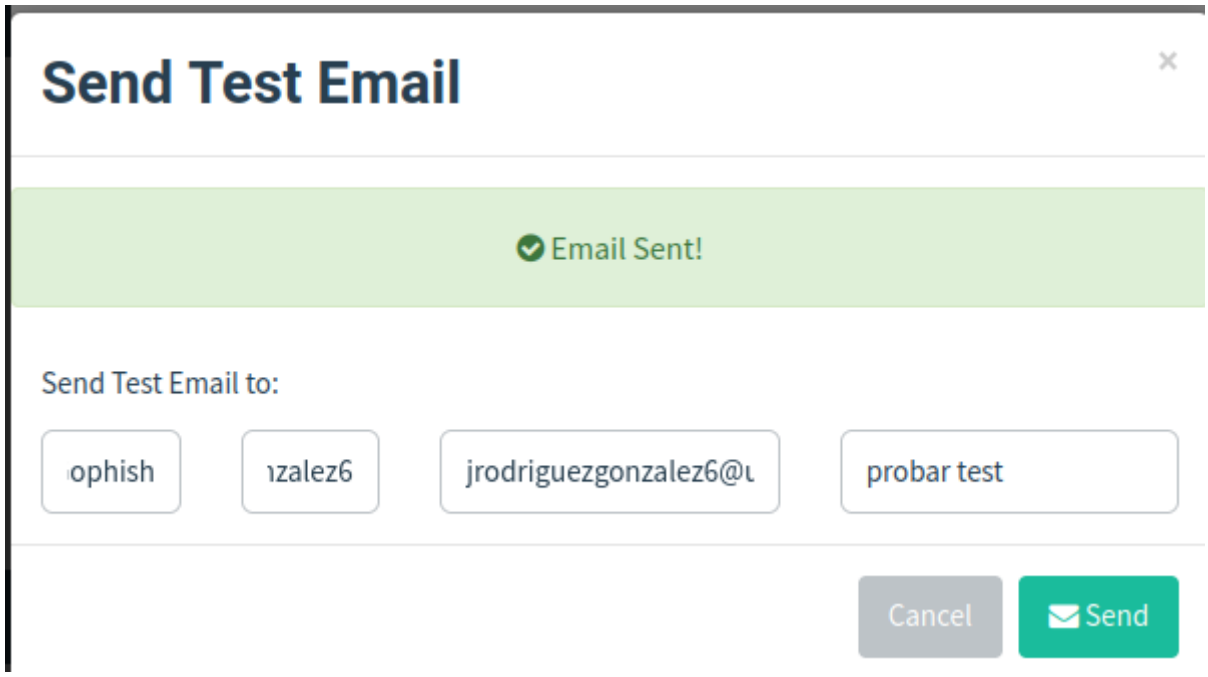
Procedemos a cambiar la contraseña a otra nueva

Una vez accedido a la aplicación, se procede a realizar el perfil falso, para ello entramos en `sending profile`, una vez dentro procedemos a rellenar los datos. En este caso usaremos un mail de Hotmail, como prueba, introducimos la configuración tal y como indica en la siguiente imagen.

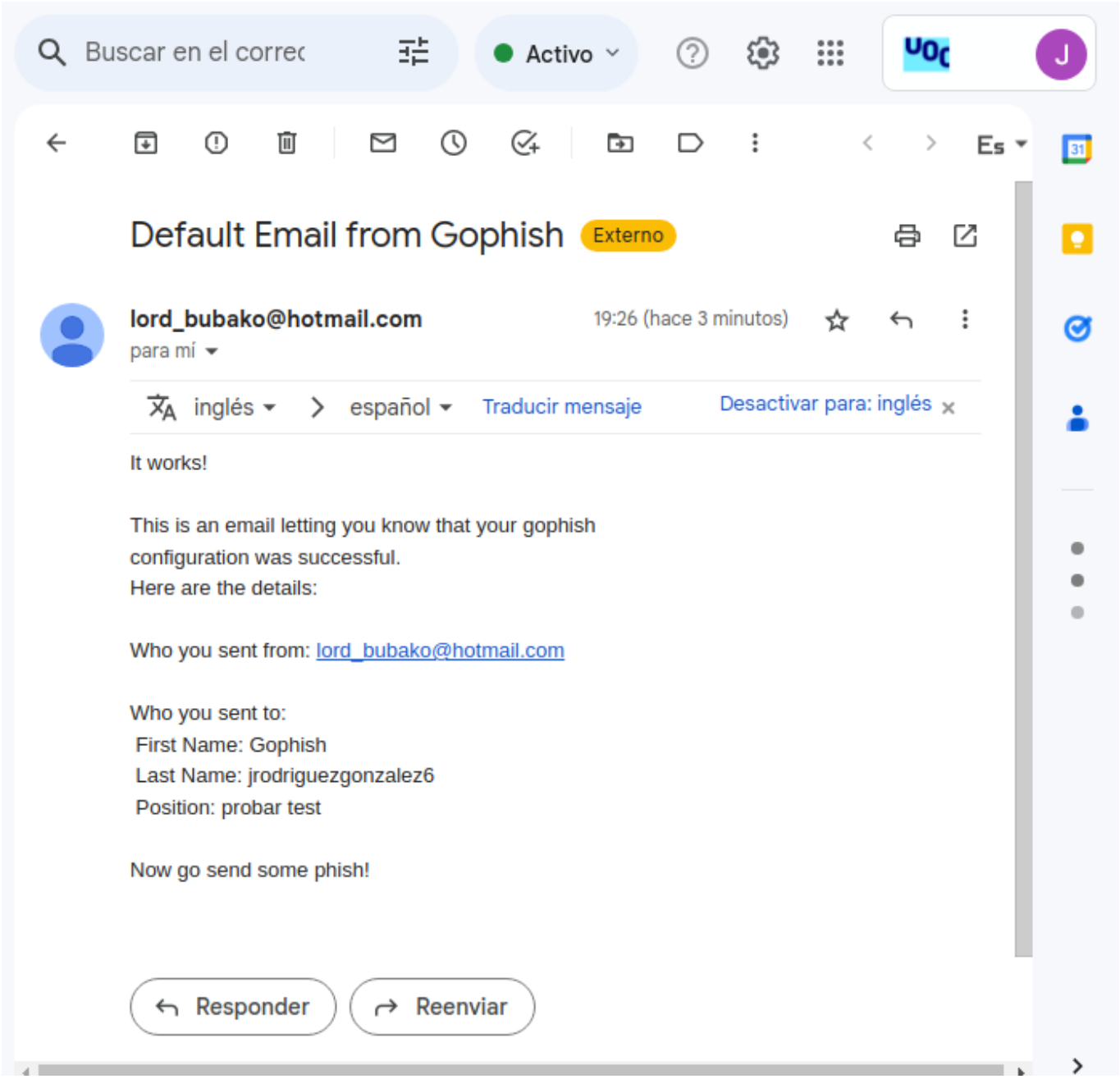

gophish
Dashboard
Campaigns
Users & Groups
Email Templates
Landing Pages
Sending Profiles
Account Settings
User Management
Webhooks
User Guide
Name:
prueba1
Interface Type:
SMTP
SMTP From:
lord_bubako@hotmail.com
Host:
smtp.office365.com:587
Username:
lord_bubako@hotmail.com
Password:
Ignore Certificate Errors
Email Headers:



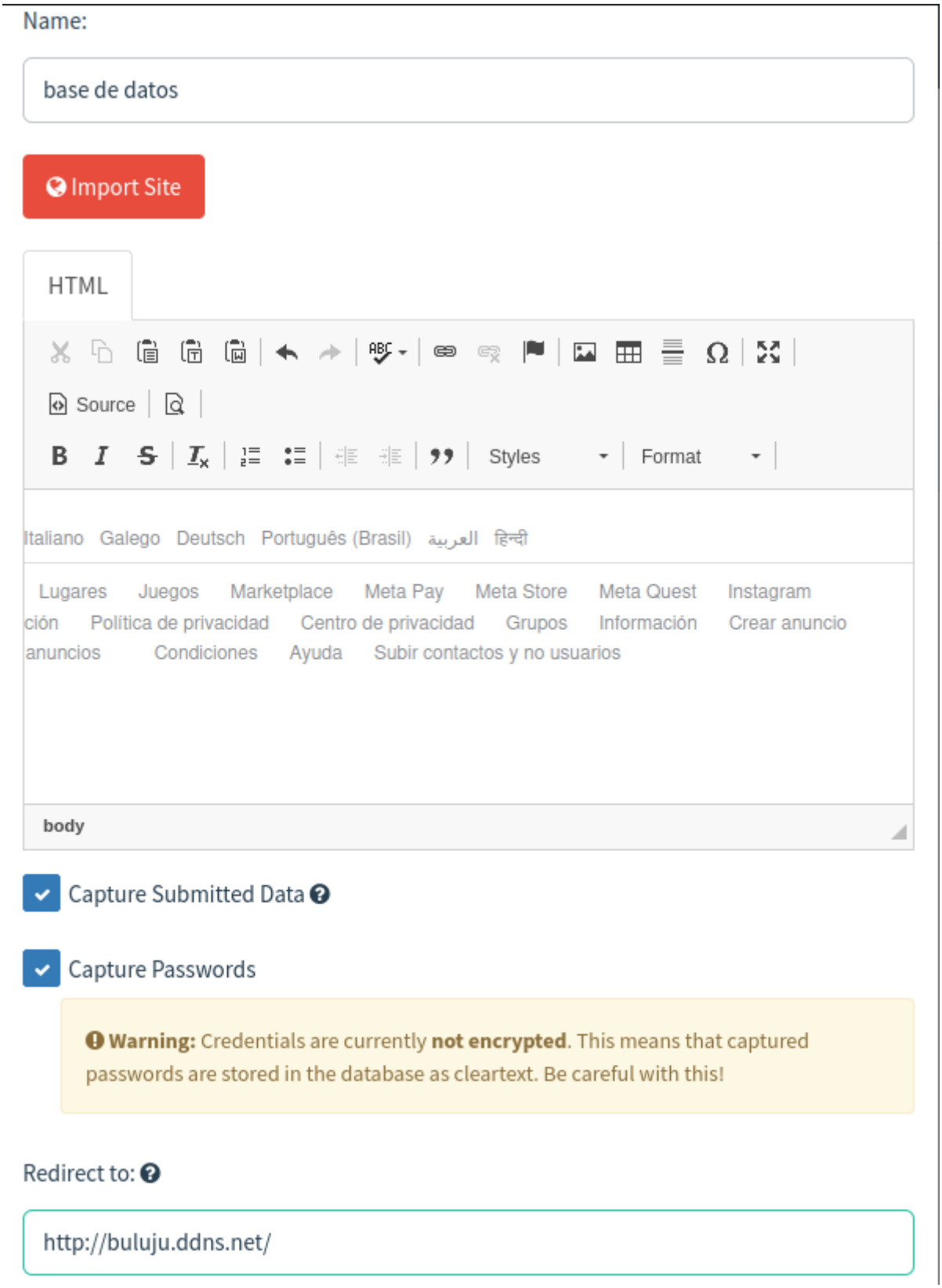
Ahora pasaremos a realizar un mail de prueba para comprobar que todo funciona. En este caso procederemos a realizarlo a la cuenta personal de email de la UOC. Rellenamos los datos y hacemos click en enviar.



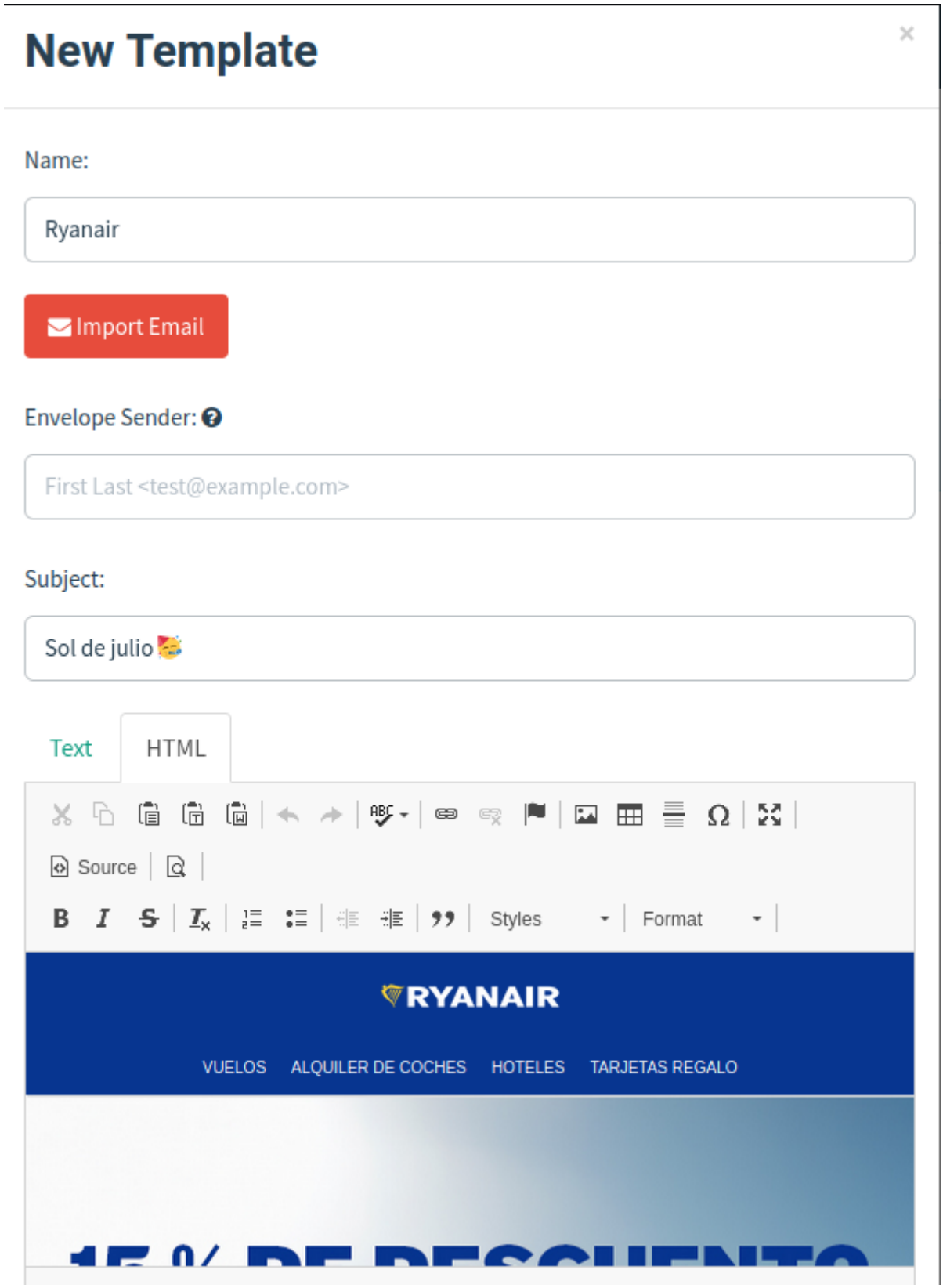
Dentro del nuestro buzón de la UOC, se puede ver el correo recibido.

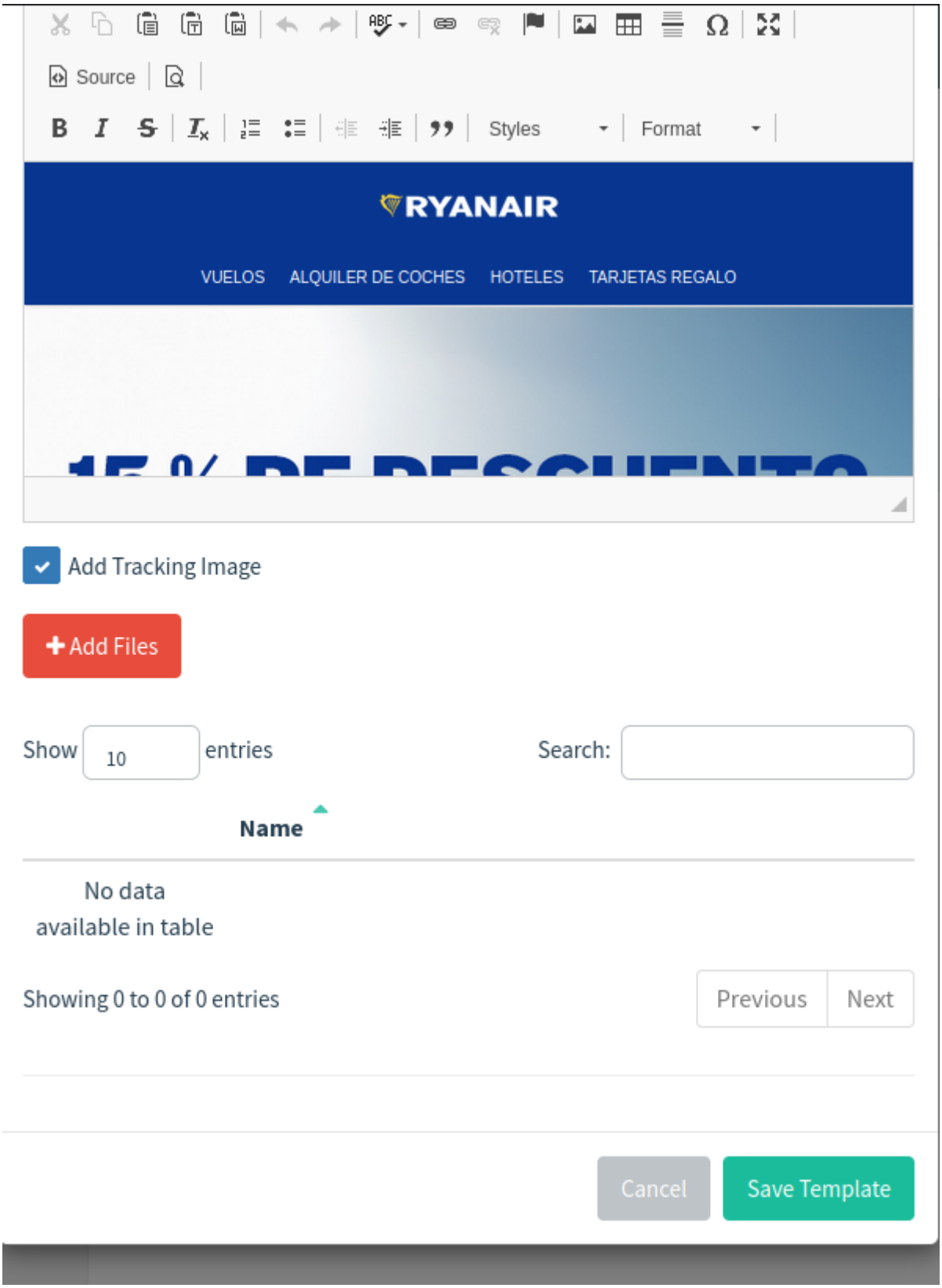


A continuación pasamos al apartado de landing page, que es como crear la `website fake`, hacemos crear una nueva pagina. Lo vamos a configurar para que de tal manera simule una pagina web de Facebook, la victima ingresará unos datos y posteriormente redirigirá a otra web, la cual simulará nada mas que un front que solo diría, has sido victima de un phishing de tu empresa. Tendremos activado los tic de capturar password y la información de inicio de sesión.

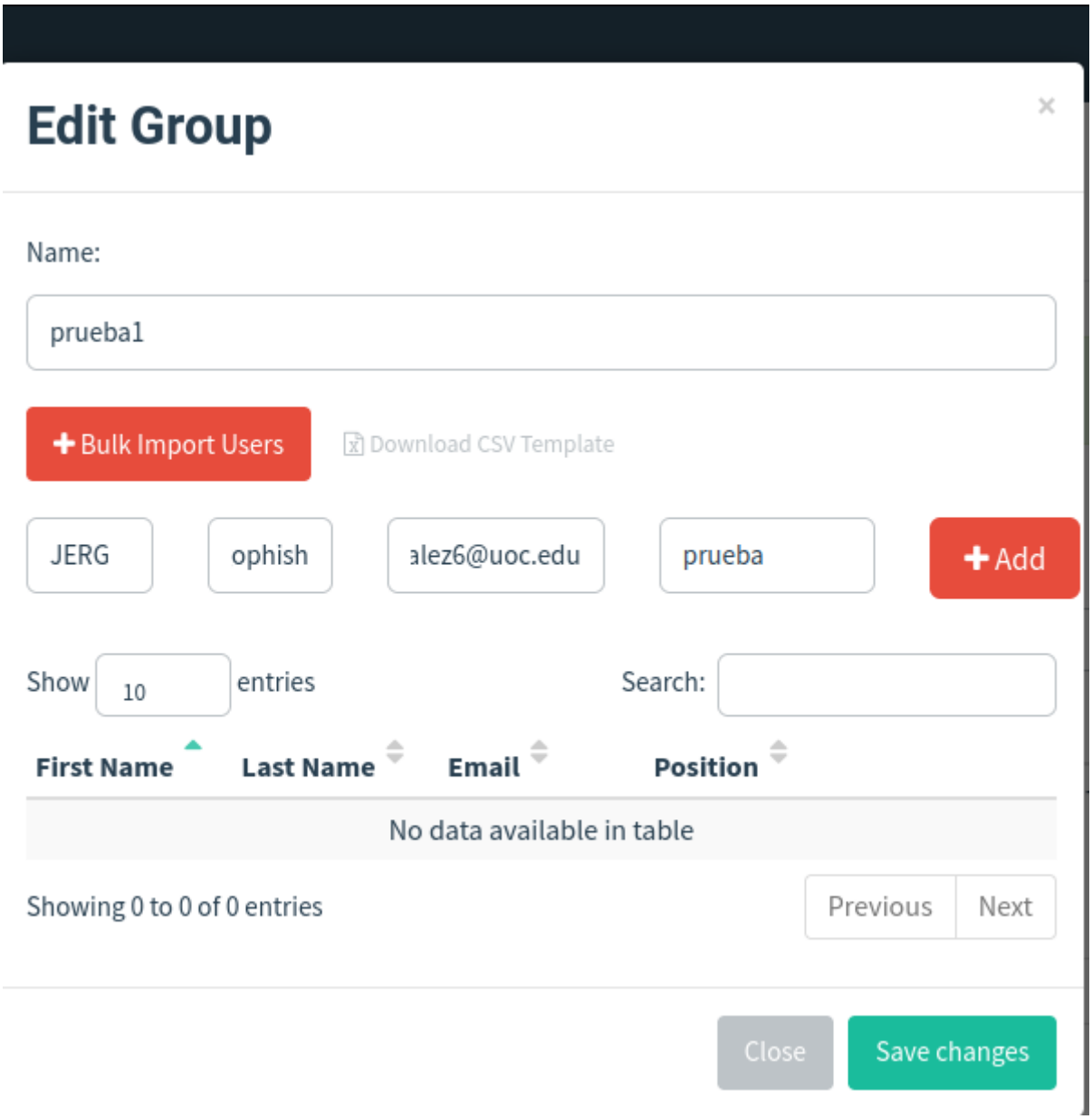


A continuación procederemos a hacer una plantilla de email, en este caso voy a usar una plantilla de ofertas de Ryanair. Para ello, lo hemos importado desde el original recibido. Activaremos el tick de add tracking Image y procederemos a guardar.


New Template
Name:
Ryanair
Import Email
Envelope Sender:
First Last <test@example.com>
Subject:
Sol de julio
Text
HTML
Source
Styles
Format
RYANAIR
VUELOS
ALQUILER DE COCHES
HOTELES
TARJETAS REGALO

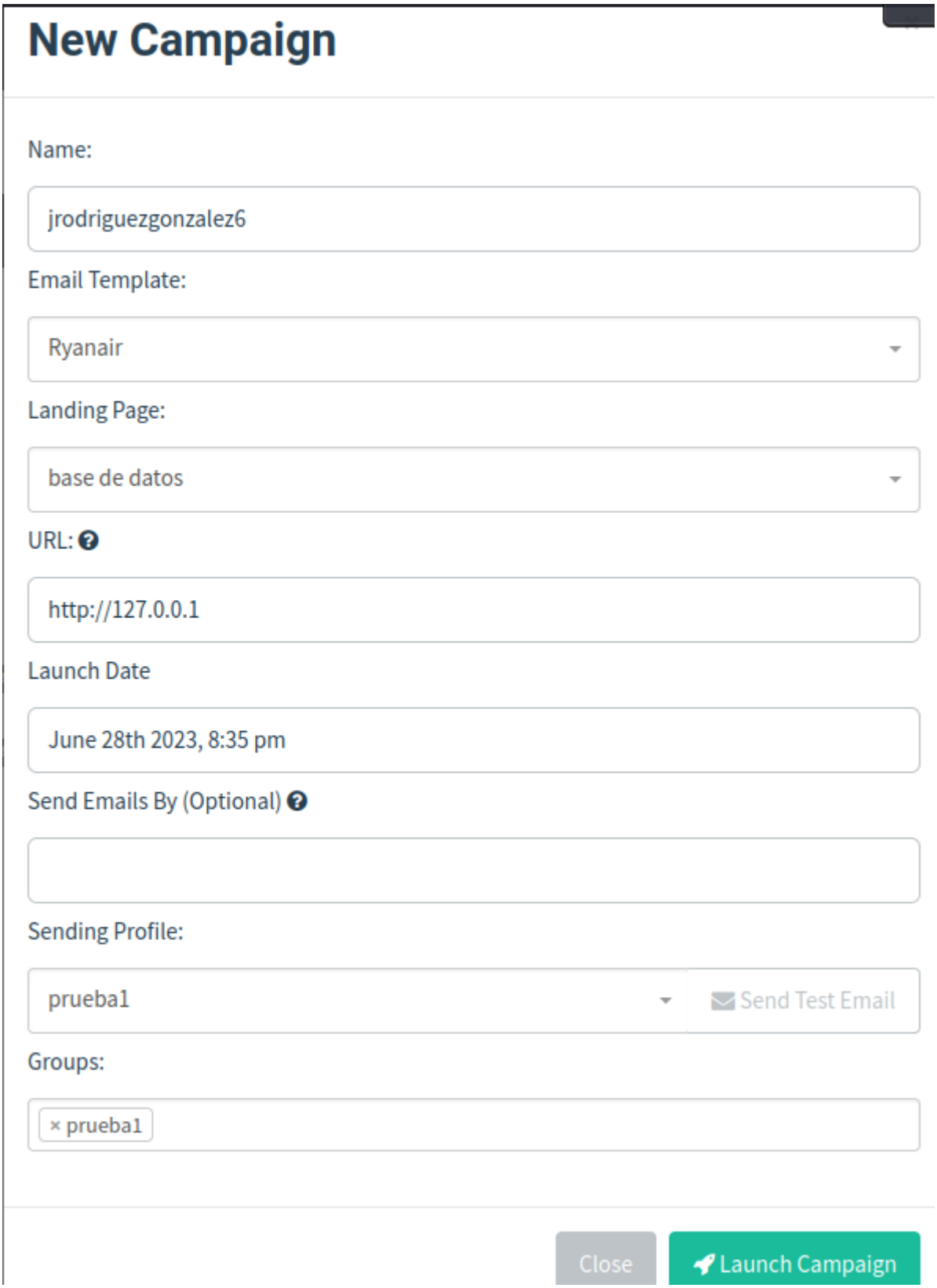


Por ultimo añadiremos el grupo de correo, en nuestro caso, volveremos a mandarlo al correo de la UOC. Procedemos a rellenar los datos y guardamos los cambios.



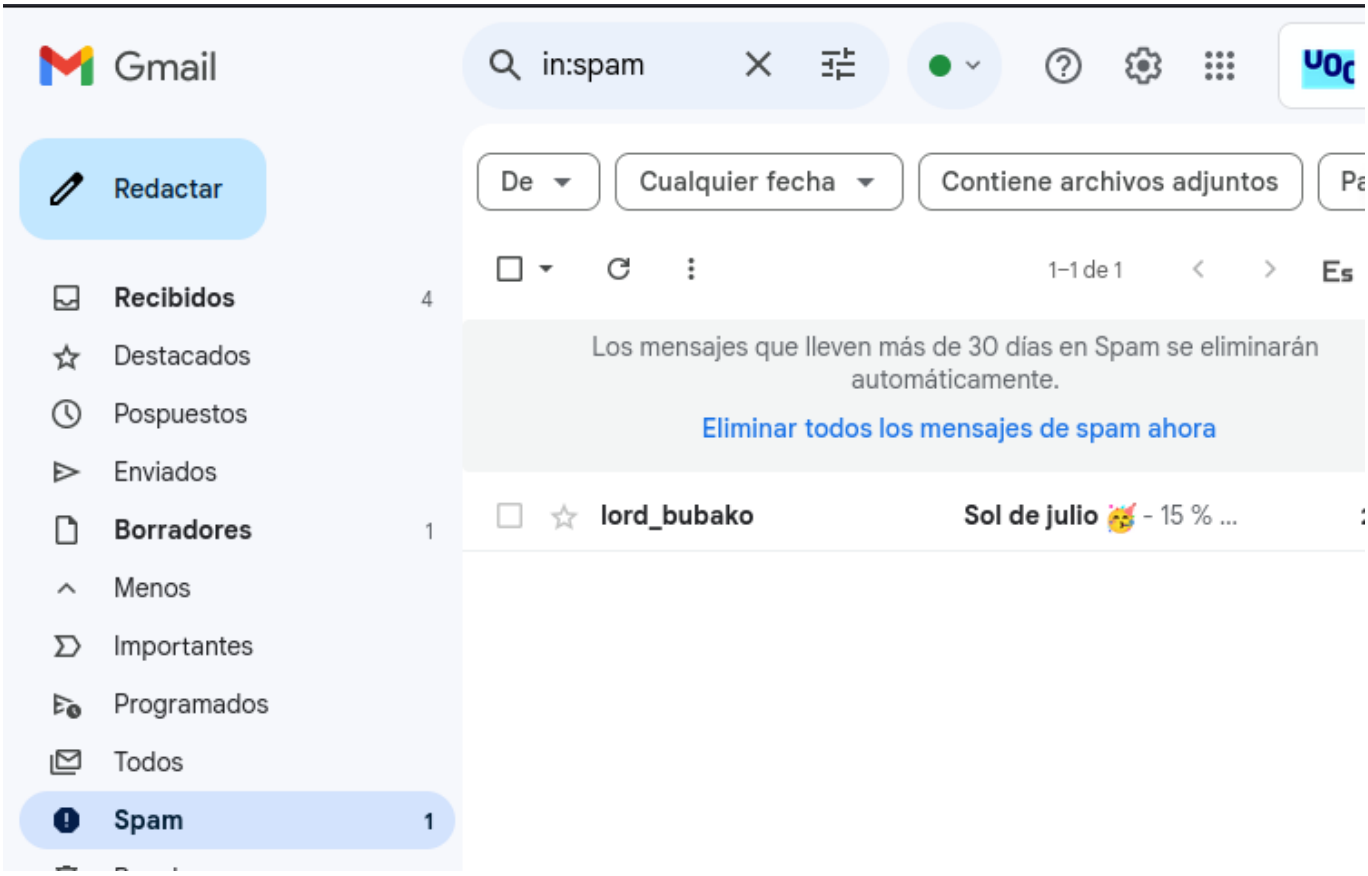
Ejecución de la campaña y recolección de datos.

Para comenzar a ejecutar la campaña, solo nos queda unificar todo en el apartado campaña. Seleccionamos los datos, y en URL, ponemos localhost para que no salga al exterior.

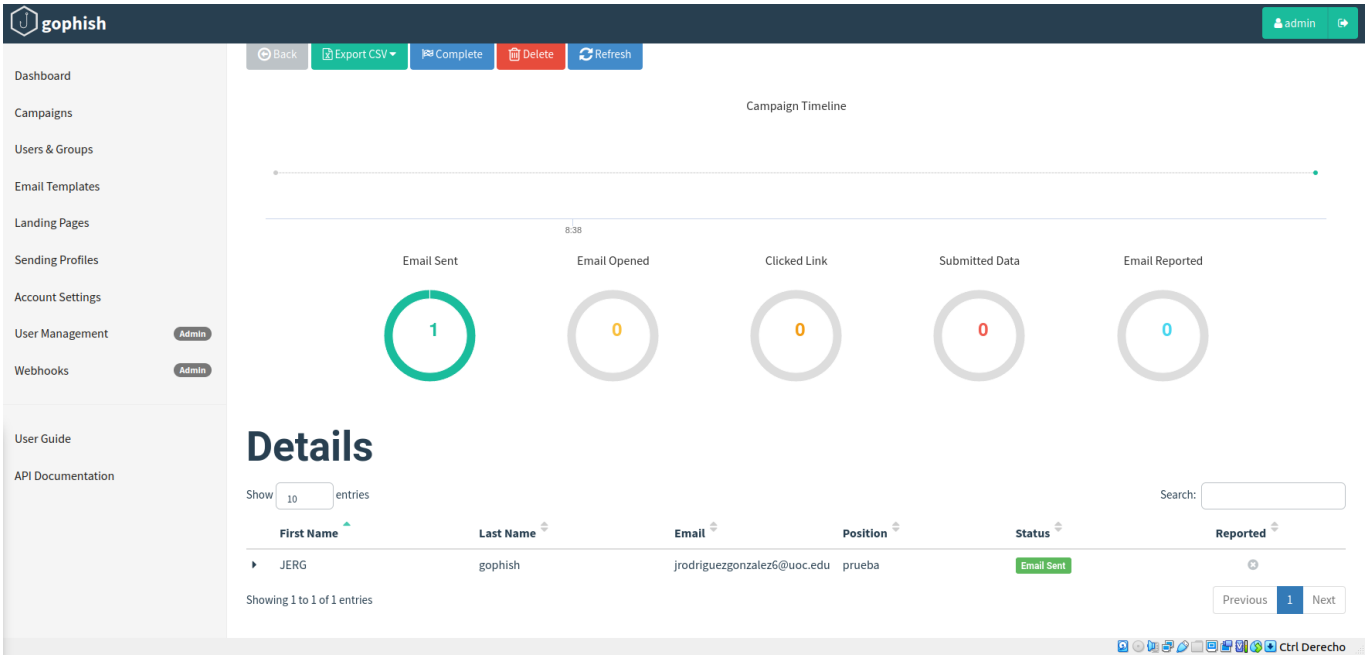


Entramos desde la maquina de Kali en nuestro correo electrónico, haciendo un ambiente simulado ya que todo lo hemos montado en la maquina de kali.

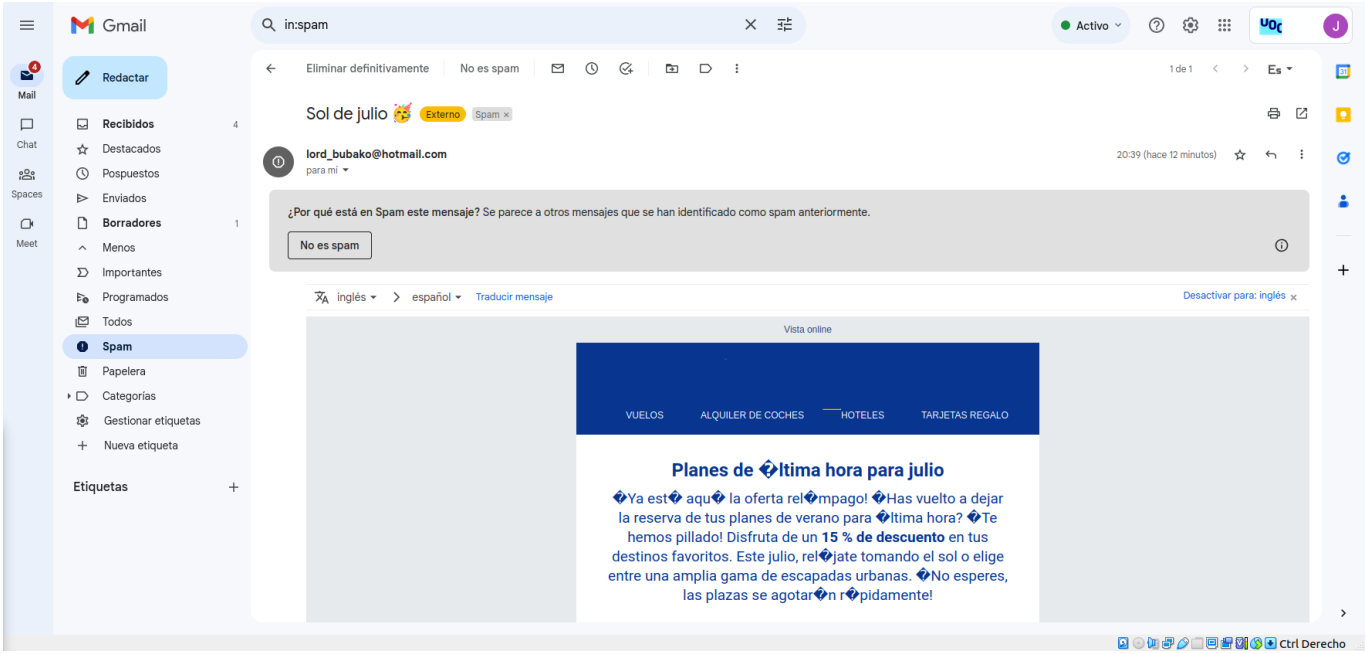
En nuestro caso hemos entrado en la parte de SPAM para ver el mail.

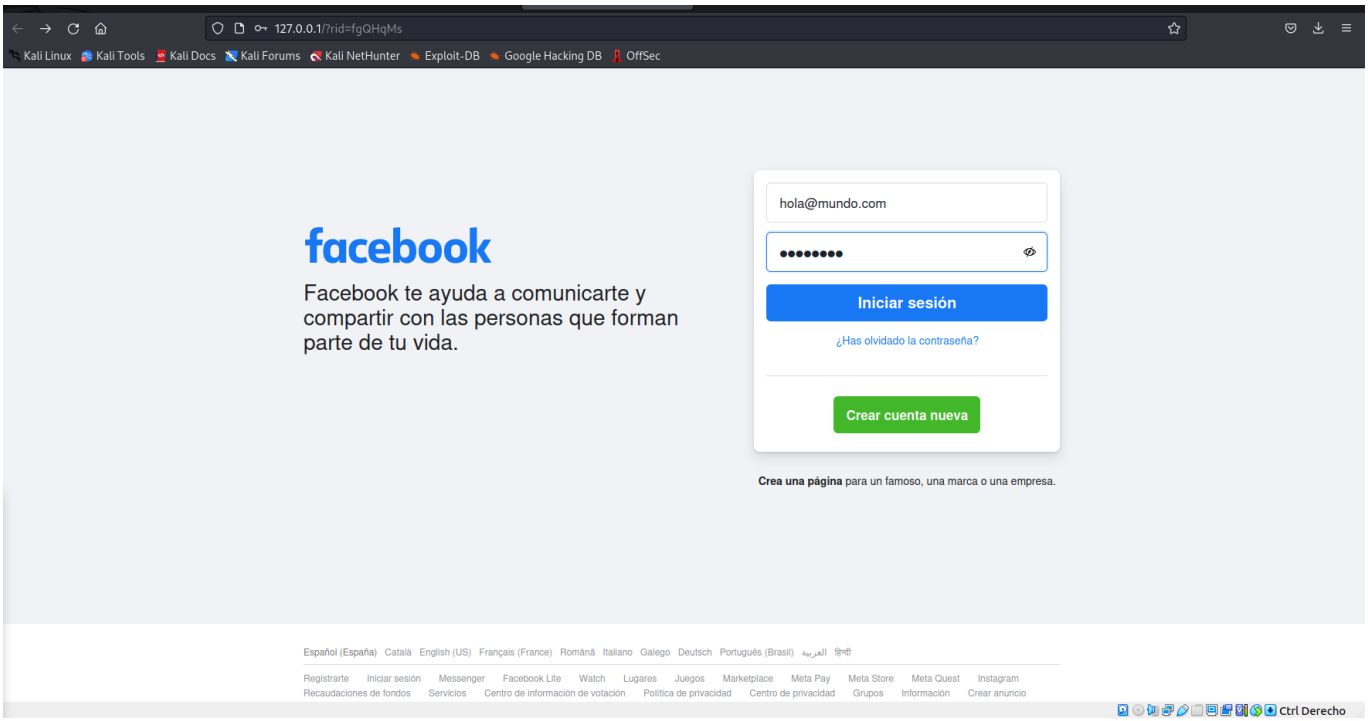


Por otro lado, podemos ver que en dashboard de la campaña, el email ha sido enviado.

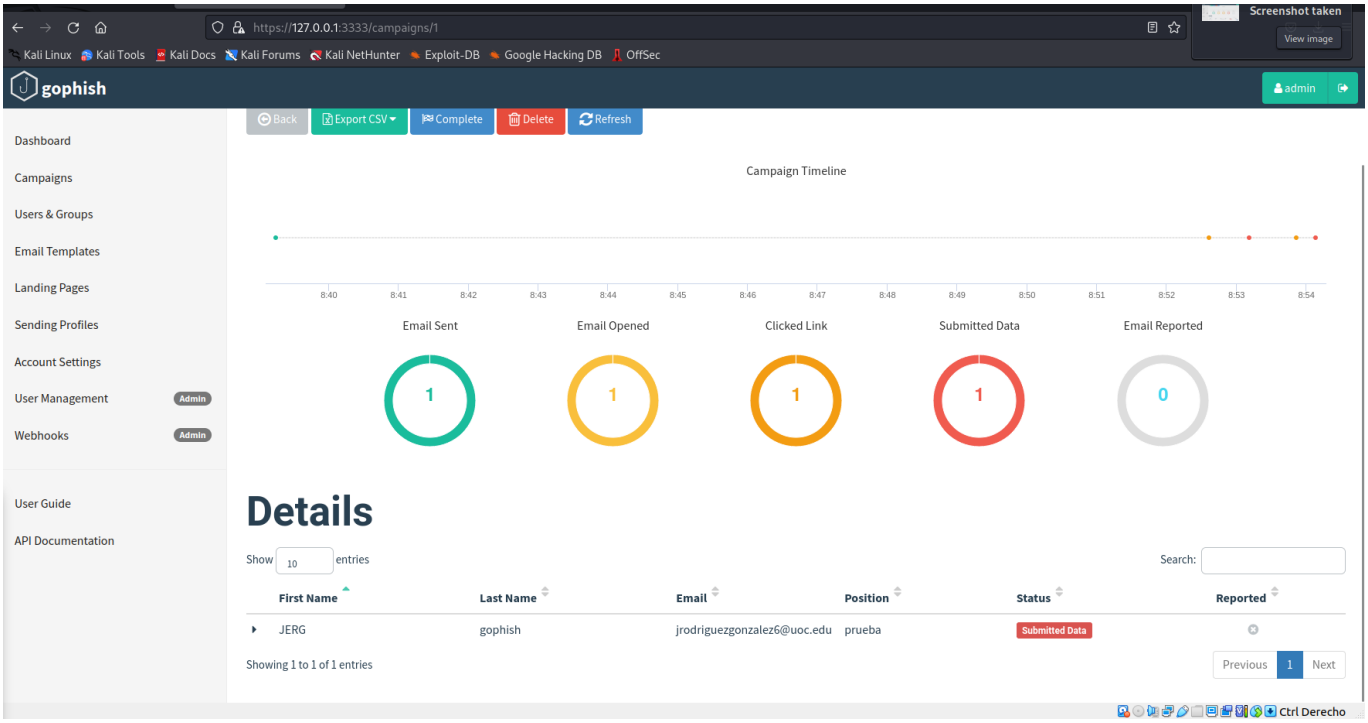


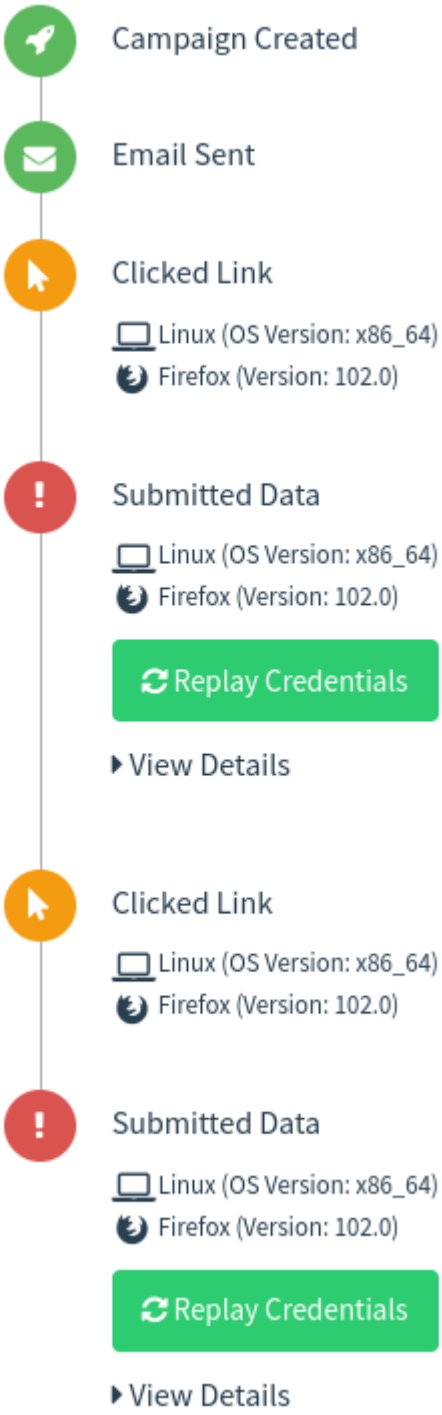
Procedemos a abrir el correo y simulamos que somos victima de phishing.


Gmail
in:spam
Activo
Redactar
Recibidos
Destacados
Pospuestos
Enviados
Borradores
Menos
Importantes
Programados
Todos
Spam
Papelera
Categorías
Gestionar etiquetas
Nueva etiqueta
Etiquetas
Eliminar definitivamente
No es spam
Sol de julio
Externo
Spam
lord_bubako@hotmail.com
para mí
20:39 (hace 12 minutos)
¿Por qué está en Spam este mensaje? Se parece a otros mensajes que se han identificado como spam anteriormente.
No es spam
inglés
español
Traducir mensaje
Desactivar para: inglés
Vista online
VUELOS
ALQUILER DE COCHES
HOTELES
TARJETAS REGALO
Planes de Última hora para julio
¡Ya está aquí la oferta relámpago! ¿Has vuelto a dejar la reserva de tus planes de verano para última hora? ¡Te hemos pillado! Disfruta de un 15 % de descuento en tus destinos favoritos. Este julio, relájate tomando el sol o elige entre una amplia gama de escapadas urbanas. ¡No esperes, las plazas se agotarán rápidamente!
Ctrl Derecho

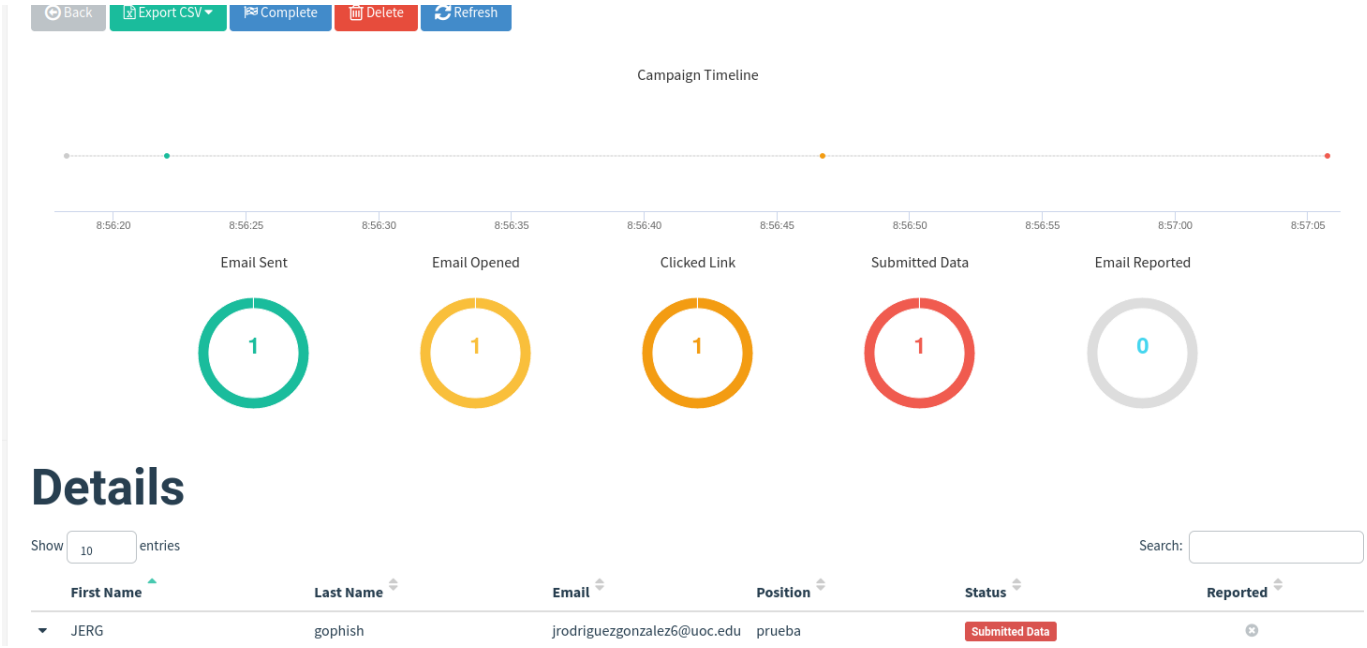

127.0.0.1/?rid=fgQHqMs
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
facebook
Facebook te ayuda a comunicarte y compartir con las personas que forman parte de tu vida.
hola@mundo.com
Iniciar sesión
¿Has olvidado la contraseña?
Crear cuenta nueva
Crea una página para un famoso, una marca o una empresa.
Español (España) Català English (US) Français (France) Română Italiano Galego Deutsch Português (Brasil)
Registrarte Iniciar sesión Messenger Facebook Lite Watch Lugares Juegos Marketplace Meta Pay Meta Store Meta Quest Instagram
Recaudaciones de fondos Servicios Centro de información de votación Política de privacidad Centro de privacidad Grupos Información Crear anuncio
Ctrl Derecho

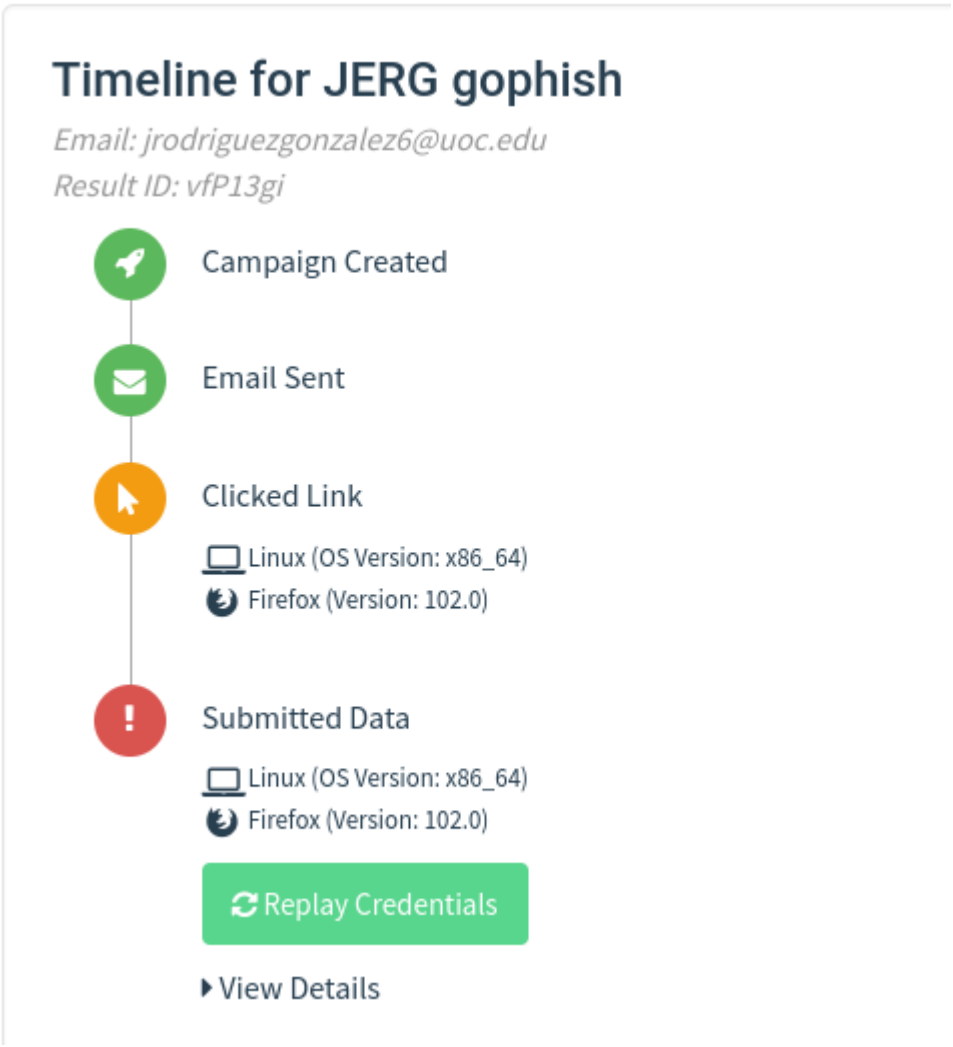
En el dashboard de gophish, podemos ver la evolución y una linea del tiempo de las acciones realizadas.



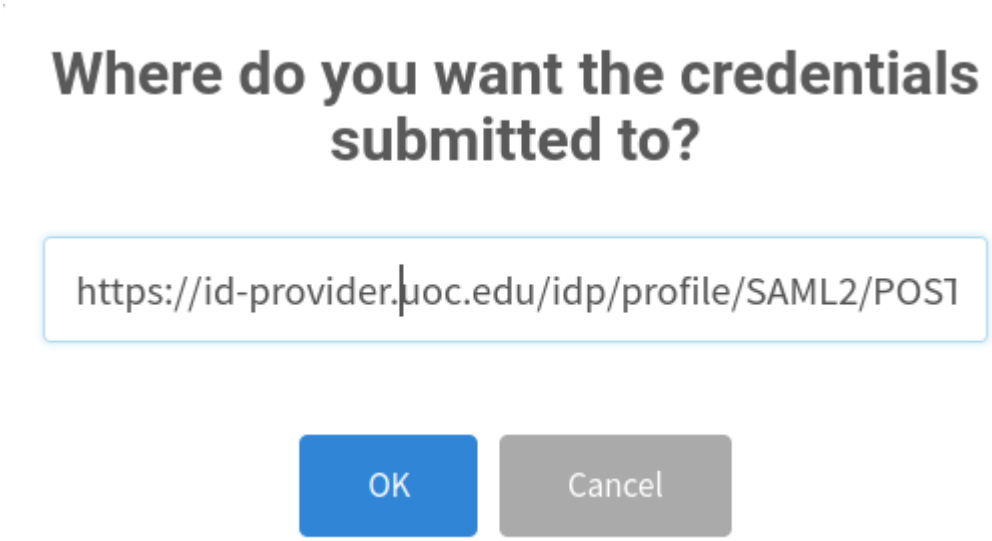


Como las credenciales han sido falsas, he decidido hacer una nueva campaña y remitir mis credenciales de acceso a al portal de la uoc, ya que he usado ahora esa cuenta en la nueva campaña para comprobar su funcionamiento.





A continuación, haremos click en replay credentials y seleccionaremos a donde quieres que re remitan las credenciales.



Hacemos click en ok

En este caso, como la sesión esta caducada, da fallo de acceso, por lo que el token esta ya inactivo.

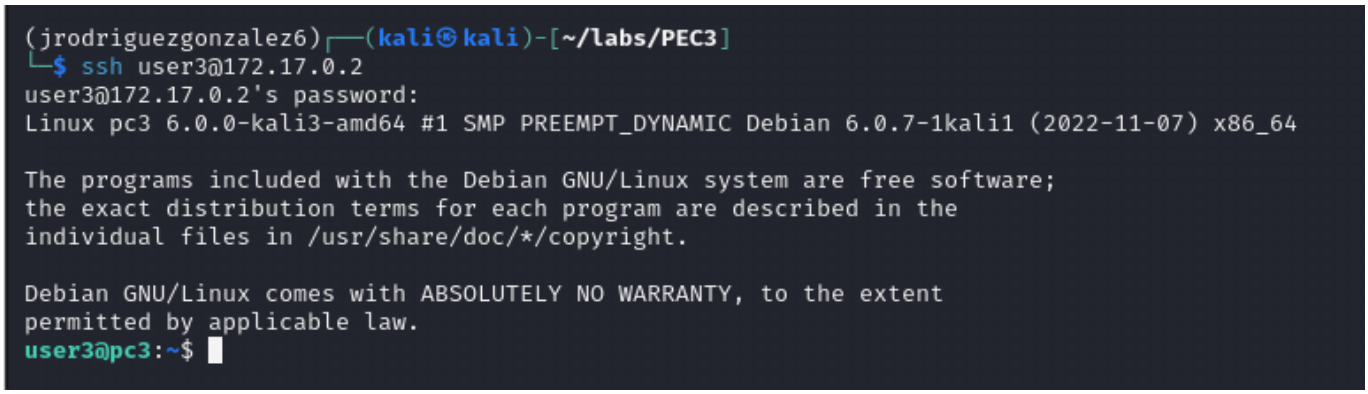
Volver al indice.

Respuesta a la pregunta 2.

Para la realización de este ejercicio, usare de fuente el siguiente video del canal de YouTube de DigiParty League

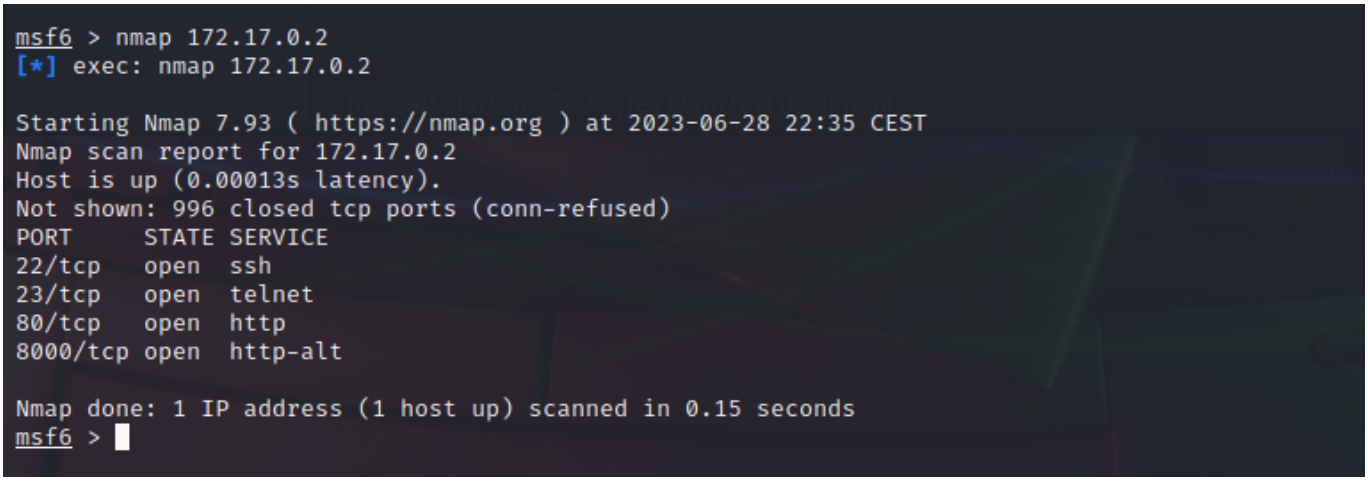
Iniciamos en la VM de kali Metasploit

Ahora, procederé a comprobar de que tengo acceso a pc3 desde la VM de kali

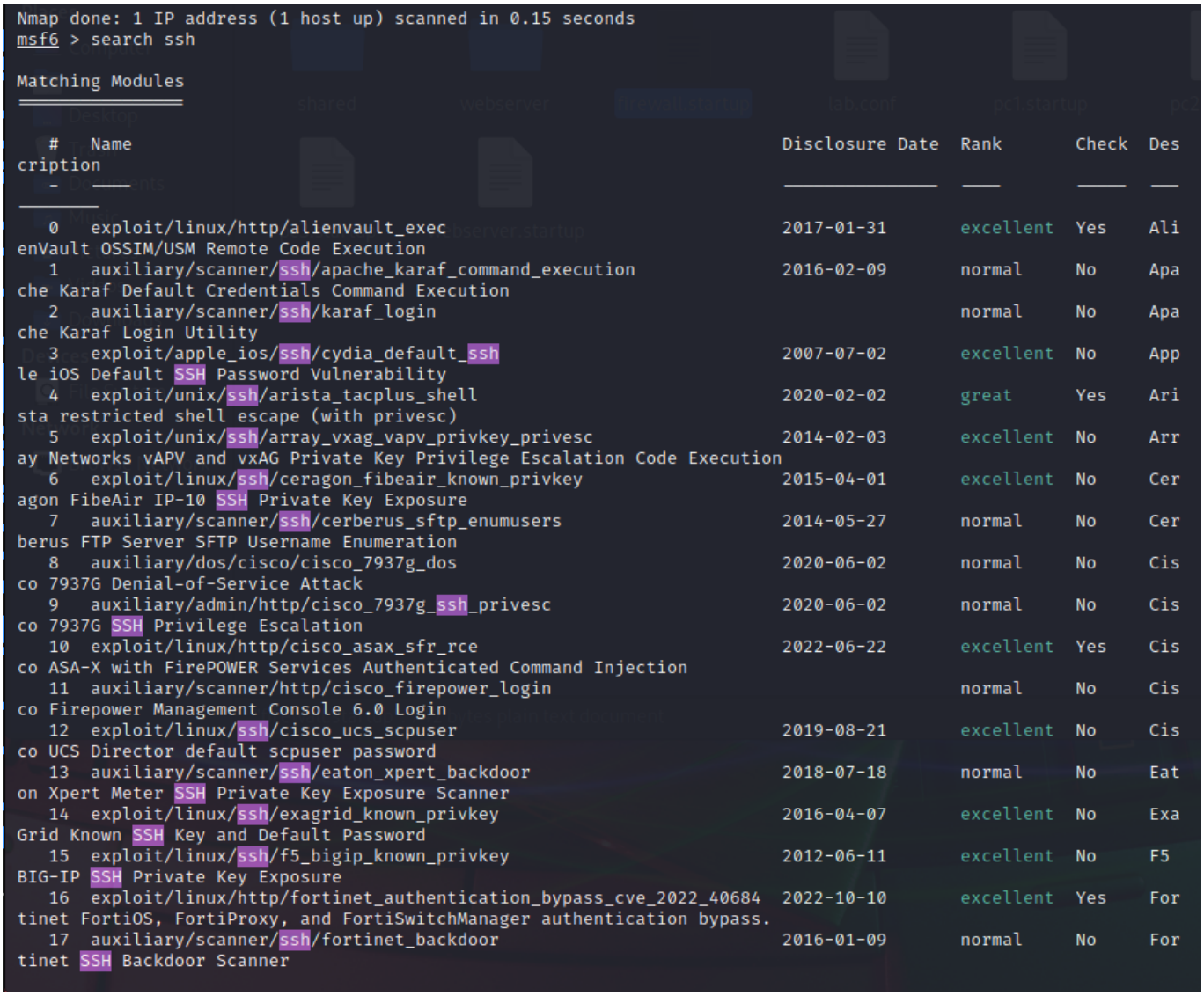


Nosotros, para acceder al ambiente de Kathará desde la VM de kali, accederemos a traves del firewall, por tanto nuestra victima sera el firewall.

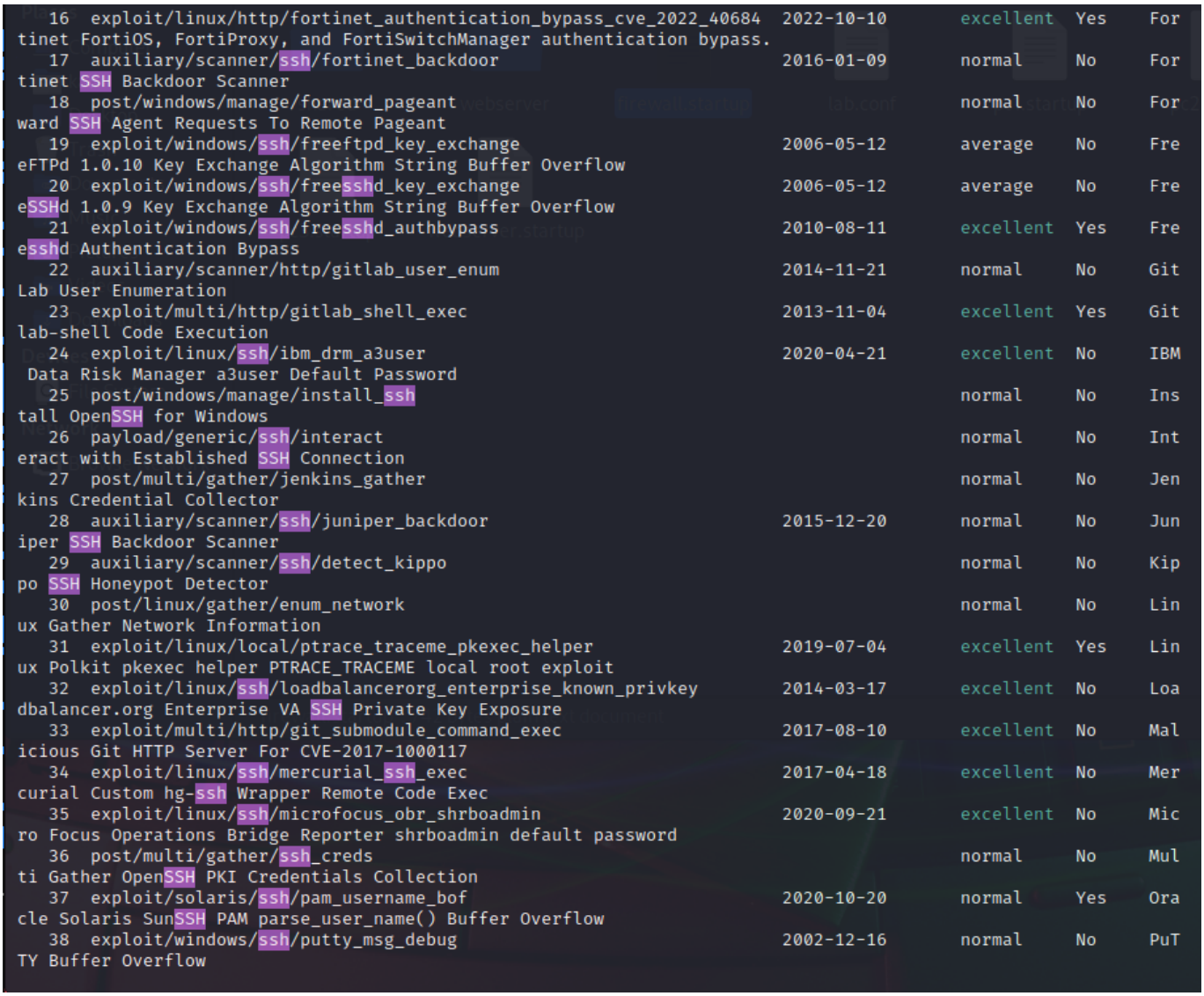
procedemos a hacer un `nmap` al firewall obteniendo la siguiente respuesta:



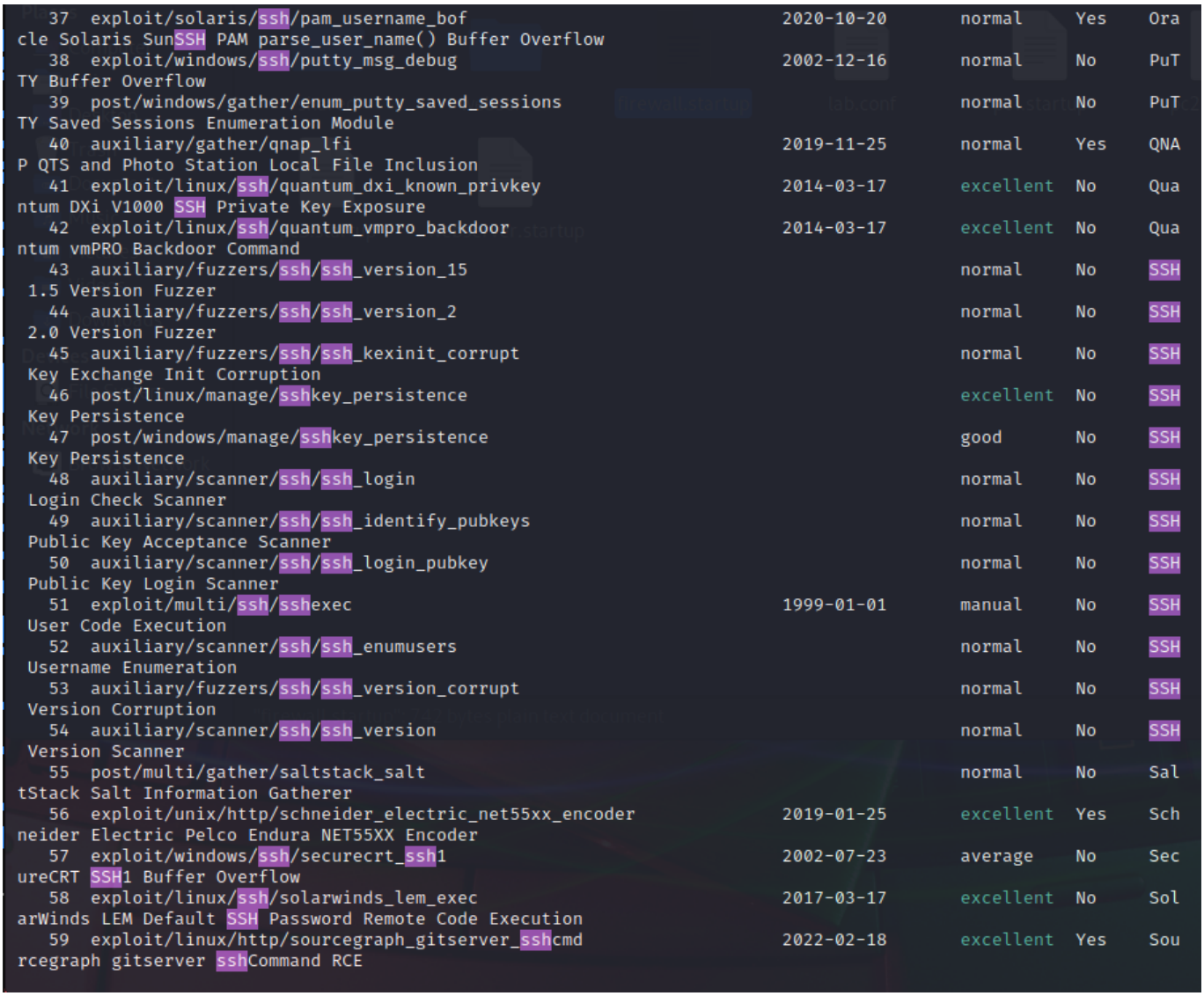
Podemos observar que tenemos el puerto 22 a la escucha. Ahora buscaremos vulnerabilidades del tipo ssh con el comando `search ssh`.



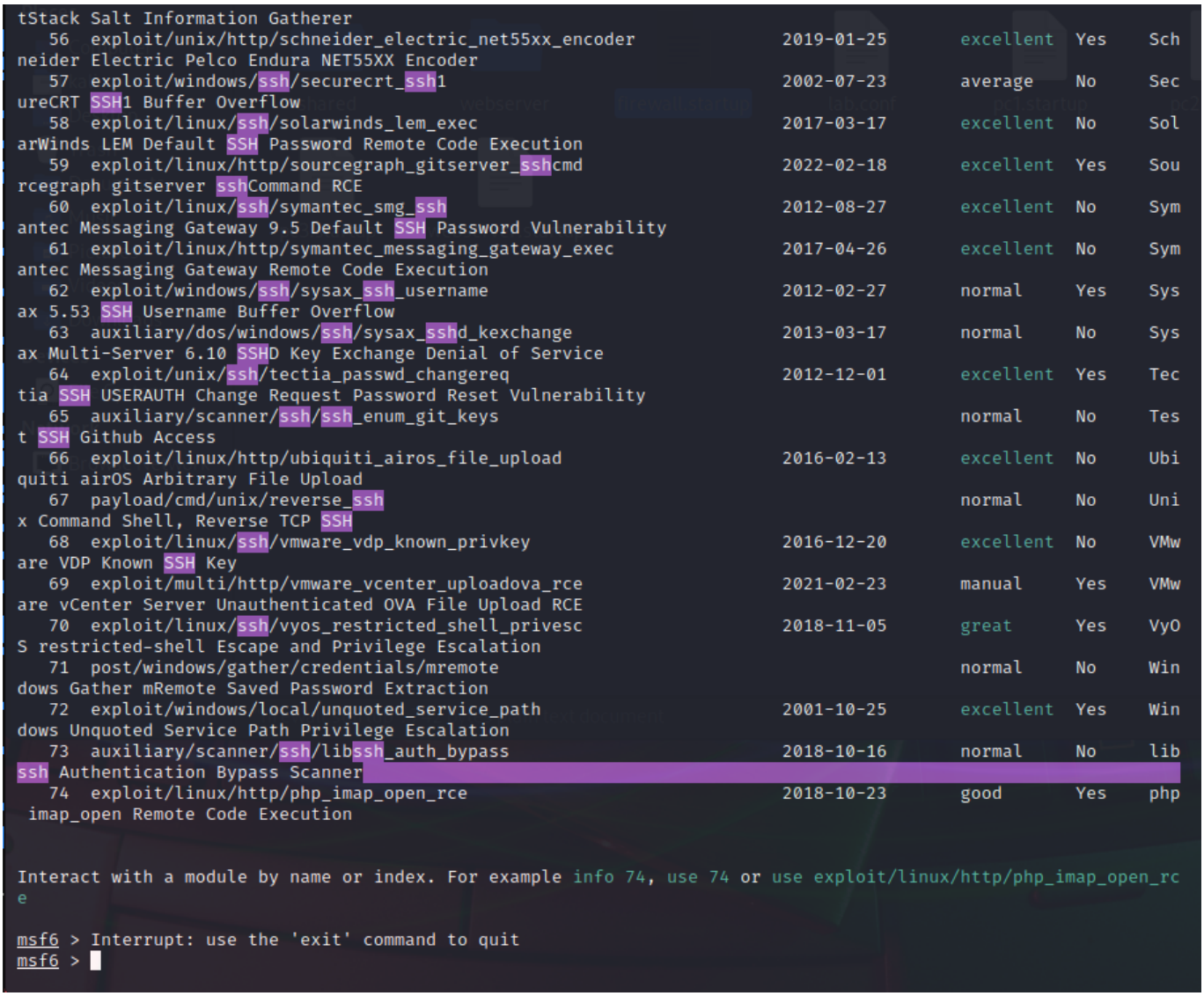
```
Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
msf6 > search ssh

Matching Modules
================

   #   Name                                                                  Disclosure Date  Rank       Check  Des
cription
   -   ----                                                                  ---------------  ----       -----  ---
--------
   0   exploit/linux/http/alienvault_exec                                    2017-01-31       excellent  Yes    Ali
enVault OSSIM/USM Remote Code Execution
   1   auxiliary/scanner/ssh/apache_karaf_command_execution                  2016-02-09       normal     No     Apa
che Karaf Default Credentials Command Execution
   2   auxiliary/scanner/ssh/karaf_login                                                      normal     No     Apa
che Karaf Login Utility
   3   exploit/apple_ios/ssh/cydia_default_ssh                               2007-07-02       excellent  No     App
le iOS Default SSH Password Vulnerability
   4   exploit/unix/ssh/arista_tacplus_shell                                 2020-02-02       great      Yes    Ari
sta restricted shell escape (with privesc)
   5   exploit/unix/ssh/array_vxag_vapv_privkey_privesc                      2014-02-03       excellent  No     Arr
ay Networks vAPV and vxAG Private Key Privilege Escalation Code Execution
   6   exploit/linux/ssh/ceragon_fibeair_known_privkey                       2015-04-01       excellent  No     Cer
agon FibeAir IP-10 SSH Private Key Exposure
   7   auxiliary/scanner/ssh/cerberus_sftp_enumusers                         2014-05-27       normal     No     Cer
berus FTP Server SFTP Username Enumeration
   8   auxiliary/dos/cisco/cisco_7937g_dos                                   2020-06-02       normal     No     Cis
co 7937G Denial-of-Service Attack
   9   auxiliary/admin/http/cisco_7937g_ssh_privesc                          2020-06-02       normal     No     Cis
co 7937G SSH Privilege Escalation
   10  exploit/linux/http/cisco_asax_sfr_rce                                 2022-06-22       excellent  Yes    Cis
co ASA-X with FirePOWER Services Authenticated Command Injection
   11  auxiliary/scanner/http/cisco_firepower_login                                           normal     No     Cis
co Firepower Management Console 6.0 Login
   12  exploit/linux/ssh/cisco_ucs_scpuser                                   2019-08-21       excellent  No     Cis
co UCS Director default scpuser password
   13  auxiliary/scanner/ssh/eaton_xpert_backdoor                            2018-07-18       normal     No     Eat
on Xpert Meter SSH Private Key Exposure Scanner
   14  exploit/linux/ssh/exagrid_known_privkey                               2016-04-07       excellent  No     Exa
Grid Known SSH Key and Default Password
   15  exploit/linux/ssh/f5_bigip_known_privkey                              2012-06-11       excellent  No     F5
BIG-IP SSH Private Key Exposure
   16  exploit/linux/http/fortinet_authentication_bypass_cve_2022_40684      2022-10-10       excellent  Yes    For
tinet FortiOS, FortiProxy, and FortiSwitchManager authentication bypass.
   17  auxiliary/scanner/ssh/fortinet_backdoor                               2016-01-09       normal     No     For
tinet SSH Backdoor Scanner
```



```
   16  exploit/linux/http/fortinet_authentication_bypass_cve_2022_40684  2022-10-10       excellent  Yes    For
tinet FortiOS, FortiProxy, and FortiSwitchManager authentication bypass.
   17  auxiliary/scanner/ssh/fortinet_backdoor                           2016-01-09       normal     No     For
tinet SSH Backdoor Scanner
   18  post/windows/manage/forward_pageant                                               normal     No     For
ward SSH Agent Requests To Remote Pageant
   19  exploit/windows/ssh/freeftpd_key_exchange                         2006-05-12       average    No     Fre
eFTPd 1.0.10 Key Exchange Algorithm String Buffer Overflow
   20  exploit/windows/ssh/freesshd_key_exchange                         2006-05-12       average    No     Fre
eSSHd 1.0.9 Key Exchange Algorithm String Buffer Overflow
   21  exploit/windows/ssh/freesshd_authbypass                           2010-08-11       excellent  Yes    Fre
esshd Authentication Bypass
   22  auxiliary/scanner/http/gitlab_user_enum                           2014-11-21       normal     No     Git
Lab User Enumeration
   23  exploit/multi/http/gitlab_shell_exec                              2013-11-04       excellent  Yes    Git
lab-shell Code Execution
   24  exploit/linux/ssh/ibm_drm_a3user                                  2020-04-21       excellent  No     IBM
 Data Risk Manager a3user Default Password
   25  post/windows/manage/install_ssh                                                    normal     No     Ins
tall OpenSSH for Windows
   26  payload/generic/ssh/interact                                                       normal     No     Int
eract with Established SSH Connection
   27  post/multi/gather/jenkins_gather                                                   normal     No     Jen
kins Credential Collector
   28  auxiliary/scanner/ssh/juniper_backdoor                            2015-12-20       normal     No     Jun
iper SSH Backdoor Scanner
   29  auxiliary/scanner/ssh/detect_kippo                                                 normal     No     Kip
po SSH Honeypot Detector
   30  post/linux/gather/enum_network                                                     normal     No     Lin
ux Gather Network Information
   31  exploit/linux/local/ptrace_traceme_pkexec_helper                  2019-07-04       excellent  Yes    Lin
ux Polkit pkexec helper PTRACE_TRACEME local root exploit
   32  exploit/linux/ssh/loadbalancerorg_enterprise_known_privkey        2014-03-17       excellent  No     Loa
dbalancer.org Enterprise VA SSH Private Key Exposure
   33  exploit/multi/http/git_submodule_command_exec                     2017-08-10       excellent  No     Mal
icious Git HTTP Server For CVE-2017-1000117
   34  exploit/linux/ssh/mercurial_ssh_exec                              2017-04-18       excellent  No     Mer
curial Custom hg-ssh Wrapper Remote Code Exec
   35  exploit/linux/ssh/microfocus_obr_shrboadmin                       2020-09-21       excellent  No     Mic
ro Focus Operations Bridge Reporter shrboadmin default password
   36  post/multi/gather/ssh_creds                                                        normal     No     Mul
ti Gather OpenSSH PKI Credentials Collection
   37  exploit/solaris/ssh/pam_username_bof                              2020-10-20       normal     Yes    Ora
cle Solaris SunSSH PAM parse_user_name() Buffer Overflow
   38  exploit/windows/ssh/putty_msg_debug                               2002-12-16       normal     No     PuT
TY Buffer Overflow
```

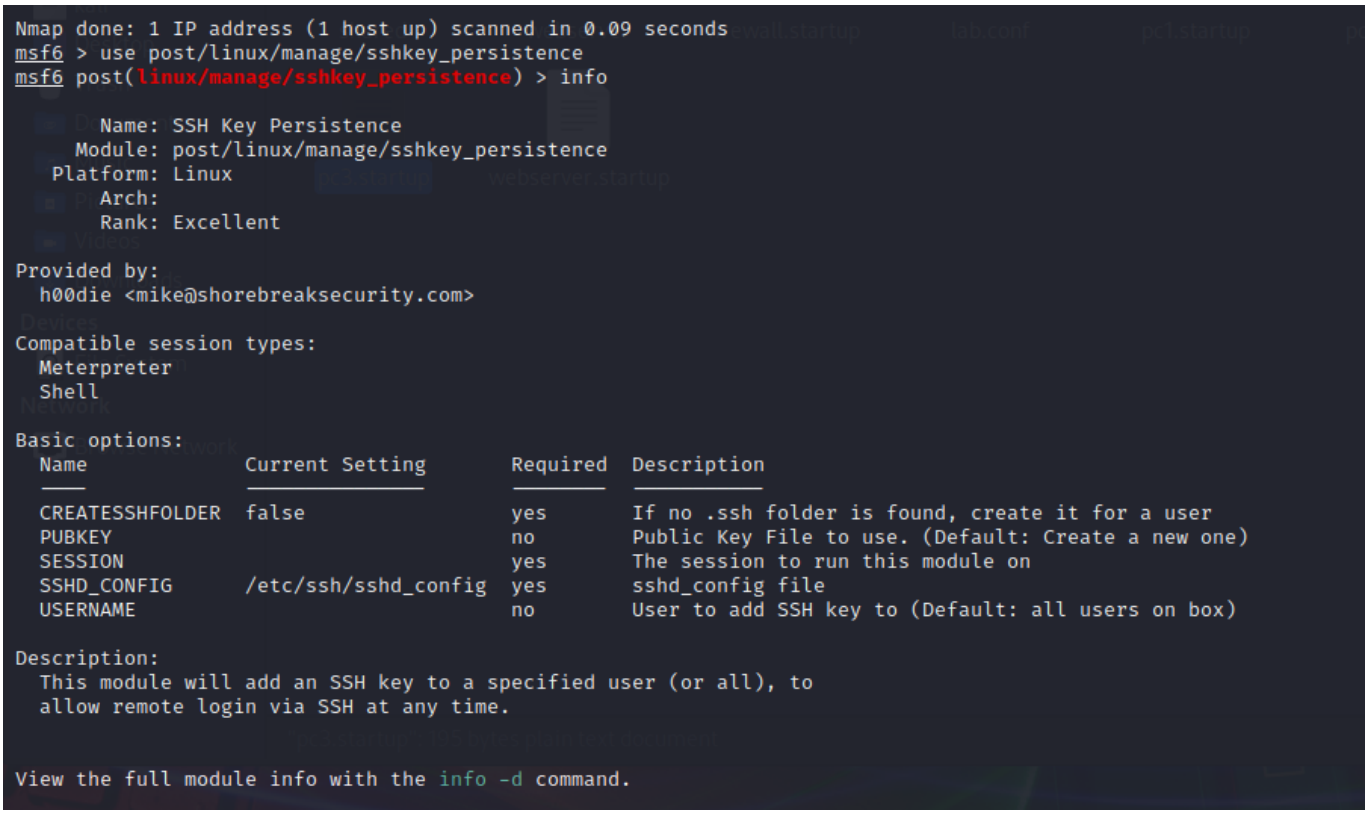


```
   37  exploit/solaris/ssh/pam_username_bof                    2020-10-20       normal     Yes    Ora
cle Solaris SunSSH PAM parse_user_name() Buffer Overflow
   38  exploit/windows/ssh/putty_msg_debug                     2002-12-16       normal     No     PuT
TY Buffer Overflow
   39  post/windows/gather/enum_putty_saved_sessions                            normal     No     PuT
TY Saved Sessions Enumeration Module
   40  auxiliary/gather/qnap_lfi                               2019-11-25       normal     Yes    QNA
P QTS and Photo Station Local File Inclusion
   41  exploit/linux/ssh/quantum_dxi_known_privkey             2014-03-17       excellent  No     Qua
ntum DXi V1000 SSH Private Key Exposure
   42  exploit/linux/ssh/quantum_vmpro_backdoor                2014-03-17       excellent  No     Qua
ntum vmPRO Backdoor Command
   43  auxiliary/fuzzers/ssh/ssh_version_15                                     normal     No     SSH
 1.5 Version Fuzzer
   44  auxiliary/fuzzers/ssh/ssh_version_2                                      normal     No     SSH
 2.0 Version Fuzzer
   45  auxiliary/fuzzers/ssh/ssh_kexinit_corrupt                                normal     No     SSH
 Key Exchange Init Corruption
   46  post/linux/manage/sshkey_persistence                                     excellent  No     SSH
 Key Persistence
   47  post/windows/manage/sshkey_persistence                                   good       No     SSH
 Key Persistence
   48  auxiliary/scanner/ssh/ssh_login                                          normal     No     SSH
 Login Check Scanner
   49  auxiliary/scanner/ssh/ssh_identify_pubkeys                               normal     No     SSH
 Public Key Acceptance Scanner
   50  auxiliary/scanner/ssh/ssh_login_pubkey                                   normal     No     SSH
 Public Key Login Scanner
   51  exploit/multi/ssh/sshexec                               1999-01-01       manual     No     SSH
 User Code Execution
   52  auxiliary/scanner/ssh/ssh_enumusers                                      normal     No     SSH
 Username Enumeration
   53  auxiliary/fuzzers/ssh/ssh_version_corrupt                                normal     No     SSH
 Version Corruption
   54  auxiliary/scanner/ssh/ssh_version                                        normal     No     SSH
 Version Scanner
   55  post/multi/gather/saltstack_salt                                         normal     No     Sal
tStack Salt Information Gatherer
   56  exploit/unix/http/schneider_electric_net55xx_encoder    2019-01-25       excellent  Yes    Sch
neider Electric Pelco Endura NET55XX Encoder
   57  exploit/windows/ssh/securecrt_ssh1                      2002-07-23       average    No     Sec
ureCRT SSH1 Buffer Overflow
   58  exploit/linux/ssh/solarwinds_lem_exec                   2017-03-17       excellent  No     Sol
arWinds LEM Default SSH Password Remote Code Execution
   59  exploit/linux/http/sourcegraph_gitserver_sshcmd         2022-02-18       excellent  Yes    Sou
rcegraph gitserver sshCommand RCE
```



Dentro de este tipo de payload disponibles, los que nos interesan son los de key persistence.

Usaremos el relativo a windows, ya que kathará funciona bajo Kali. de modo que usaremos en este caso ejecutando el comando `use post/linux/manage/sshkey_persistence`. Posteriormente procederemos a ver lo que necesita el payload para funcionar con el comando `info`.

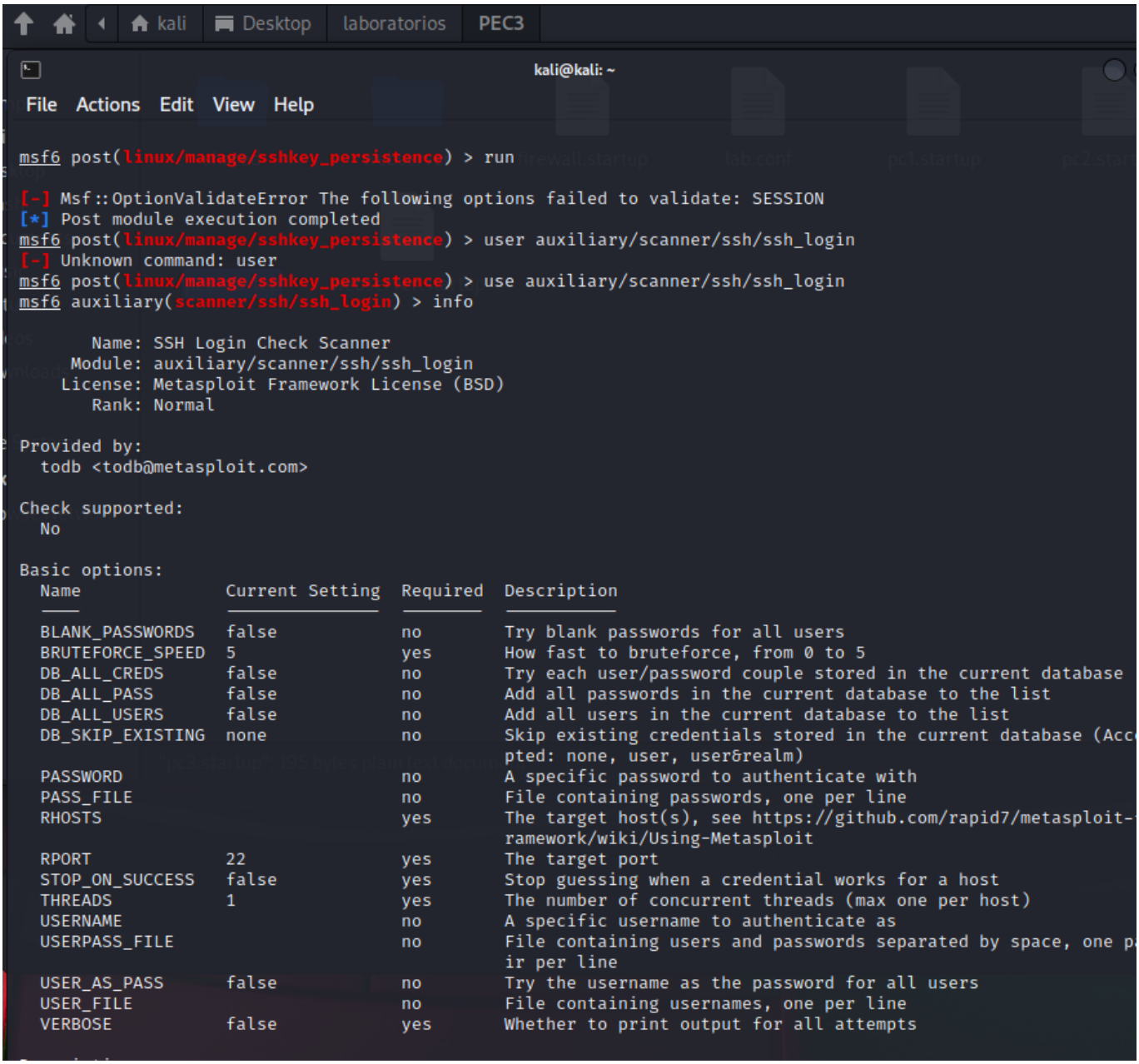


En nuestro caso, setearemos la sesión y el user. Tal y como se indica en la siguiente imagen y volvemos a comprobar.

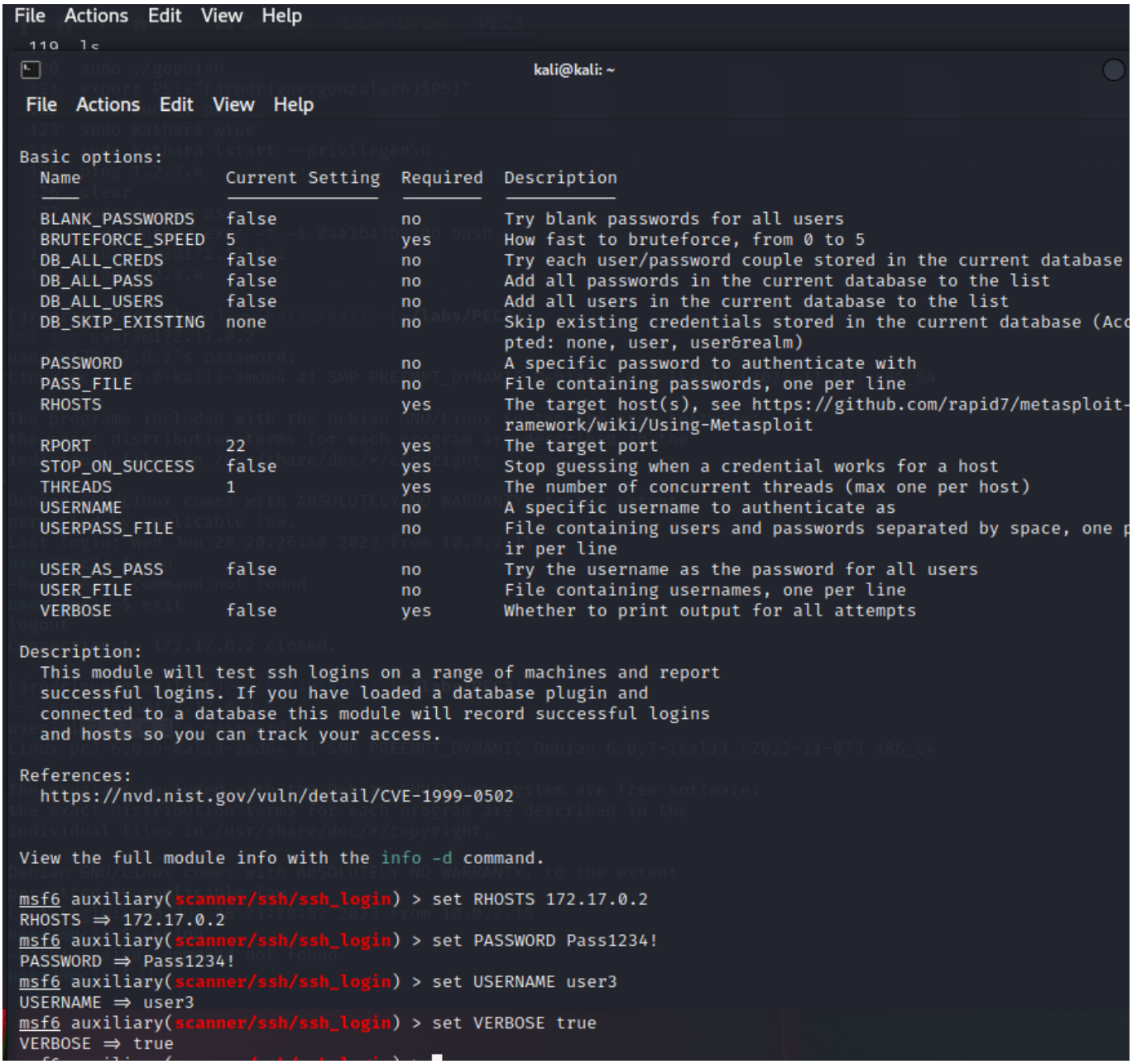
En nuestro caso, nos ha dado un fallo

```
Msf::OptionValidateError The following options failed to validate: SESSION
```

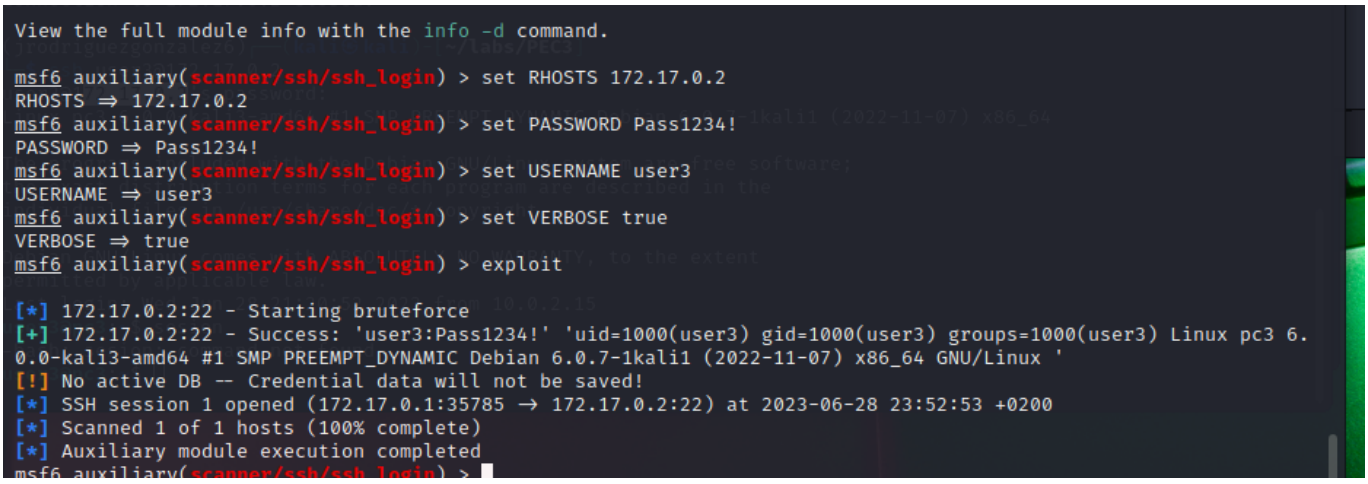
Esto se debe a que necesitamos obtener una sesión activa en la máquina objetivo. Esta sesión podría obtenerse, por ejemplo, explotando alguna vulnerabilidad en la máquina objetivo utilizando un módulo de exploit de Metasploit. En este caso usaremos el exploit de `auxiliary/scanner/ssh/ssh_login`tal y como indica el video citado como fuente.



Procedemos a configurar el auxiliary con los datos necesarios que ya disponemos y seteamos VERBOSE a true.



procedemos a ejecutar con el comando `exploit` o `run`

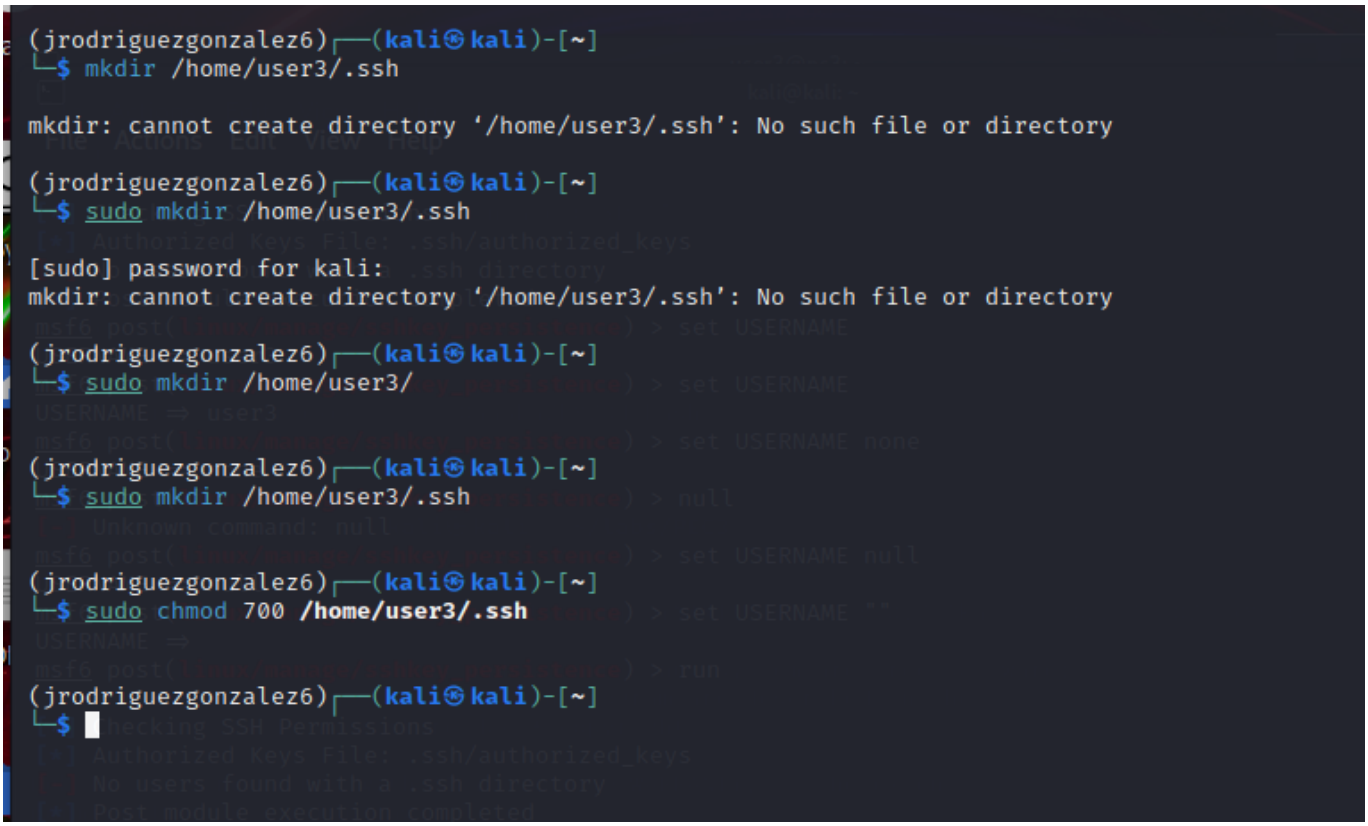


En nuestro caso nos ha dado el siguiente error:

```
[-] No users found with a .ssh directory
```

Esto se debe a que en sistemas Unix y Linux, el directorio .ssh en el directorio home de un usuario es donde se almacenan las claves SSH del usuario. Si este directorio no existe, es posible que el usuario user3 nunca haya utilizado SSH.

Procedemos a realizar los siguientes comandos desde terminal.



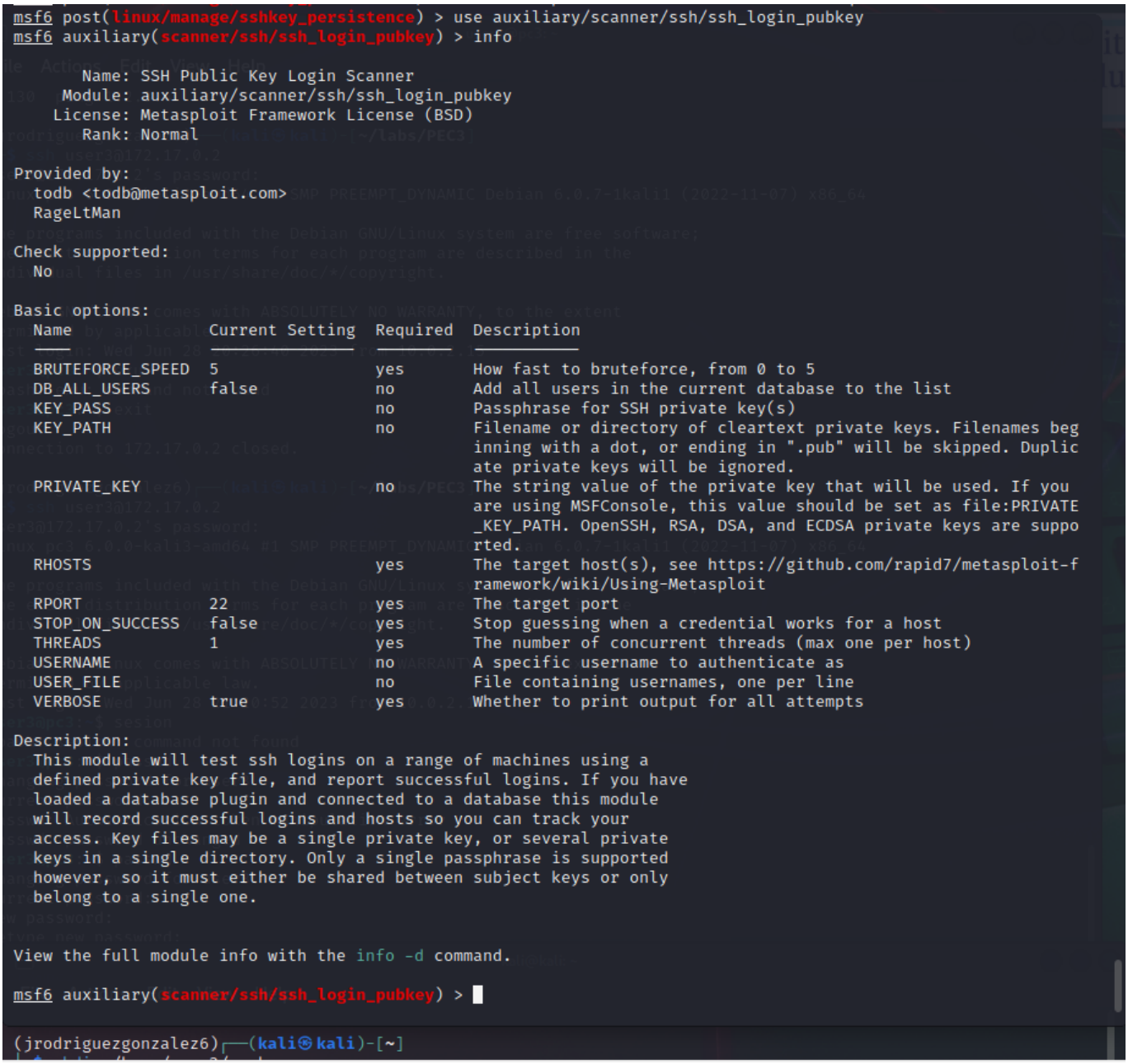
Este cambio ha seguido dando problemas por lo que se ha procedido a buscar por internet, encontrando la web de infosecmatter donde nos recomienda ejecutar el comando `set createsshfolder true`, consiguiéndose ejecutar el exploit correctamente.

```
Basic options:
  Name             Current Setting       Required  Description
  ----             ---------------       --------  -----------
  CREATESSHFOLDER  false                 yes       If no .ssh folder is found, create it for a user
  PUBKEY                                 no        Public Key File to use. (Default: Create a new one)
  SESSION          1                     yes       The session to run this module on
  SSHD_CONFIG      /etc/ssh/sshd_config  yes       sshd_config file
  USERNAME         user3                 no        User to add SSH key to (Default: all users on box)

Description:
  This module will add an SSH key to a specified user (or all), to
  allow remote login via SSH at any time.


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

msf6 post(linux/manage/sshkey_persistence) > set createsshfolder true
createsshfolder ⇒ true
msf6 post(linux/manage/sshkey_persistence) > run

[*] Checking SSH Permissions
[*] Authorized Keys File: .ssh/authorized_keys
[*] Creating /home/user3/.ssh folder
[+] Storing new private key as /home/kali/.msf4/loot/20230629000345_default_172.17.0.2_id_rsa_425819.txt
[*] Adding key to /home/user3/.ssh/authorized_keys
[+] Key Added
[*] Post module execution completed
```

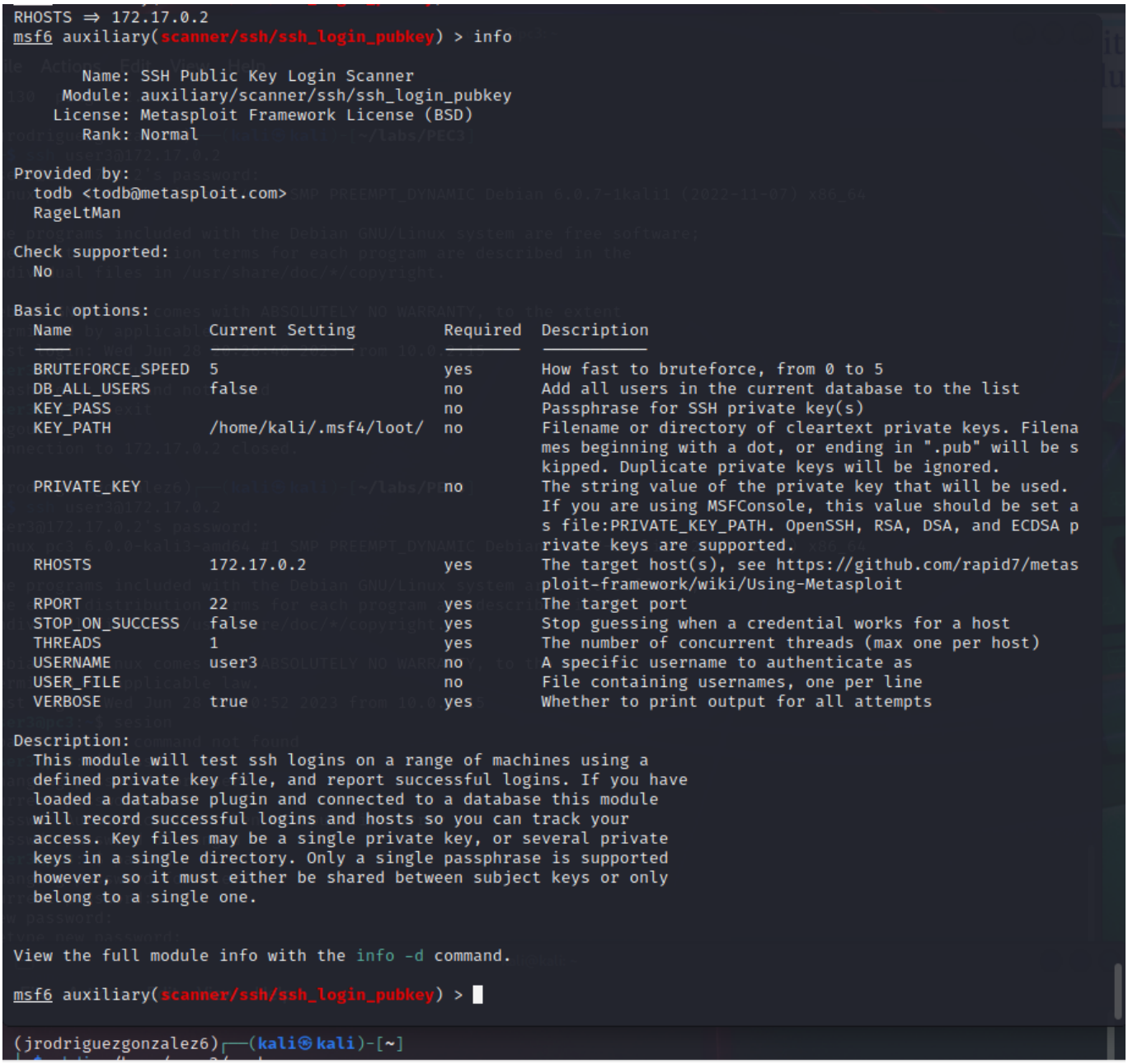
Ahora con la sesión anteriormente iniciada, procedemos a cambiar la pass de user3.

```
passwd: password unchanged
user3@pc3:~$ passwd
Changing password for user3.
Current password:
New password:
Retype new password:
passwd: password updated successfully
user3@pc3:~$
```

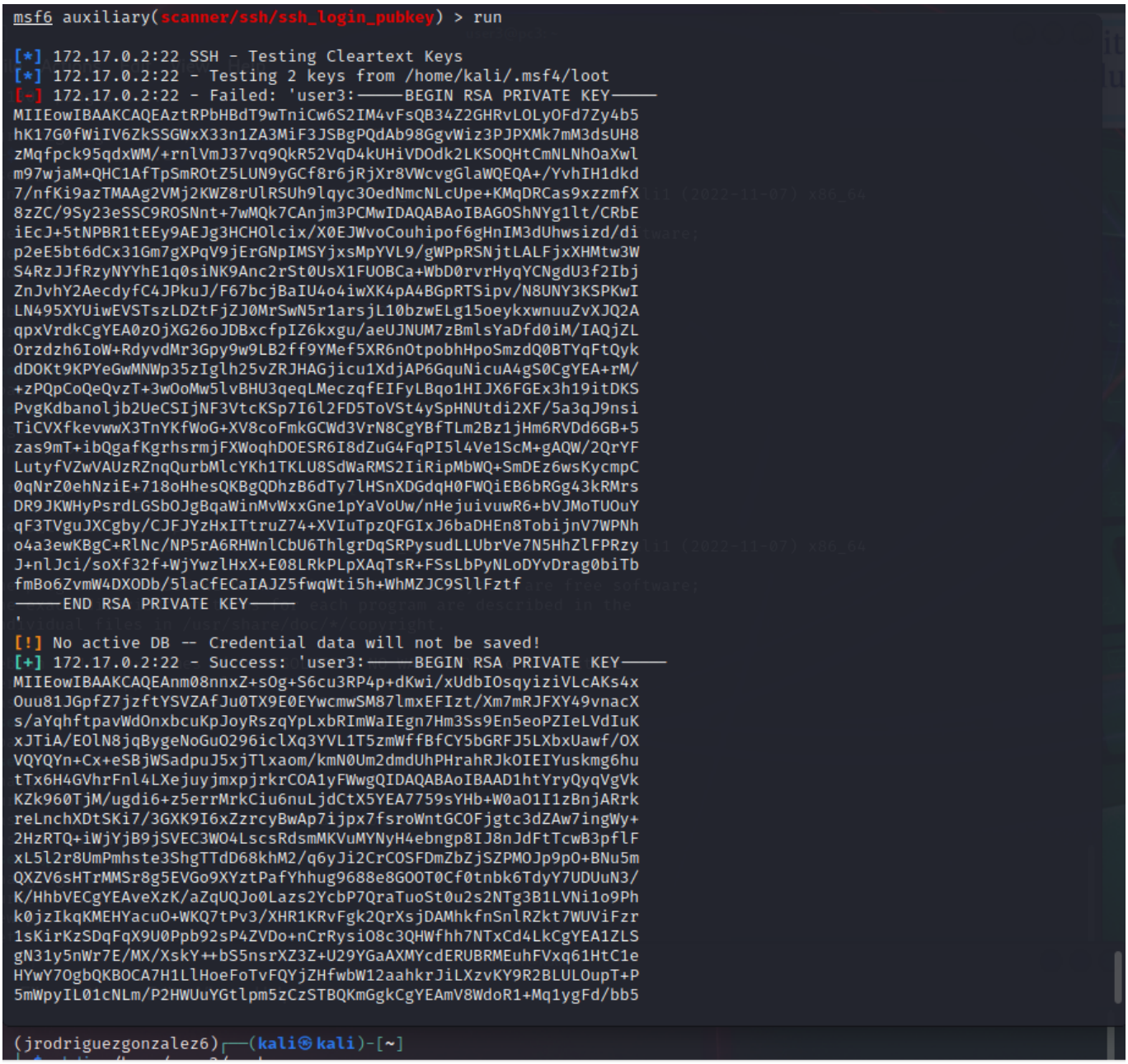
Procedemos a volver a ejecutar comandos en metasploit, procedemos a loguearnos con la pubkey anterior. Para ello ejecutaremos el comando `use auxiliary/scanner/ssh/ssh_login_pubkey`

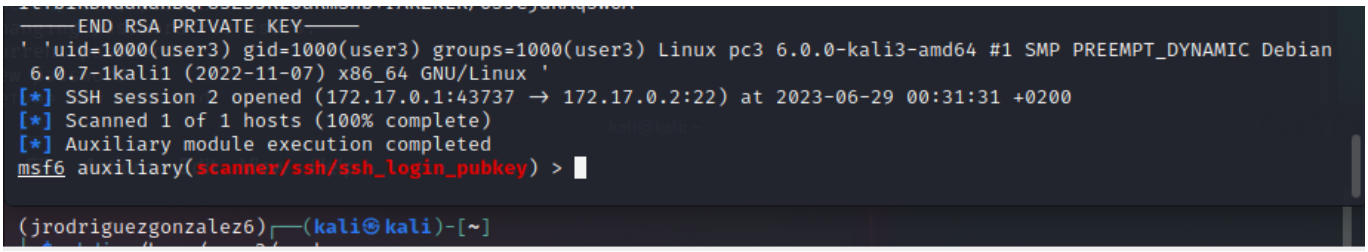


Como podemos observar en el info, necesitamos setear RHOSTS, pero también setearemos KEY_PATH y USERNAME.



Por ultimo procedemos a ejecutar el exploit, resultando satisfactorio.





Volver al indice.

Respuesta a la pregunta 3.

El ataque de Directory Transversal, también conocido como Path Traversal, se produce cuando un atacante manipula variables que hacen referencia a rutas de archivos con la intención de acceder a archivos y directorios que están fuera del directorio de trabajo previsto.

En nuestro caso el código del main.py que esta alojado en el `webserver` es el siguiente:

```
import os
from http.server import BaseHTTPRequestHandler, HTTPServer
from urllib.parse import urlparse, parse_qs

class FileHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        parsed_path = urlparse(self.path)
        query_params = parse_qs(parsed_path.query)
        file_path = query_params.get('file', [''])[0]

        if not file_path:
            self.send_response(400)
            self.end_headers()
            self.wfile.write(b'Missing file parameter')
            return

        if not os.path.exists(file_path):
            self.send_response(404)
            self.end_headers()
            self.wfile.write(b'File not found')
            return

        with open(file_path, 'rb') as f:
            file_content = f.read()

        self.send_response(200)
        self.send_header('Content-type', 'text/plain')
        self.end_headers()
        self.wfile.write(file_content)

if __name__ == '__main__':
    server_address = ('', 8000)
    httpd = HTTPServer(server_address, FileHandler)
    httpd.serve_forever()
```

Para solucionarlo, se debe insertar las siguientes lineas entre `if not file_path:` y `if not os.path.exists(file_path):`

```
absolute_path = os.path.abspath(file_path)

server_dir = os.path.abspath(os.path.dirname(__file__))
```

```
if not absolute_path.startswith(server_dir):
    self.send_response(403)
    self.end_headers()
    self.wfile.write(b'Access denied')
    return
```

Con `absolute_path = os.path.abspath(file_path)`, se calcula la ruta absoluta del archivo en cuestión, mientas con `server_dir = os.path.abspath(os.path.dirname(__file__))`y el siguiente condicional lo que hacemos es verificar si la ruta absoluta comienza con el directorio del servidor.

El archivo main.py debe de quedar de la siguiente manera:

```
import os
from http.server import BaseHTTPRequestHandler, HTTPServer
from urllib.parse import urlparse, parse_qs

class FileHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        parsed_path = urlparse(self.path)
        query_params = parse_qs(parsed_path.query)
        file_path = query_params.get('file', [''])[0]

        if not file_path:
            self.send_response(400)
            self.end_headers()
            self.wfile.write(b'Missing file parameter')
            return

        # Calcula la ruta absoluta del archivo
        absolute_path = os.path.abspath(file_path)

        # Verifica si la ruta absoluta comienza con el directorio del
servidor
        server_dir = os.path.abspath(os.path.dirname(__file__))
        if not absolute_path.startswith(server_dir):
            self.send_response(403)
            self.end_headers()
            self.wfile.write(b'Access denied')
            return

        if not os.path.exists(file_path):
            self.send_response(404)
            self.end_headers()
            self.wfile.write(b'File not found')
            return

        with open(file_path, 'rb') as f:
            file_content = f.read()

        self.send_response(200)
```

```
            self.send_header('Content-type', 'text/plain')
            self.end_headers()
            self.wfile.write(file_content)

    if __name__ == '__main__':
        server_address = ('', 8000)
        httpd = HTTPServer(server_address, FileHandler)
        httpd.serve_forever()
```

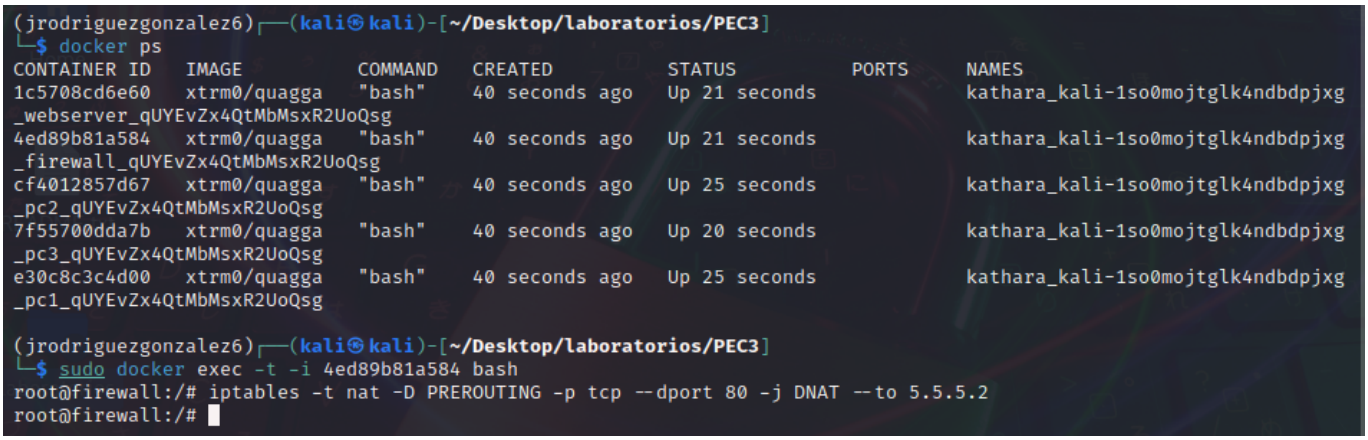
Volver al indice.

Respuesta a la pregunta 4.

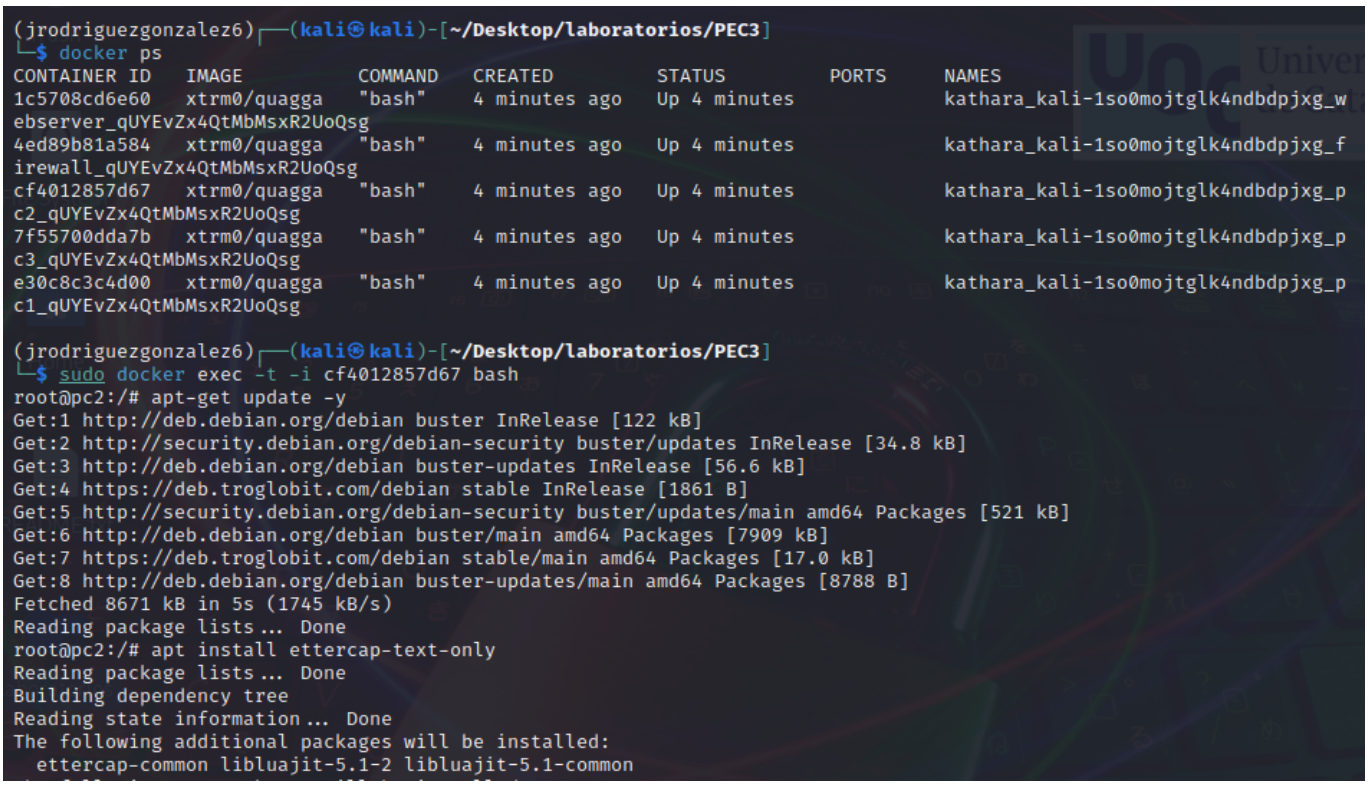
Ettercap es una herramienta de auditoría de red ampliamente utilizada para la interceptación activa y pasiva de comunicaciones en una red. Para ejecutar un ataque de DNS spoofing con Ettercap, necesitaremos modificar el archivo etter.dns (la configuración del DNS spoofing para Ettercap) y luego iniciar el ataque utilizando la interfaz de la terminal.

Para realizar este ejercicio, primero me tengo que colocar en el firewall y abrir el firewall para poder acceder al exterior e instalar ettercap en pc2.

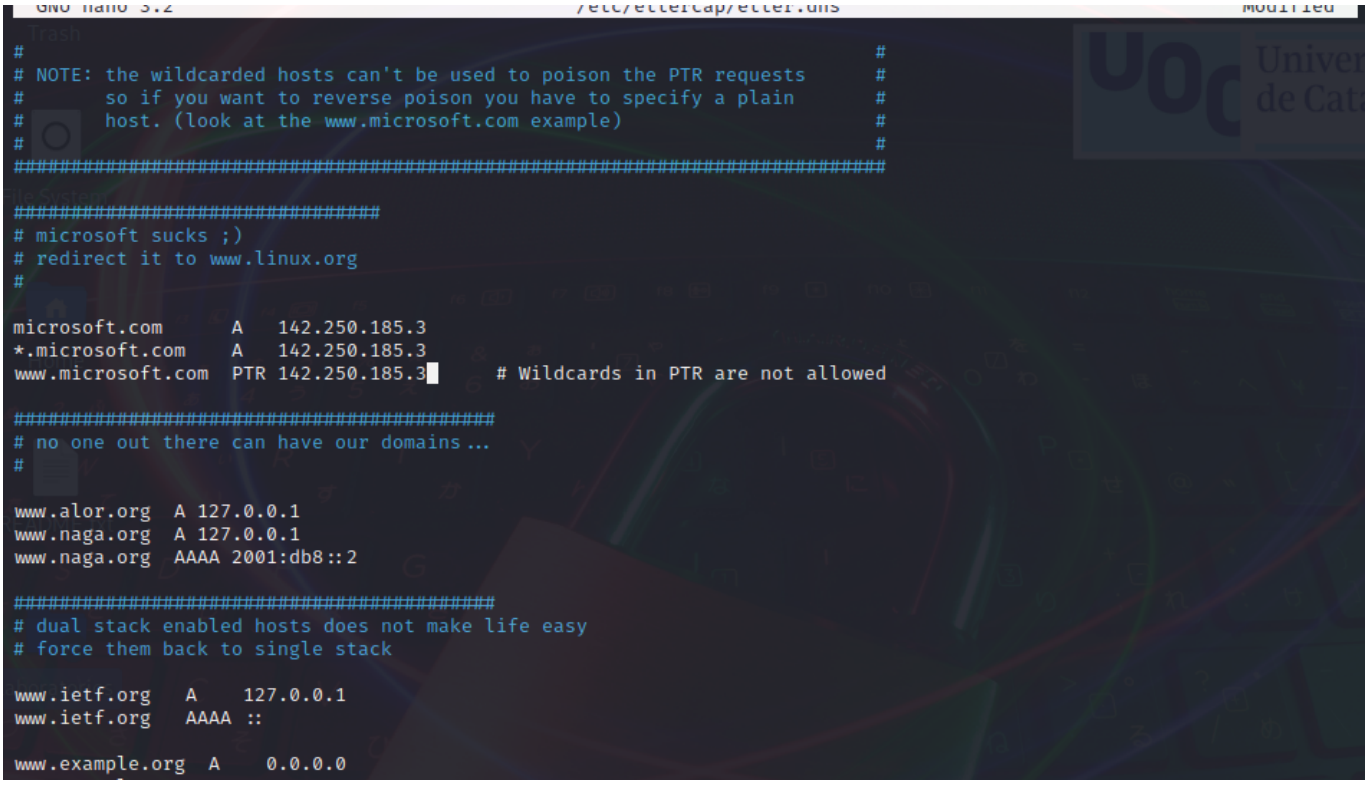
Procedemos a cambar la config del firewall.



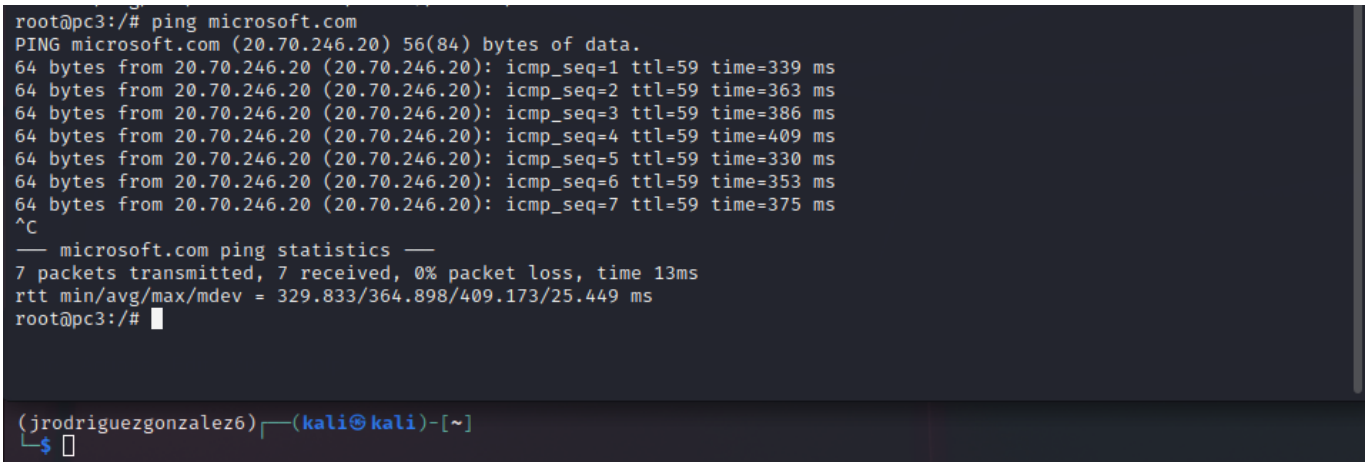
Entramos en pc2 actualizamos e instalamos ettercap.



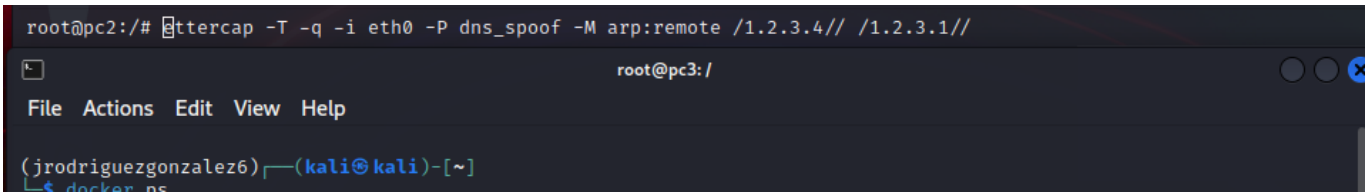
A continuación procedemos a modificar el archivo alojado en de google, cuya ip es en España 142.250.185.3, quedando el archivo como queda.



Nos metemos en PC3 y lo que vamos a hacer es hacer ping a microsoft.com, donde observamos que la ip es 20.70.246.20.

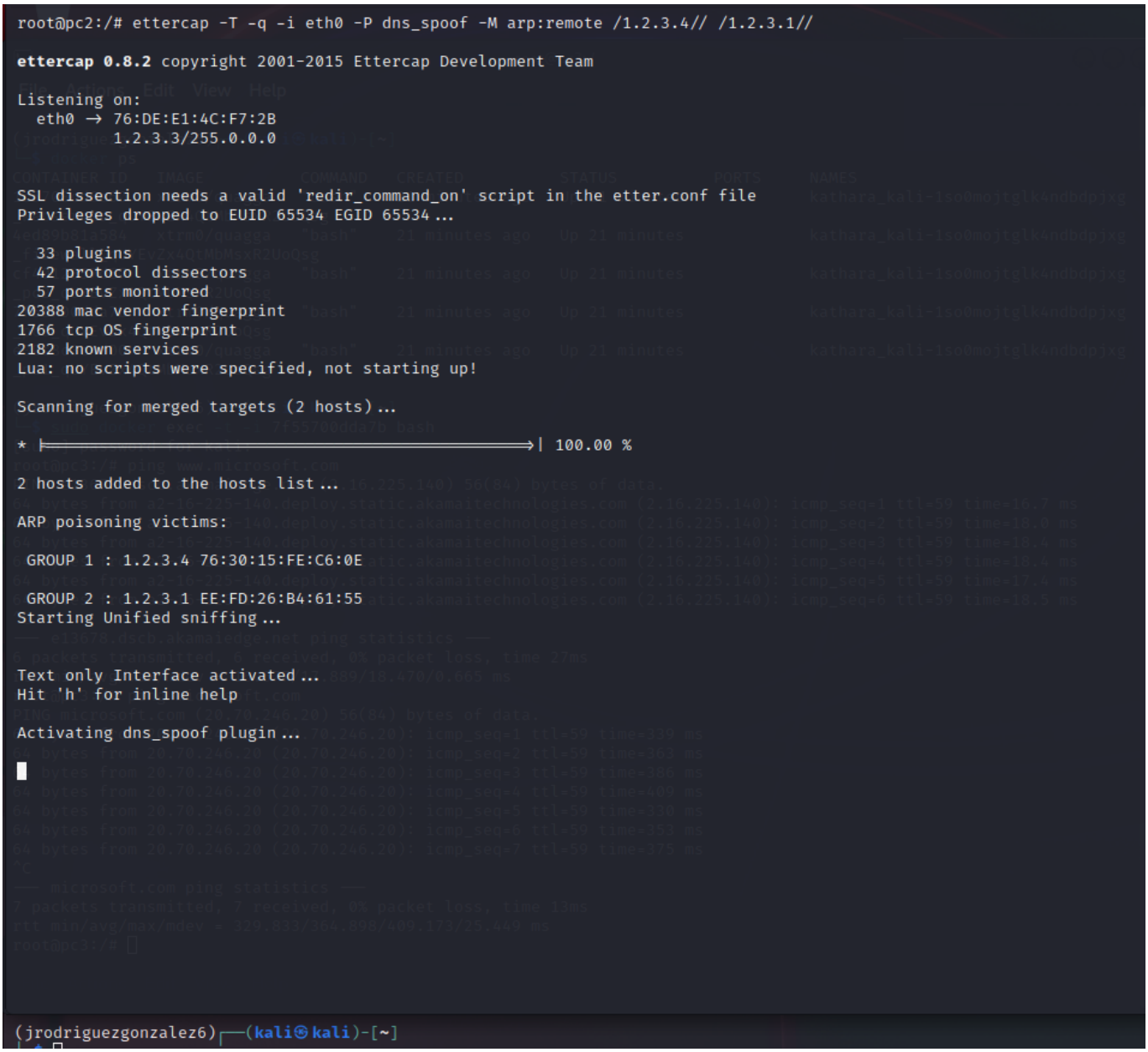


ahora volvemos a pc2, donde tenemos instalado ettercap y lanzamos el ataque.

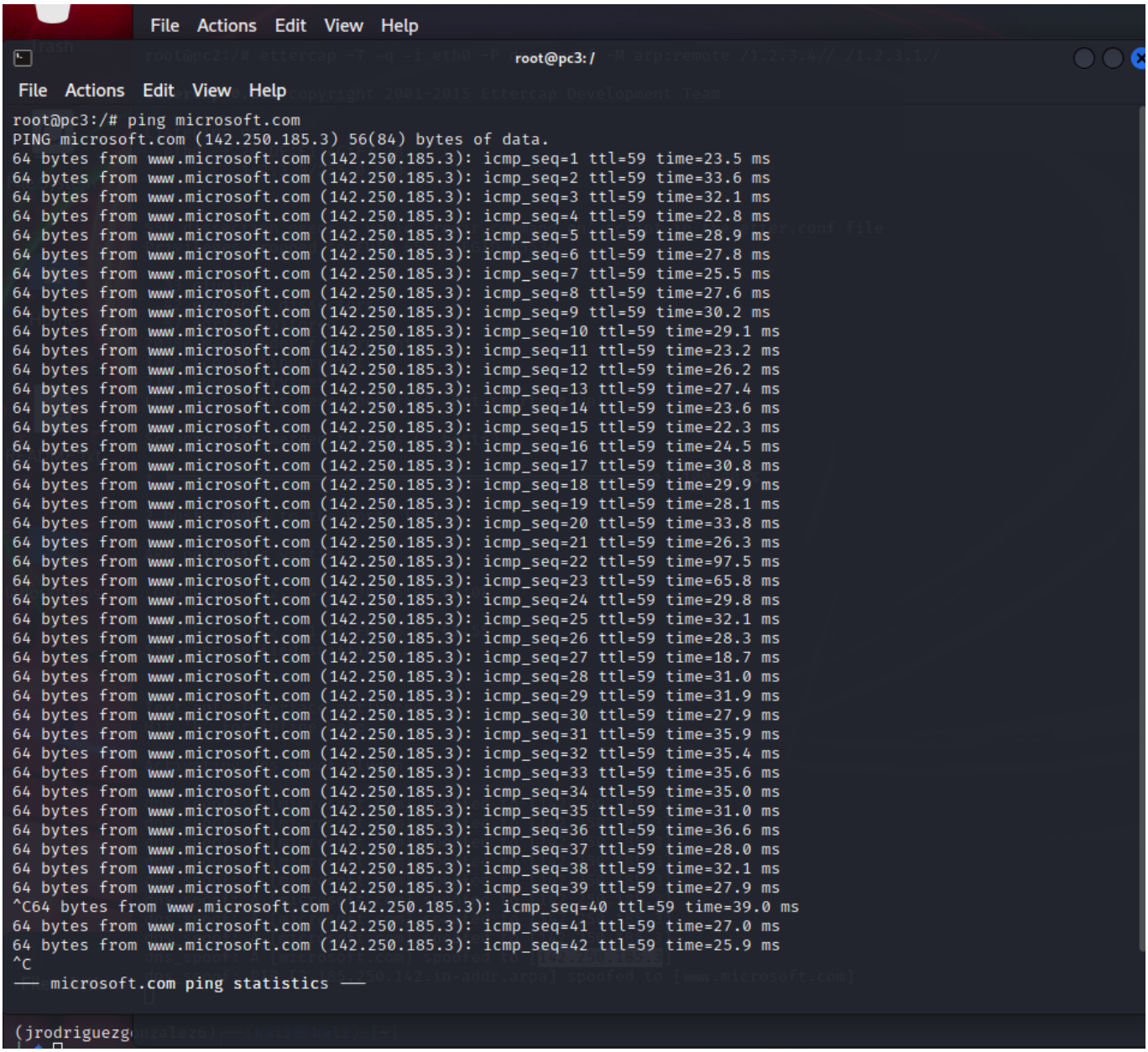


- -T: indica el modo de texto.
- -q es para el modo silencioso.
- -i eth0 selecciona la interfaz de red.
- -P dns_spoof selecciona el complemento de DNS spoofing
- -M arp:remote inicia un ataque ARP spoofing contra la víctima y el router, y /1.2.3.4// y /1.2.3.1// son las direcciones IP de la víctima y el router, respectivamente.

Lanzamos el ataque:

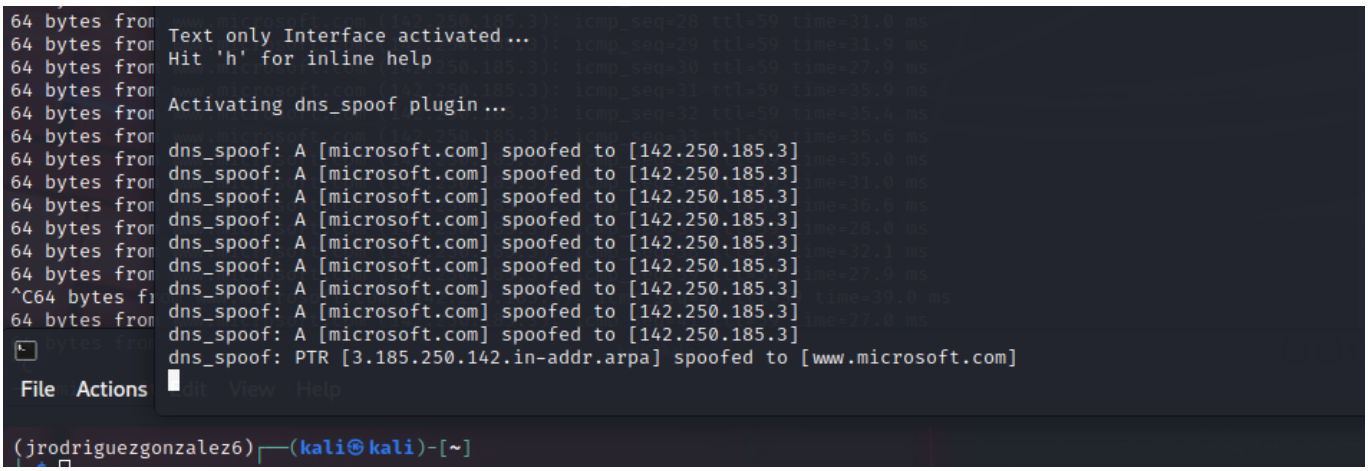


Entramos en pc3 e intentamos hacer ping a microsoft.com



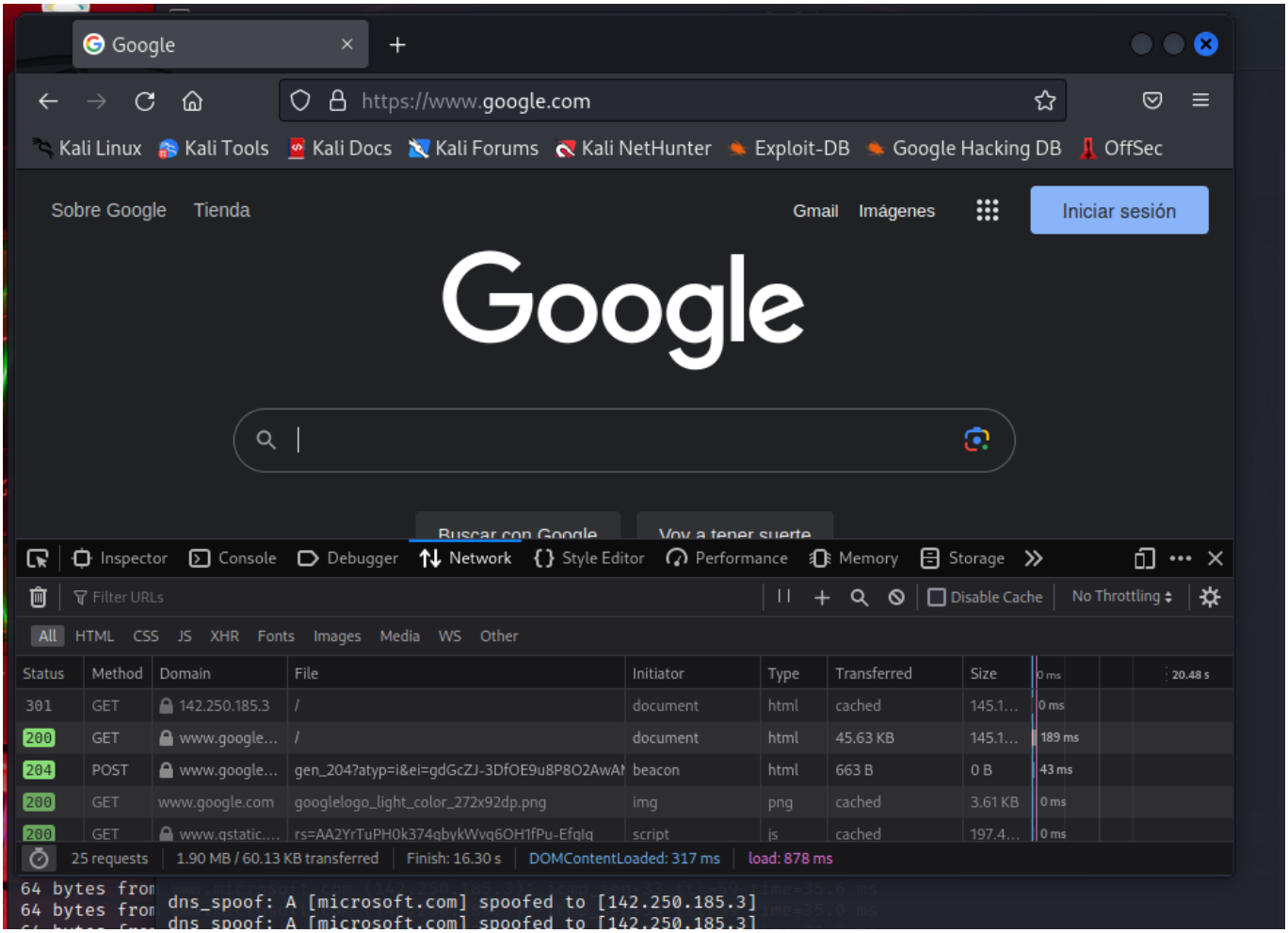
Podemos ver que ahora la dirección ip ha cambiado a 142.250.185.3.

Por otro lado en la consola de pc2, nos aparece lo siguiente.



Por ultimo para comprobar que redirige a google procedemos a acceder a esa dirección IP, abriendo la consola de red de herramientas del desarrollador. En este caso podemos ver que 142.250.185.3 nos da un

estatus del tipo 301, el cual significa que nos redirige a www.google.com tal y como aparece en la siguiente linea con un status 200,

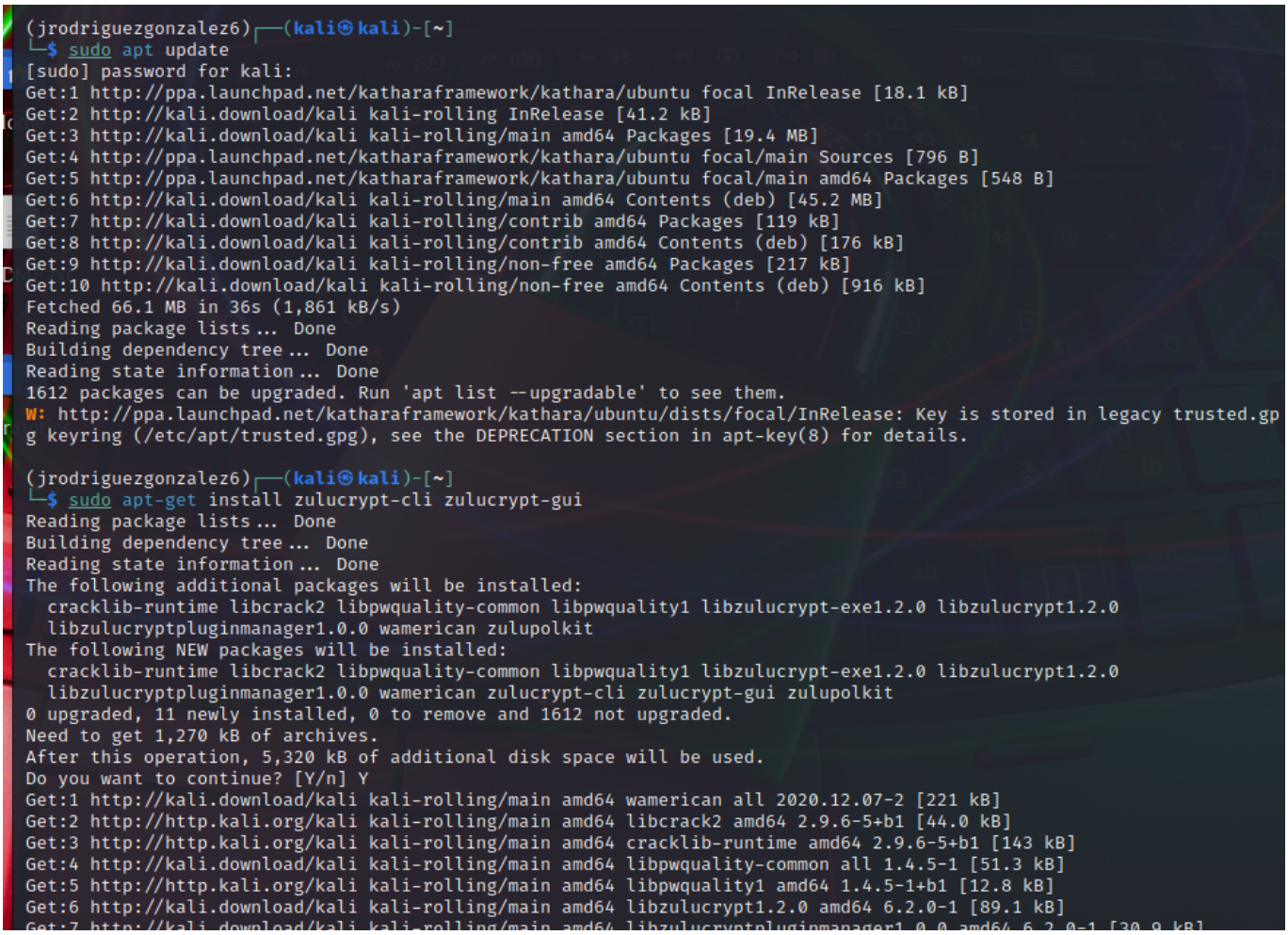


Volver al indice.

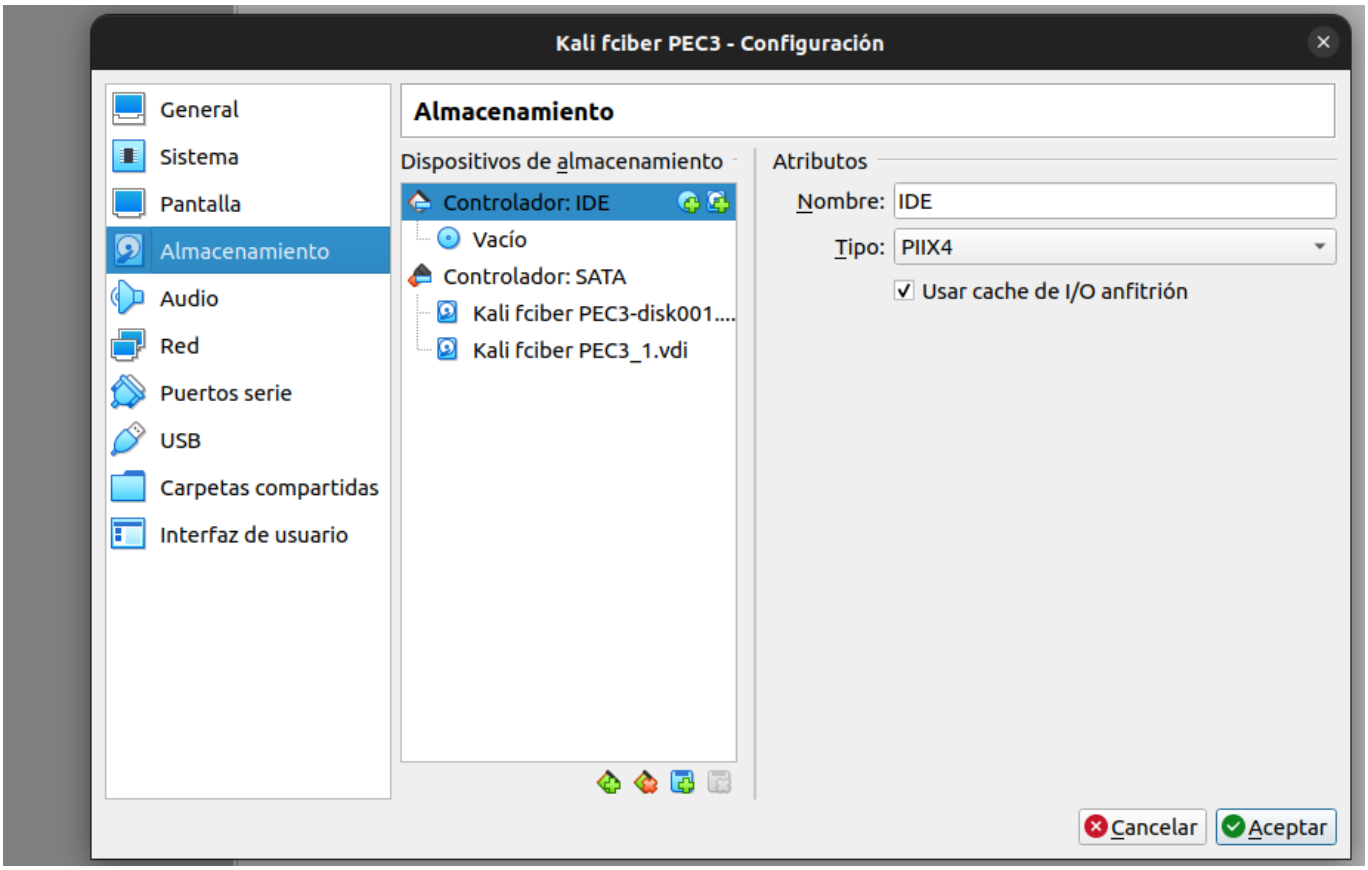
Respuesta a la pregunta 5.1.

Vamos a proceder primero a instalar zulucrypt, para ello introducimos los siguientes comandos en la terminal de kali:

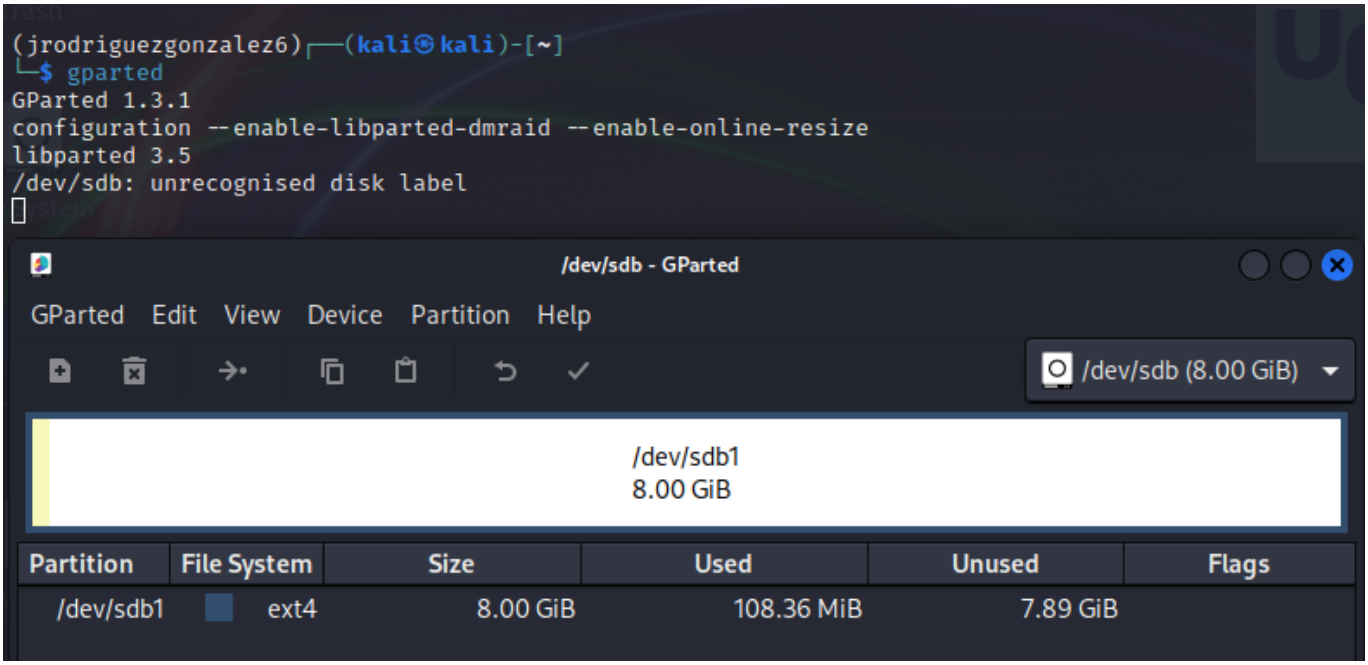
```
sudo apt update

sudo apt-get install zulucrypt-cli zulucrypt-gui
```



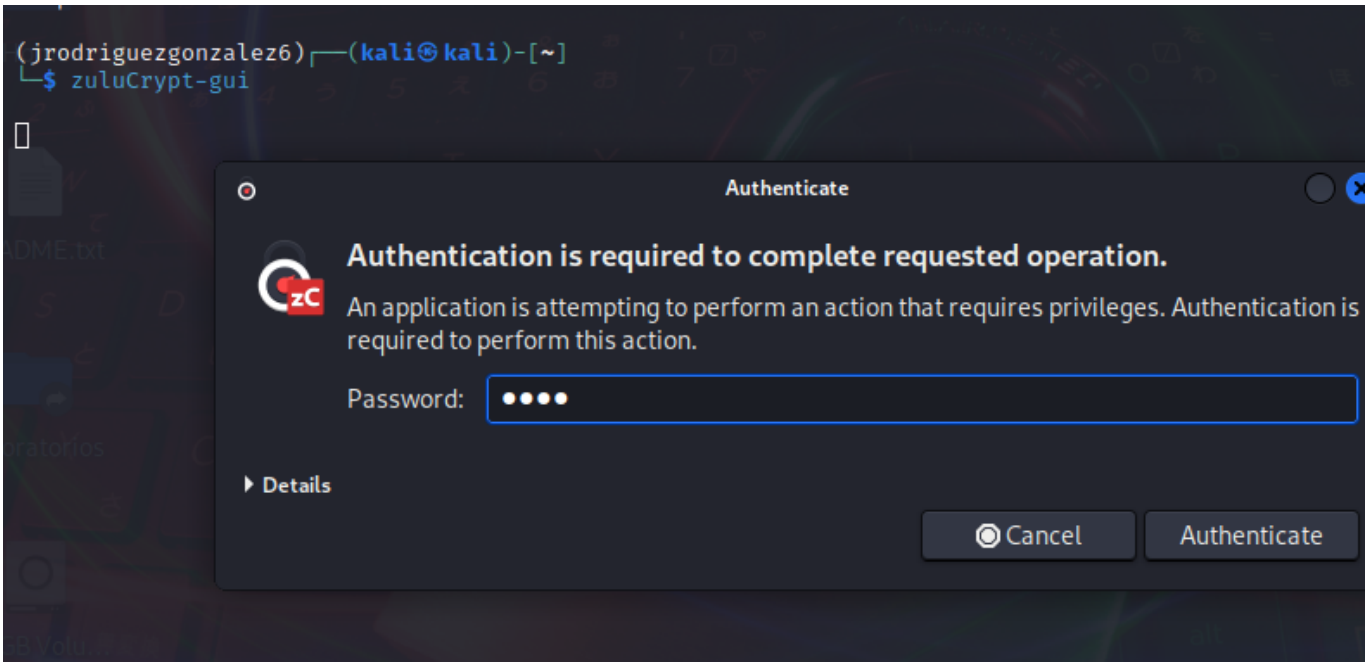
Ahora apagaremos kali para añadir un disco duro sin formatear dentro de la VM de Kali.



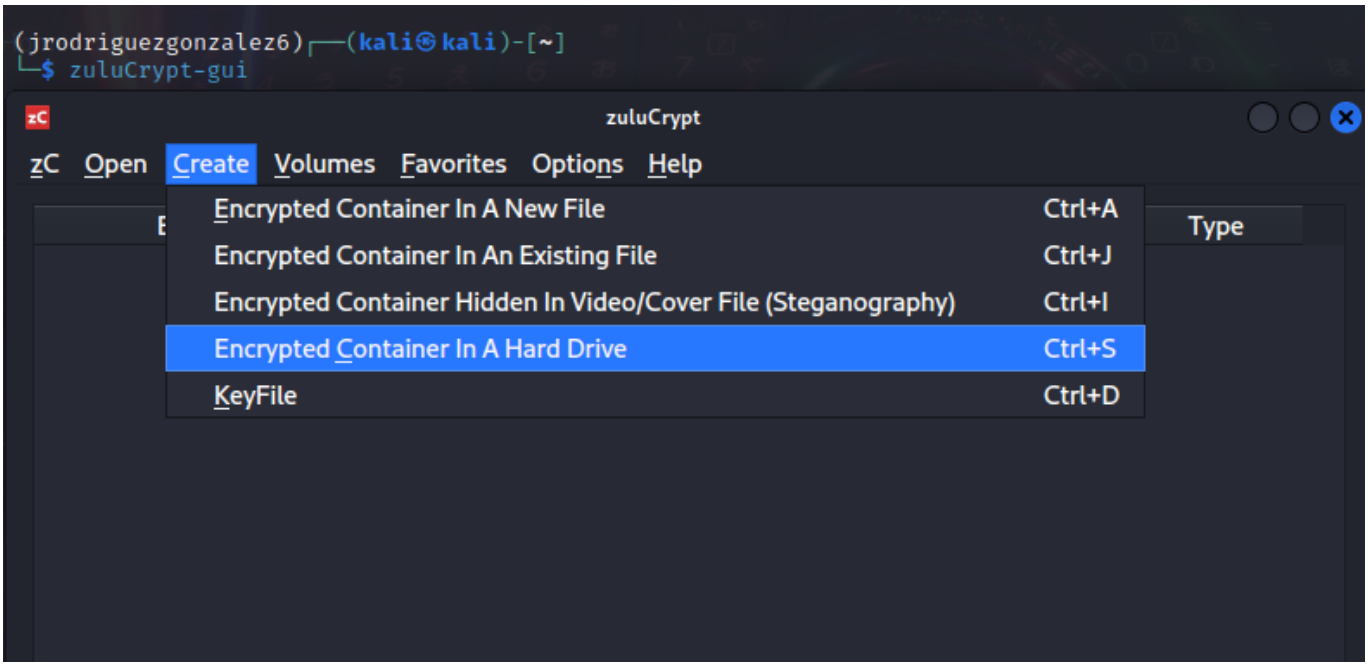
Procedemos a abrir la aplicación gparted y damos formato al disco duro insertado.



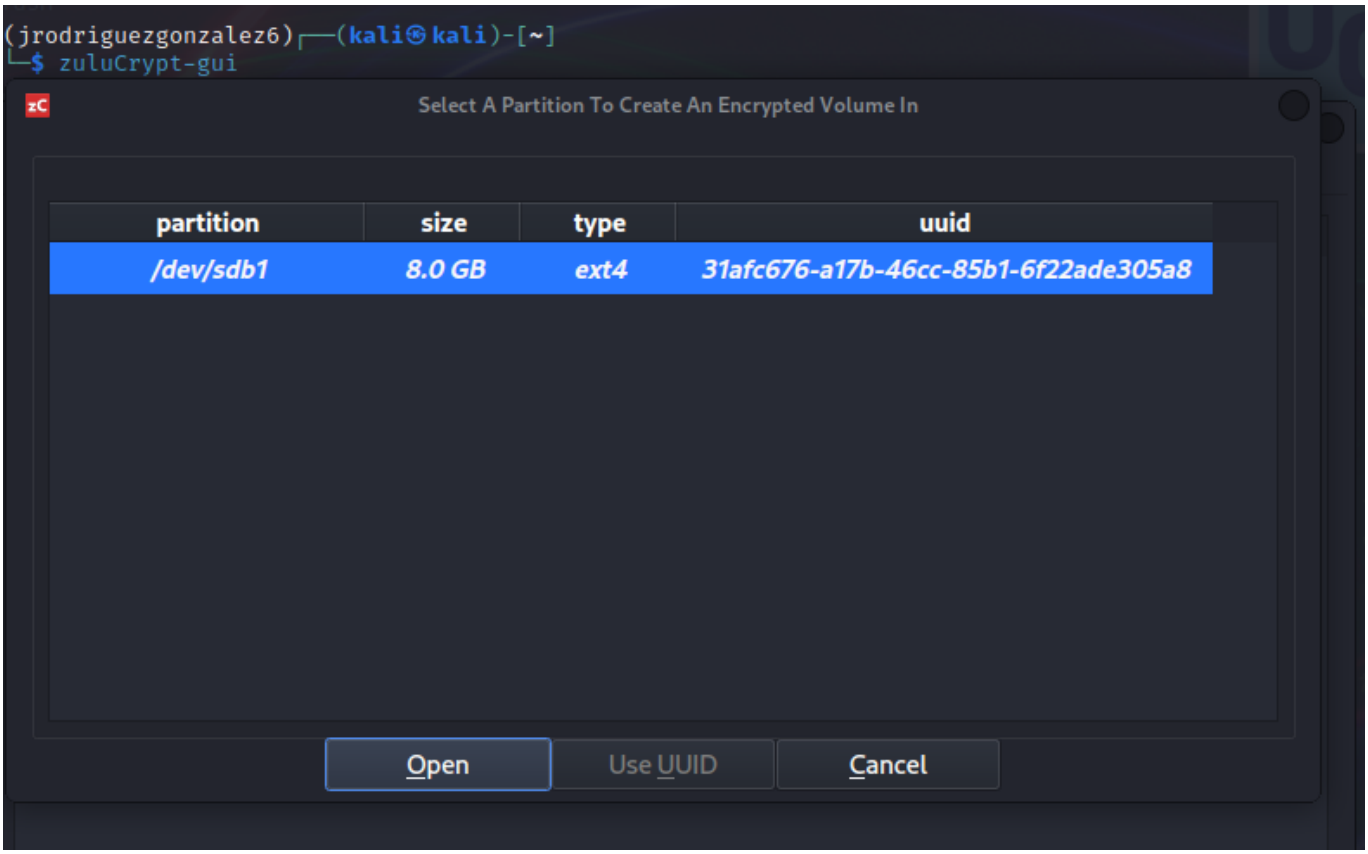
Arrancamos la aplicación con el comando `zuluCrypt-gui`.



A continuación, hacemos clic en create... Encrypted Container y a Hard Drive.

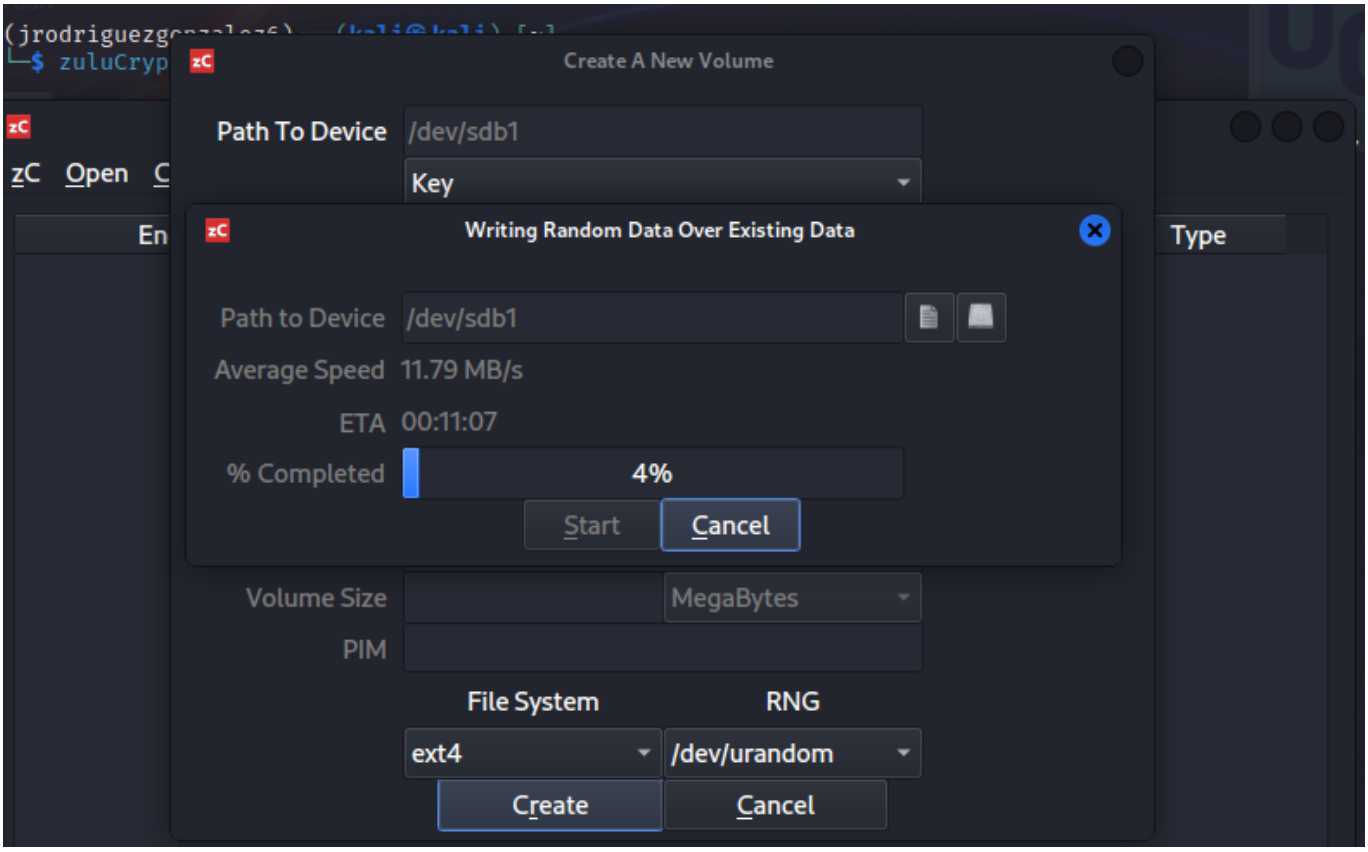


Seleccionamos el único disco duro `/dev/sdb1`, hacemos click en open

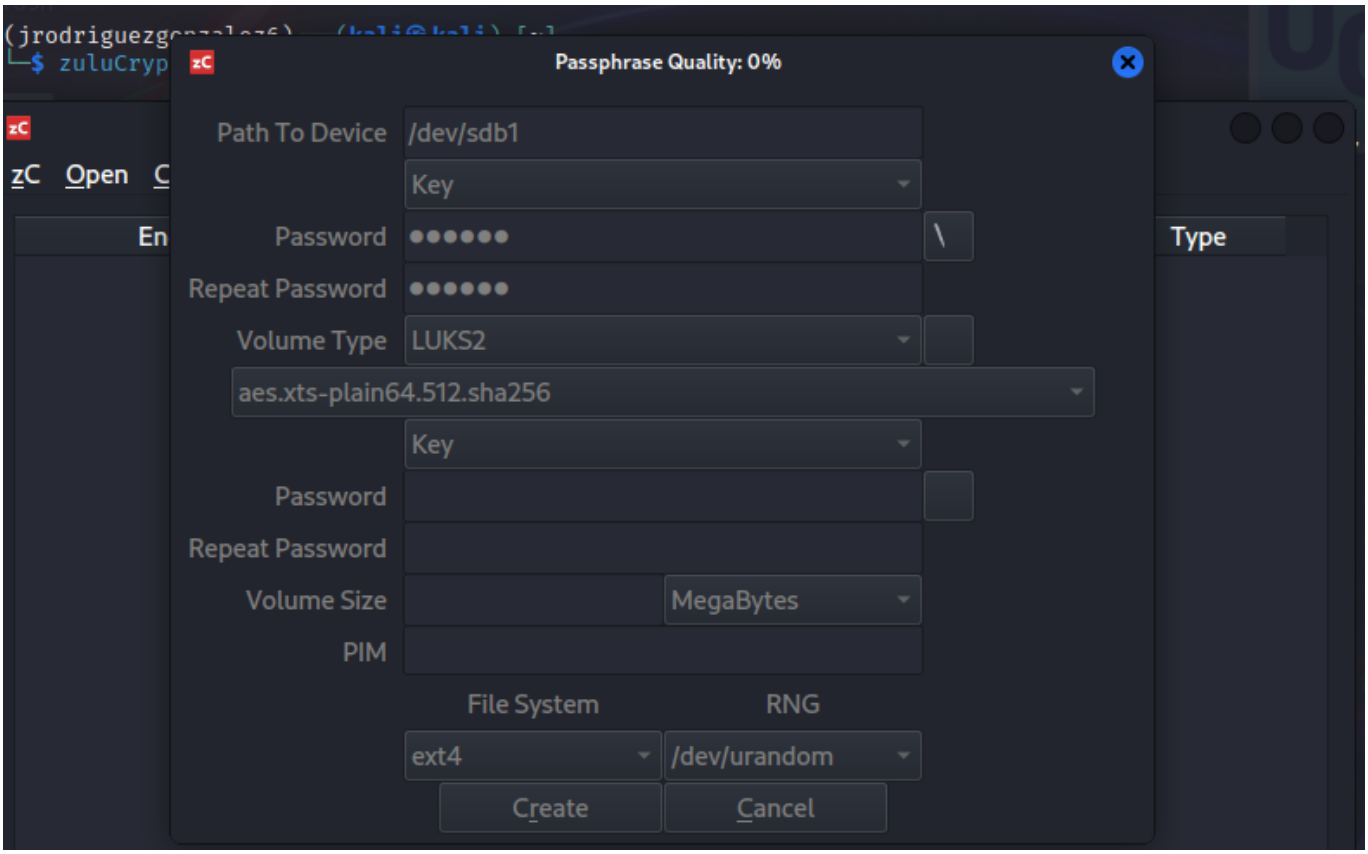


Aceptamos las advertencias de que podemos perder toda la documentación de este disco duro.

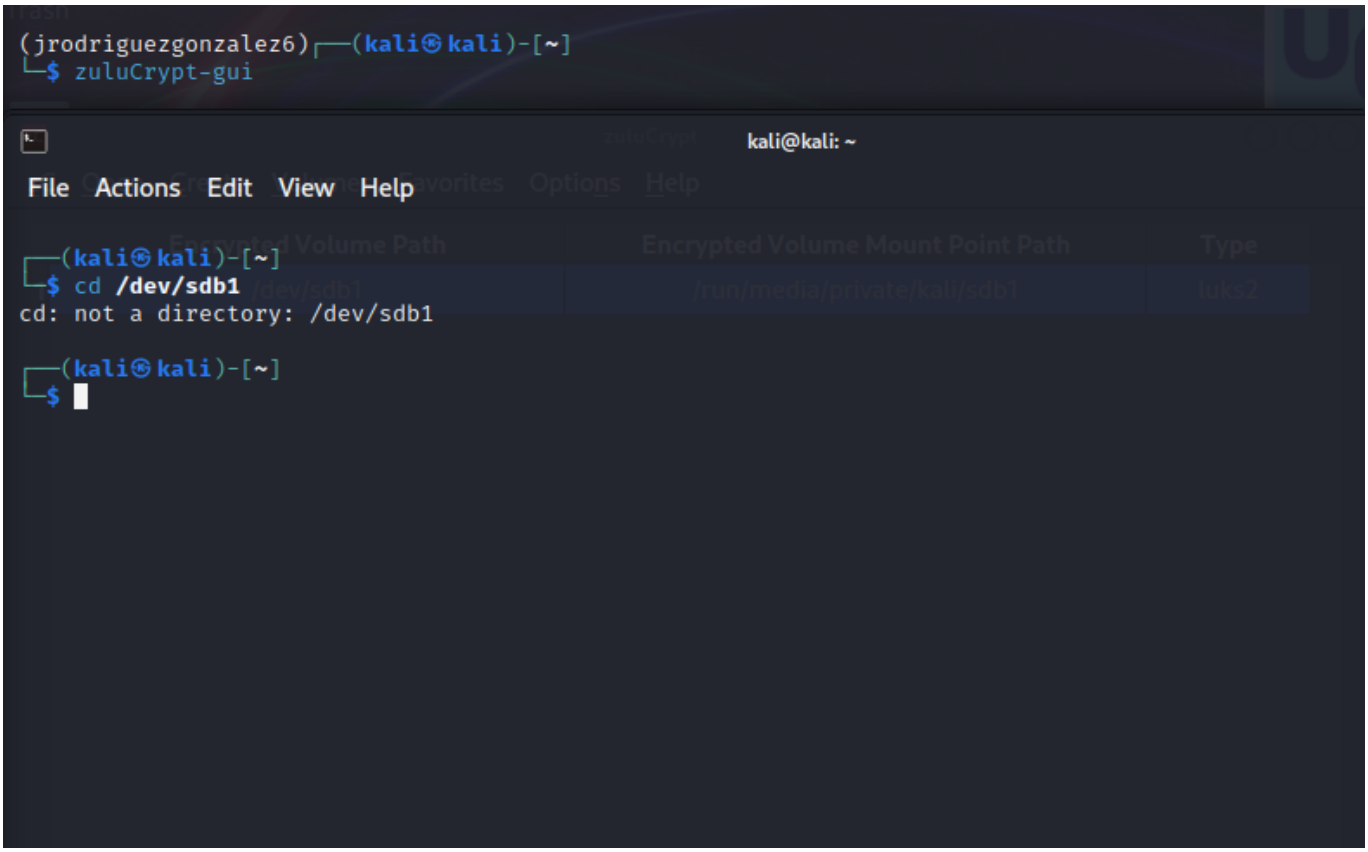
A continuación, se inicia el proceso de cifrado.



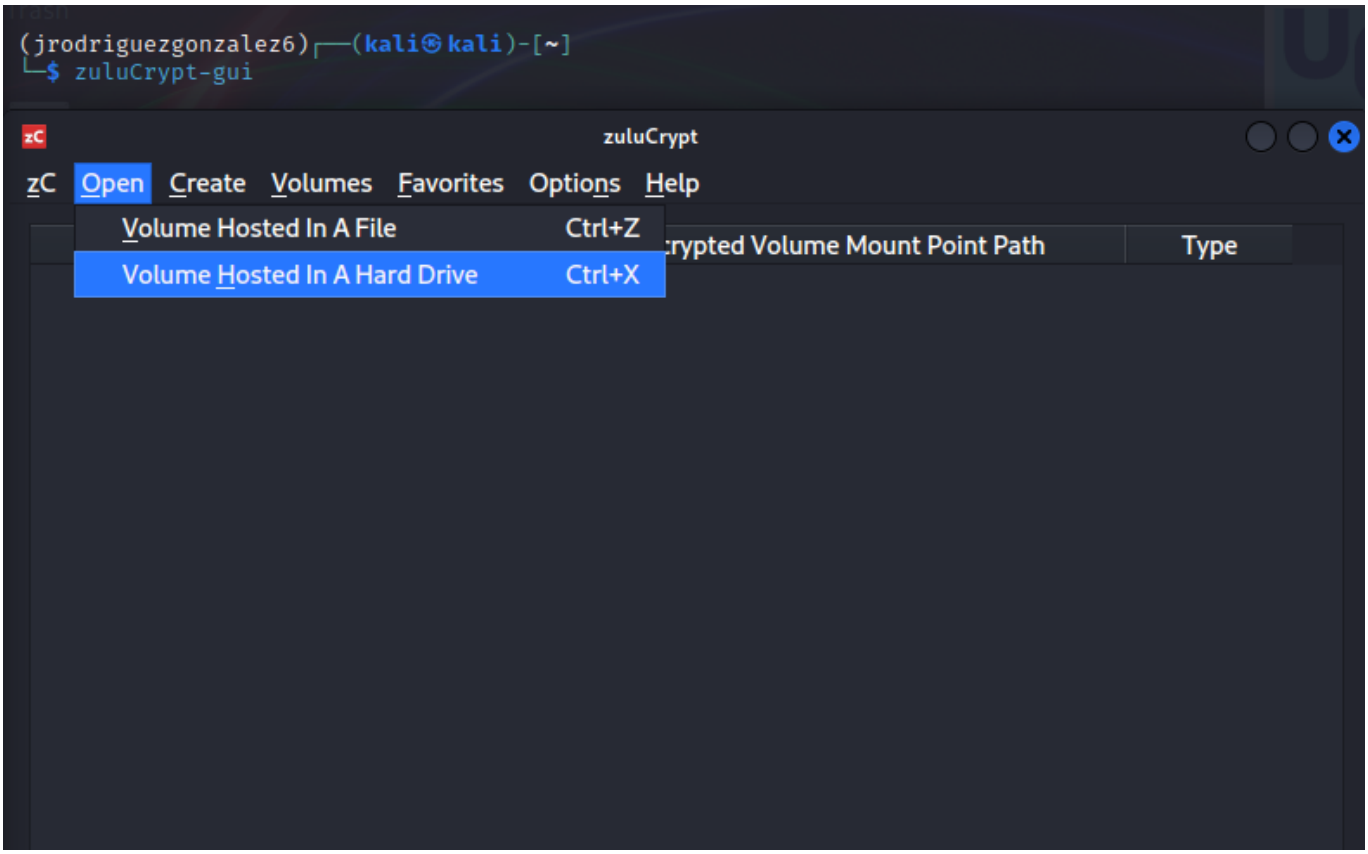
Por ultimo introducimos una contraseña.



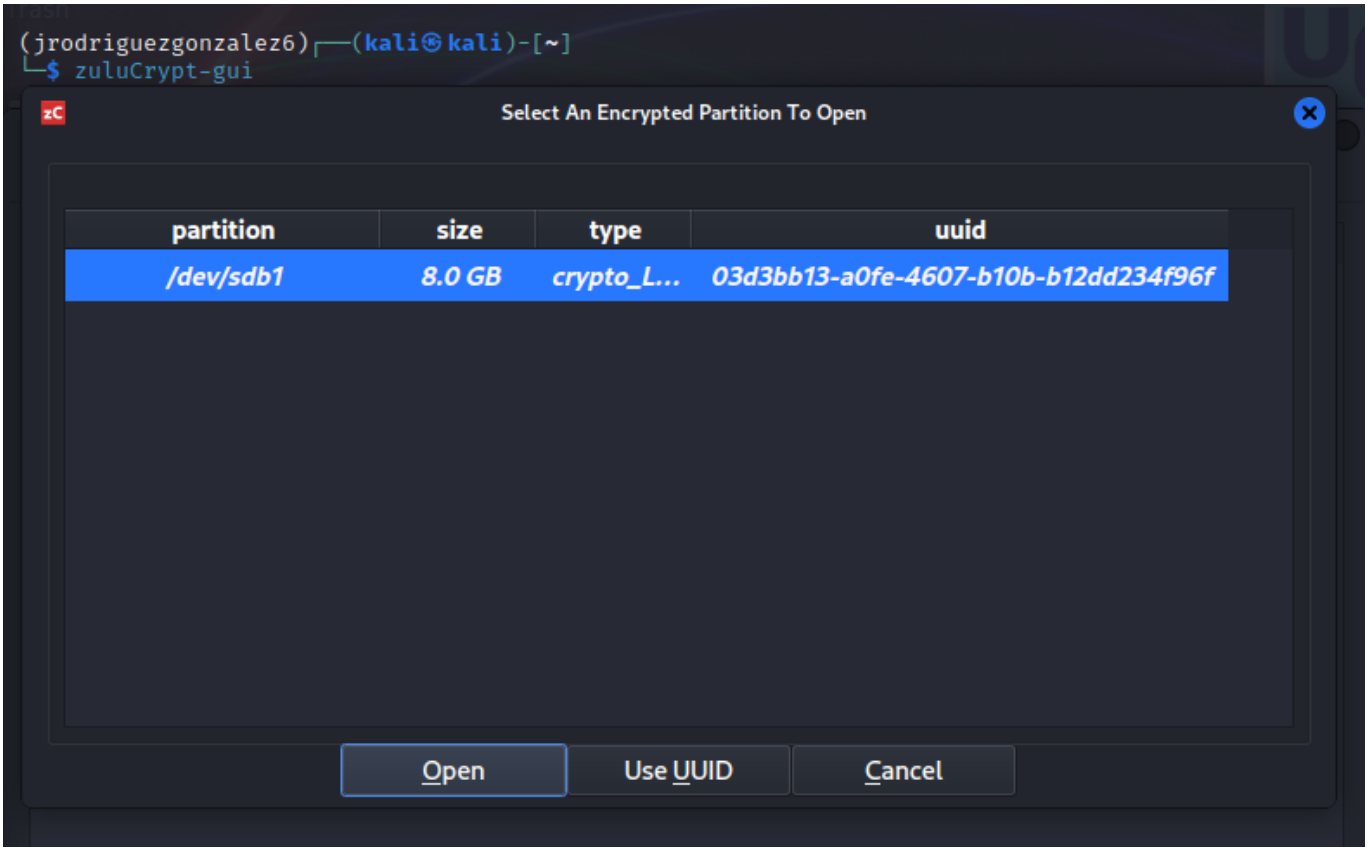
Ahora procederemos a intentar acceder a el desde la terminal, y nos dice que el directorio no existe, solo podemos acceder desde la aplicación `zuluCrypt`.

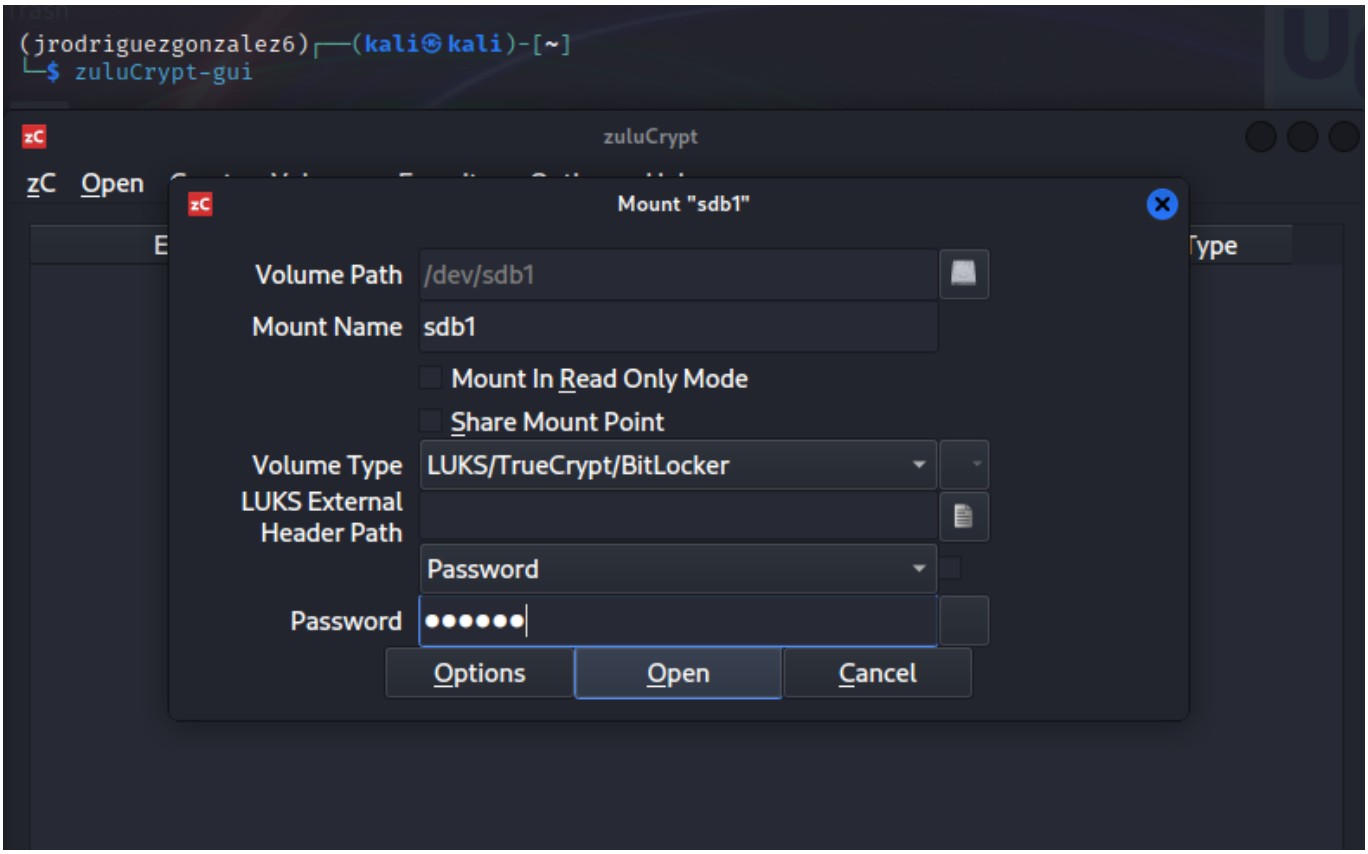


Para ello, entraremos y haremos click en Open... Volume Hosted in A Hard Drive.

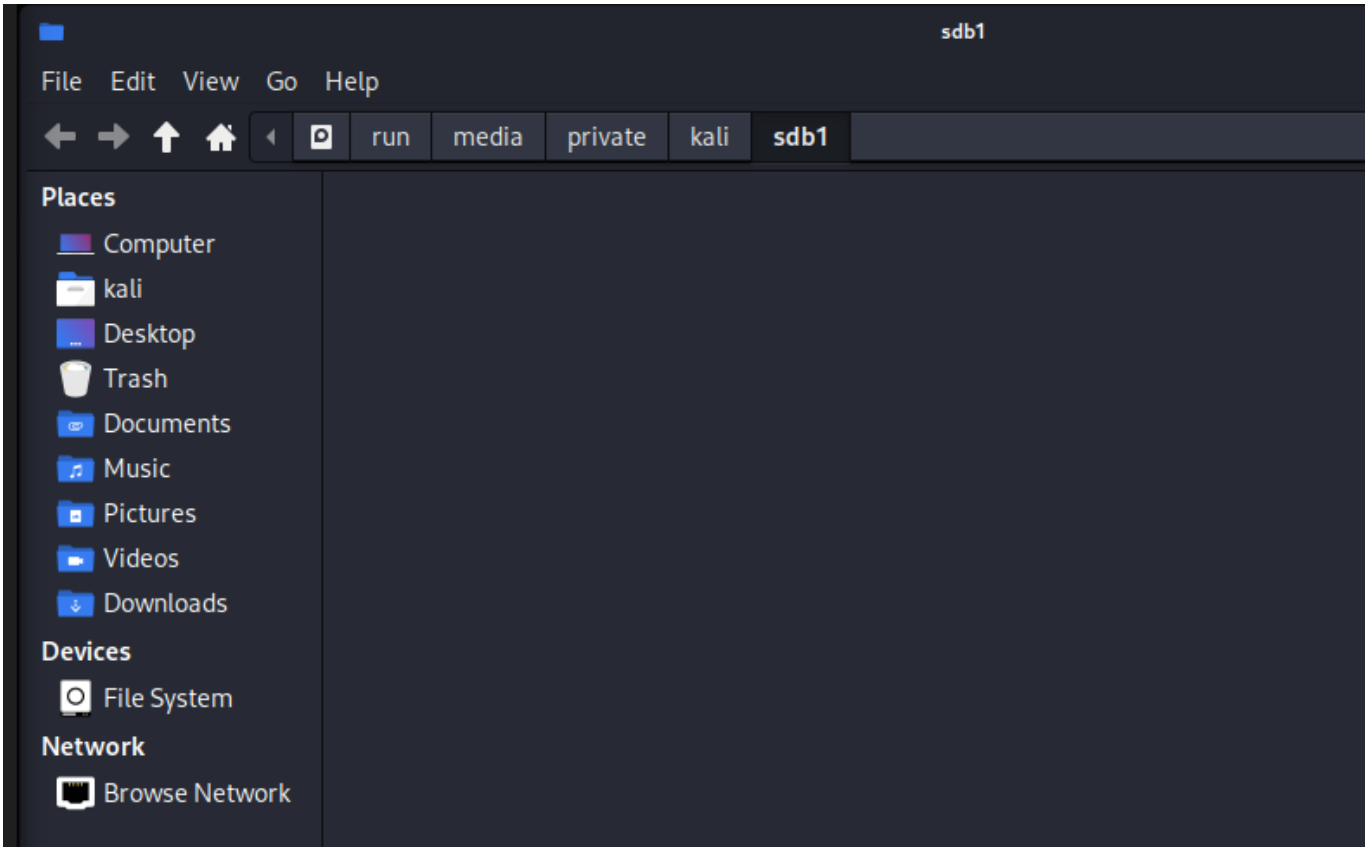


Seleccionamos la unidad, previo a su apertura, se procede a escribir la contraseña





Una vez escrita la contraseña, se podrá acceder a la unidad como con cualquier otra unidad sin cifrar.



Volver al indice.

Respuesta a la pregunta 5.2.

Snort es un potente sistema de detección de intrusiones de red (IDS) de código abierto que puede ser configurado para monitorizar el tráfico de red y generar alertas basadas en reglas definidas por el usuario.

Procederemos a instalar snort, según lo indicado en la web de luismiguelmorales.

Para configurar Snort para detectar y alertar sobre todas las conexiones SYN entrantes, tenemos que definir una regla adecuada en el archivo de configuración de Snort. Un ejemplo de como realizarlo es el siguiente.

Abriremos en consola el archivo `/etc/snort/snort.conf`, este archivo es el archivo de configuración de snort, procederemos a introducir ls siguiente linea en el archivo

```
alert tcp any any -> $HOME_NET any (flags: S; msg: "SYN packet detected";
sid: 1000001;)
```

- `alert` indica a Snort que genere una alerta cuando se cumpla esta regla.
- `tcp` es el protocolo que se va a monitorizar.
- `any any` significa que la regla se aplicará a cualquier tráfico de cualquier dirección IP y cualquier puerto.
- `$HOME_NET any` indica que la regla se aplicará a cualquier tráfico destinado a la red doméstica en cualquier puerto. $HOME_NET es una variable que normalmente se define en el archivo de configuración de Snort para representar la red que se está protegiendo.
- `flags: S;` indica que la regla se aplicará a los paquetes con la bandera SYN establecida.
- `msg: "SYN packet detected";` define el mensaje que se mostrará cuando se genere una alerta. **sid: 1000001**; es un identificador único para la regla.

Procedemos a guardar y cerrar el editor nano. Por ultimo reiniciamos el servicio de Snort con el comando `sudo service snort restart`.

De este modo Snort alertará al usuario por cada conexión entrante.

Volver al indice.

Respuesta a la pregunta 5.3.

Para permitir solo conexiones SSH desde fuera (es decir, desde la red externa) y bloquear todas las conexiones SSH desde las máquinas internas (PC1 y PC2) utilizando iptables dentro de pc3, ejecutaremos primero los siguientes comandos para que pc1, pc2 y el desde el firewall no se permitan conexiones entrantes.

```
sudo iptables -A INPUT -s 1.2.3.2 -p tcp --dport 22 -j DROP

sudo iptables -A INPUT -s 1.2.3.3 -p tcp --dport 22 -j DROP

sudo iptables -A INPUT -s 1.2.3.4 -p tcp --dport 22 -j DROP
```

Por otro lado ejecutaremos iptables para que permita cualquier conexión ssh a traves de otra dirección IP desde el exterior.

```
sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -
j ACCEPT
```

Por ultimo, guardaremos la configuración con el comando `iptables-save`

Por ultimo, siempre se recomienda verificar las reglas de iptables con el comando sudo iptables -L -v para asegurarnos de que se han aplicado correctamente.

Volver al indice.
