Forensic100

By: KALRONG

Enunciado:

Esta maldita imagen 100 points

Se que la he dejado en algún sitio, pero dónde..

Fichero para la prueba: https://mega.nz/#!YcBw2A4K!jUqgKnf5YZhmjZtpNEH7jM3h2rQ4KyvrOiBaLYmdWys

Formato de la clave
- SHA256 de la clave
- Sin espacios
- Mayúsculas, minúsculas, números y caracteres especiales tal y como aparece en el fichero de la prueba

Solución:

En esta prueba se nos pide descargarnos un archivo tar.gz que contiene el archivo image.raw.

File nos devuelve la siguiente información:

```
imagen.raw: DOS/MBR boot sector; partition 1 : ID=0xee, start-CHS (0x0,0,2), end-CHS (0x3ff,255,63), startsector 1, 5242879 sectors
```

Parece ser que es una imagen de un disco, el siguiente paso será ver que dice fdisk al respecto:

```
root@drakon:/home/kalrong# fdisk -l imagen.raw
Disco imagen.raw: 2,5 GiB, 2684354560 bytes, 5242880 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: gpt
Identificador del disco: 0D48A8DC-4A74-11E5-88B4-080027330CE1

Disposit.   Comienzo   Final Sectores Tamaño Tipo
imagen.raw1      128 4194431  4194304     2G swap de FreeBSD
imagen.raw2  4194432 5242839  1048408 511,9M ZFS de FreeBSD
```

Parece que casi todo es swap y tenemos una particion de 500Mb en formato ZFS. El problema es que ZFS no permite file I/O así que no podemos montarla directamente, primero, debemos utilizar losetup para montarla y que zfs la pueda reconocer.

Para poder hacer esto tenemos que calcular el offset de dicha partición, sabiendo que los bloques son de 512 bytes y que la partición comienza en 4194432:
512 * 4194432 = 2147549184

Una vez conocido el offset ejecutaremos losetup de la siguiente manera:

```
root@drakon:/home/kalrong# losetup --offset 2147549184 /dev/loop2 /home/kalrong/imagen.raw
```

Para realizar los siguientes pasos necesitareis tener instalador y cargados los modulos de ZFS. En mi caso tuve ciertos problemas de versiones ya que la utilizada para la prueba era más nueva que la de los repositorios con lo cual tuve que compilar los módulos desde los sources.

Utilizamos zpool para importar la pool de ZFS:

```
root@drakon:/home/kalrong# zpool import -d /dev
   pool: Mugar2x3
     id: 4184644559435377404
  state: ONLINE
 status: The pool was last accessed by another system.
 action: The pool can be imported using its name or numeric identifier and
        the '-f' flag.
   see: http://zfsonlinux.org/msg/ZFS-8000-EY
 config:

        Mugar2x3    ONLINE
          loop2     ONLINE
```

Zpool nos avisa de que la pool fue usada en otro sistema previamente con lo cual tenemos que forzar con la opcion -f la importación y además utilizar el nombre de la misma (Mugar2x3); y confirmamos con zfs que la pool esta importada:

```
root@drakon:/home/kalrong# zpool import -f -d /dev Mugar2x3
root@drakon:/home/kalrong# zfs list
NAME                                                 USED  AVAIL  REFER  MOUNTPOINT
Mugar2x3                                            11,8M   452M    96K  /Mugar2x3
Mugar2x3/.system                                    1,77M   452M   104K  legacy
Mugar2x3/.system/configs-f1ae6c68bbe041c7bb38cadeec088781   96K   452M    96K  legacy
Mugar2x3/.system/cores                               720K   452M   720K  legacy
Mugar2x3/.system/rrd-f1ae6c68bbe041c7bb38cadeec088781    96K   452M    96K  legacy
Mugar2x3/.system/samba4                              540K   452M   540K  legacy
Mugar2x3/.system/syslog-f1ae6c68bbe041c7bb38cadeec088781   256K   452M   256K  legacy
Mugar2x3/AlalaLA                                    9,00M   452M  2,48M  /Mugar2x3/AlalaLA
```

Con esto la partición ya debería de estar montada en el root de nuestro sistema bajo la carpeta Mugar2x3. En caso de que no fuera así se podría montar utilizando el siguiente comando:

zfs mount Mugar2x3

Veremos que esta solo contiene varias subcarpetas y un monton de archivos con el famoso “Lorem ipsum”, pero ninguna imágen que es lo que nos piden en el reto.

ZFS tiene la capacidad de generar spanpshots para poder recuperar estados pasados del sistema de archivos, echamos un vistazo y efectivamente comprobamos que existen varios de los mismos:

```
root@drakon:/home/kalrong# zfs list -t snapshot
NAME                          USED  AVAIL  REFER  MOUNTPOINT
Mugar2x3/AlalaLA@20150824      80K      -    96K  -
Mugar2x3/AlalaLA@Original      64K      -    96K  -
Mugar2x3/AlalaLA@1            136K      -   576K  -
Mugar2x3/AlalaLA@2            328K      -  1,51M  -
Mugar2x3/AlalaLA@4           1,12M      -  3,38M  -
Mugar2x3/AlalaLA@8            160K      -  2,46M  -
Mugar2x3/AlalaLA@3           1,10M      -  2,94M  -
Mugar2x3/AlalaLA@5            560K      -  2,48M  -
Mugar2x3/AlalaLA@6            144K      -  2,48M  -
Mugar2x3/AlalaLA@7            152K      -  1,53M  -
Mugar2x3/AlalaLA@10            64K      -  2,48M  -
```

Los snapshots aparecen ordenados de más antiguo a más nuevo, en caso de que nos pasemos debemos destruir e importar la pool de nuevo.

En este caso la que parecia la mejor opción para comenzar era el snapshot 3 por el tamaño, así que procedemos a hacer rollback al mismo:

```
root@drakon:/home/kalrong# zfs rollback -r Mugar2x3/AlalaLA@3
```

Como seguimos teniendo un montón de archivos de por medio utilice el comando file para localizar la imagen:

```
Imagenes/c3a6217d80b836c69e5cdf4255629890:   PNG image data, 1152 x 648, 8-bit/color RGB, non-interlaced
```

Abrimos el archivo y voila:

Calculamos el SHA256 y ya tenemos la prueba superada.