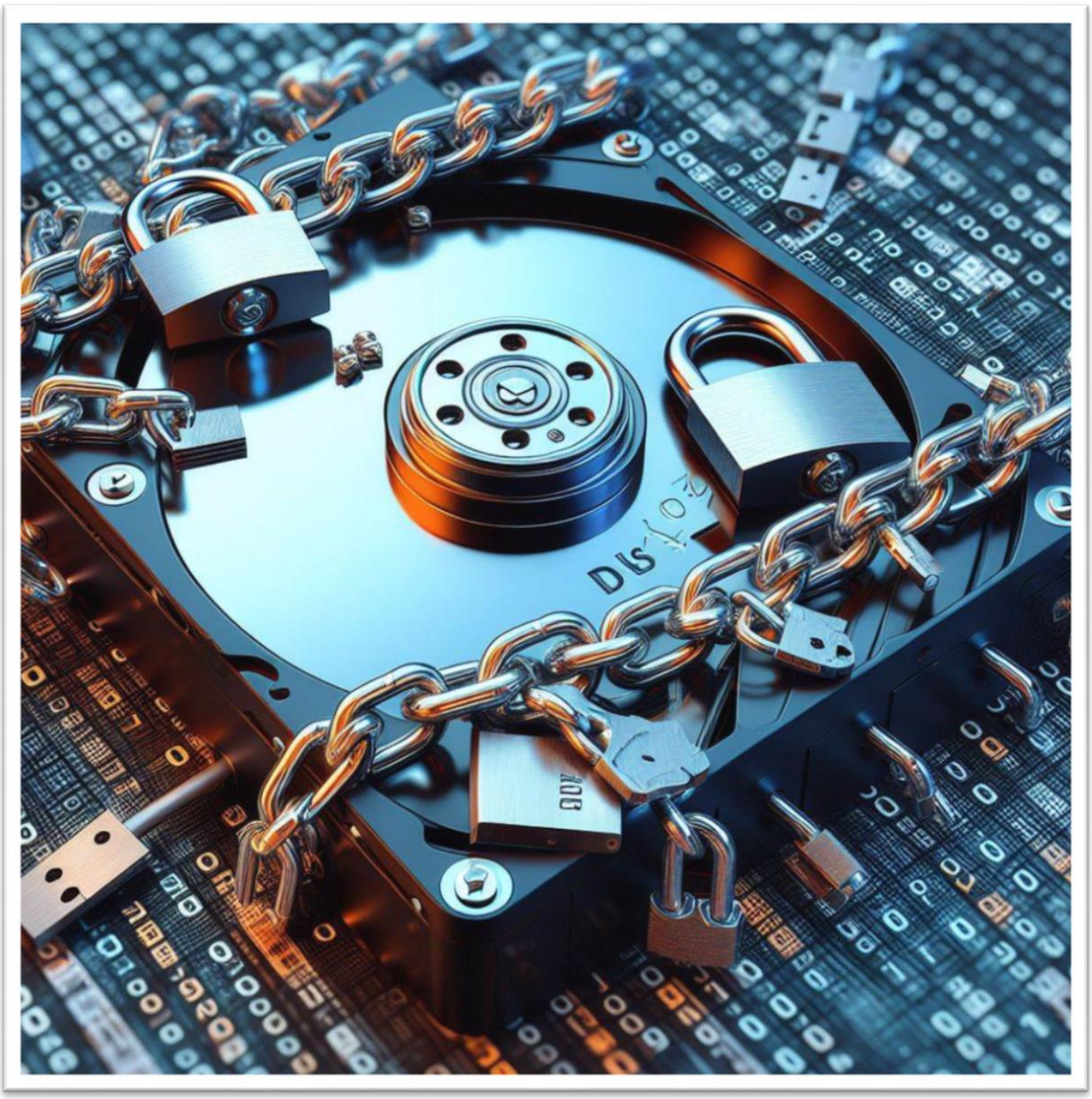
25 DE ABRIL DE 2024



ACT0603-LUKS

BASTIONADO DE REDES Y SISTEMEAS

ERIC SERRANO MARÍN
I.E.S MARTINEZ MONTAÑES
CETI

Contenido

Paso 1: Instalación de cryptsetup utility en Linux.

```
root@ubuntu14:/home/usuario# apt-get install cryptsetup
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  cryptsetup-bin libcryptsetup4
The following NEW packages will be installed:
  cryptsetup cryptsetup-bin libcryptsetup4
0 upgraded, 3 newly installed, 0 to remove and 64 not upgraded.
Need to get 273 kB of archives.
After this operation, 848 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://es.archive.ubuntu.com/ubuntu/ trusty/main libcryptsetup4 amd6
4 2:1.6.1-1ubuntu1 [77,0 kB]
```

En caso de tener que encriptar en RHEL, CentOS, Oracle o Scientific Linux, lo haremos con el siguiente comando: ***yum install cryptsetup-luks.***

Y si tuviésemos que hacerlo en Fedora Linux: ***dnf install cryptsetup-luks.***

Paso 2: Configuración de las particiones en LUKS.

Para este paso hay que tener cuidado con que partición cogemos, ya que va a borrar todos los datos de la partición que vamos a encriptar. En mi caso voy a crear un nuevo disco de 11.4GB.

Salida del comando fdisk –l

```
Disk /dev/sdb: 11.4 GB, 11377393664 bytes
255 heads, 63 sectors/track, 1383 cylinders, total 22221472 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

Encriptación de disco /dev/sdb

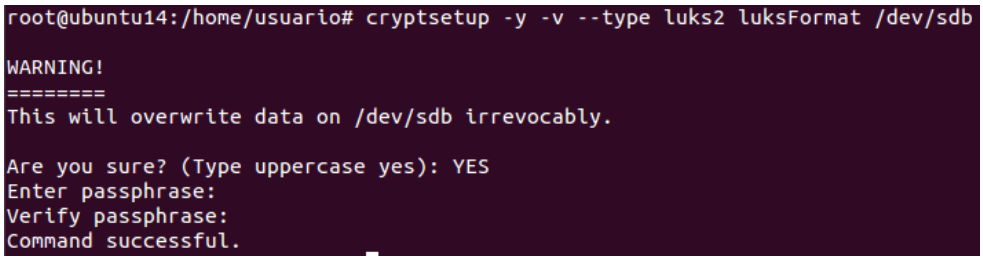
Podemos observar que se ha realizado correctamente.

```
root@ubuntu14:/home/usuario# cryptsetup -y -v luksFormat /dev/sdb

WARNING!
========
This will overwrite data on /dev/sdb irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase:
Verify passphrase:
Command successful.
```

Al leer las instrucciones, me enteré de que LUKS2 tiene más seguridad y opciones extra comparadas con LUKS1. Así que decidí cambiar a LUKS2 para asegurarme de que mis datos estén mejor protegidos y para aprovechar todas las características nuevas que ofrece esta versión.



Comparativa entre LUKS1 y LUKS2.

Ya que estaba destacando las ventajas de LUKS2, me pareció razonable proporcionar una comparativa para que se vea aún más las mejoras y beneficios de esta última versión.

Característica	LUKS1	LUKS2
Seguridad	Utiliza el cifrado CBC por defecto, susceptible a ciertos ataques.	Permite la selección de algoritmos de cifrado más modernos y seguros.
Flexibilidad	Limitado en la cantidad y tipo de algoritmos de cifrado.	Más flexible en la selección de algoritmos de cifrado y configuraciones.
Funcionalidad	Carece de algunas características avanzadas como el cifrado de datos en reposo.	Incluye soporte para el cifrado de datos en reposo y cifrado de archivos individuales.
Compatibilidad	Ampliamente compatible con la mayoría de las distribuciones de Linux.	Puede no ser completamente compatible con todas las herramientas y distribuciones de Linux, pero su compatibilidad ha ido mejorando.

Creación de mapeo para acceder al volumen encriptado.

Este comando inicializa el volumen y establece una clave inicial o una frase de contraseña.

```
root@ubuntu14:/home/usuario# cryptsetup luksOpen /dev/sdb backup2
Enter passphrase for /dev/sdb:
root@ubuntu14:/home/usuario#
```

Comprobación del mapeo

Podemos comprobar que se ha realizado correctamente accediendo a /dev/mapper/backup2

```
root@ubuntu14:/home/usuario# ls -l /dev/mapper/backup2
lrwxrwxrwx 1 root root 7 avril 25 11:46 /dev/mapper/backup2 -> ../dm-0
root@ubuntu14:/home/usuario#
```

Status del mapeo

Podemos observar que aparece en type LUKS1, esto podría deberse a que inicialmente he creado el mapeo utilizando LUKS1, y aunque después lo haya hecho con LUKS2, el sistema todavía está reconociendo el mapeo como creado con el formato LUKS1, porque así fue como se creó originalmente.

Por lo que he estado leyendo, esto no significa que el mapeo sea LUKS1, sino que se creó así inicialmente. En resumen, el sistema conserva la información original del tipo de encriptación al crear el mapeo, de ahí que nos aparezca LUKS1.

```
root@ubuntu14:/home/usuario# cryptsetup -v status backup2
/dev/mapper/backup2 is active.
  type:    LUKS1
  cipher:  aes-xts-plain64
  keysize: 256 bits
  device:  /dev/sdb
  offset:  4096 sectors
  size:    22217376 sectors
  mode:    read/write
Command successful.
root@ubuntu14:/home/usuario#
```

Información detallada con luksDump.

Este comando es útil para obtener una visión general de cómo está configurado un contenedor LUKS, qué tipo de cifrado se está utilizando y si está en un estado activo o no.

```
root@ubuntu14:/home/usuario# cryptsetup luksDump /dev/sdb
LUKS header information for /dev/sdb

Version:        1
Cipher name:    aes
Cipher mode:    xts-plain64
Hash spec:      sha1
Payload offset: 4096
MK bits:        256
MK digest:      c7 f1 6a 93 f3 48 50 c5 8b ba a7 c1 33 7d 76 a7 c9 ff e4 50
MK salt:        3d 1b 5e 0b 68 c6 43 2c f6 f7 68 55 78 34 73 4c
                5d 11 e2 84 86 9c 6d 14 3f d2 74 99 ad ce 99 57
MK iterations:  115000
UUID:           47bda96c-42be-4dd6-816e-f39628b2c040

Key Slot 0: ENABLED
        Iterations:             457142
        Salt:                   74 35 ef 77 d0 f7 97 c3 d7 39 1b 08 2f 4f 8b 13
                                7e 3c a4 46 01 33 61 31 9e 9d 1d ee b0 81 18 5f
        Key material offset:    8
        AF stripes:             4000
Key Slot 1: DISABLED
Key Slot 2: DISABLED
Key Slot 3: DISABLED
Key Slot 4: DISABLED
Key Slot 5: DISABLED
Key Slot 6: DISABLED
Key Slot 7: DISABLED
```

Paso 3: Formatear la partición LUKS de Linux

Llenando el disco encriptado con datos aleatorios.

Primero necesitaremos escribir ceros en el dispositivo encriptado /dev/mapper/backup2. Esto asignará datos de bloques con ceros, así garantizaremos que el mundo exterior vea esto como datos aleatorios.

Es totalmente normal que nos salga el mensaje de “no space left on device”, ya que indica que el dispositivo encriptado ha sido llenado de ceros como se pretendía.

```
root@ubuntu14:/home/usuario# dd if=/dev/zero of=/dev/mapper/backup2
dd: writing to ‘/dev/mapper/backup2’: No space left on device
22217377+0 records in
22217376+0 records out
11375296512 bytes (11 GB) copied, 2063,11 s, 5,5 MB/s
```

En caso de ser un disco mucho más grande de tamaño, se recomienda añadir ***pv -tpreb /dev/zero*** al principio del comando, ya que tardaría bastante en hacerse el comando dd, y así podríamos monitorizar el proceso. El comando entero sería el siguiente: ***pv -tpreb /dev/zero | dd of=/dev/mapper/backup2***

bs=128M. También podríamos pasarle la opción status=progress: dd ***if=/dev/zero of=/dev/mapper/backup2 status=progress.***

Creación del sistema de archivos en el dispositivo encriptado.

Vamos a ejecutar el comando mkfs.ext4 /dev/mapper/backup2 para que pueda ser utilizado como un espacio de almacenamiento con el sistema de archivos ext4.

```
root@ubuntu14:/home/usuario# mkfs.ext4 /dev/mapper/backup2
mke2fs 1.42.9 (4-Feb-2014)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
694960 inodes, 2777172 blocks
138858 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2847932416
85 block groups
32768 blocks per group, 32768 fragments per group
8176 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Montando el nuevo sistema de archivos en /backup2.

```
root@ubuntu14:/home/usuario# mkdir /backup2
root@ubuntu14:/home/usuario# mount /dev/mapper/backup2 /backup2
root@ubuntu14:/home/usuario# df -H
Filesystem           Size  Used Avail Use% Mounted on
udev                 1,1G  4,1k  1,1G   1% /dev
tmpfs                210M  889k  209M   1% /run
/dev/sda1             25G  6,5G   17G  29% /
none                 4,1k     0  4,1k   0% /sys/fs/cgroup
none                 5,3M     0  5,3M   0% /run/lock
none                 1,1G  156k  1,1G   1% /run/shm
none                 105M   46k  105M   1% /run/user
/dev/sr0              54M   54M     0 100% /media/usuario/VBox_GAs_7.0.12
/dev/mapper/backup2   12G   28M   11G   1% /backup2
root@ubuntu14:/home/usuario# cd /backup2/
root@ubuntu14:/backup2# ls
lost+found
root@ubuntu14:/backup2# ls -l
total 16
drwx------ 2 root root 16384 avril 26 01:42 lost+found
root@ubuntu14:/backup2#
```

¿Cómo desmonto y aseguro los datos?

Usaríamos el comando umount.

sd

```
root@ubuntu14:/# umount /backup2/
root@ubuntu14:/# ls
backup2  boot   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  var
bin      cdrom  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  vmlinuz
root@ubuntu14:/# ls -l backup2/
total 0
```

¿Puedo ejecutar fsck en una partición basada en LUKS / volumen LVM?

Primero desmontamos la partición /backup2/

Con el comando ***fsck -vy /dev/mapper/backup2*** ejecutamos fsck (filesystem check) en el dispositivo encriptado. Fsck es una herramienta que verifica y repara la integridad del sistema de archivos. El –vy se utiliza para reaizar una verificación detallada y mostrar todos los pasos realizados por fsck.

Con el comando mount montamos el dispositivo encriptado /dev/mapper/backup2 en el directorio /backu2. Después de realizar el chequeo de integridad del sistema de archivos con fsck, el dispositivo encriptado se vuelve a montar para que los archivos y directorios sean accesibles nuevamente.

```
root@ubuntu14:/# umount /backup2/
root@ubuntu14:/# fsck -vy /dev/mapper/backup2
fsck from util-linux 2.20.1
e2fsck 1.42.9 (4-Feb-2014)
/dev/mapper/backup2: clean, 13/694960 files, 83181/2777172 blocks
```

```
root@ubuntu14:/# mkdir /backu2
root@ubuntu14:/# mount /dev/mapper/backup2 /backu2
root@ubuntu14:/#
```

¿Cómo cambio la contraseña de LUKS para una partición encriptada?

Usaremos el cryptsetup con luksAddKey. Después pondremos la contraseña que ya existía y después la nueva.

```
root@ubuntu14:/# cryptsetup luksAddKey /dev/sdb
Enter any existing passphrase:
Enter new passphrase for key slot:
root@ubuntu14:/#
```