Davante

Las copias de seguridad

Actividad 11

Juan Carlos Calvo Verdaguer
1º ASIR - FW Act11

Título del índice

1. Herramientas seleccionadas

Debido a que el equipo que deseamos asegurar utiliza una distribución de Linux, en este documento nos ceñiremos a herramientas de Linux o multiplataforma. Además, aunque disponemos de múltiples herramientas gráficas para lograr nuestro objetivo, optaremos siempre que sea posible por herramientas de consola y, si es posible, herramientas que vienen pre-instaladas en la mayoría de distribuciones modernas. Los motivos de estas decisiones son priorizar la eficiencia y el uso de métodos universales que funcionen en cualquier sistema, es decir, que la mayoría de este documento debería funcionar igual en Debian, RedHat, SUSE, Arch, o cualquier otra distribución de Linux.

Las herramientas principales que usaremos son:

- Realizar la imagen de sistema: dd.[1]
  - Herramienta básica de Unix de creación y restauración de copias de seguridad; presente en todas las distribuciones de Linux como parte de las “Coreutils”.[2]
- Automatizar la creación de la imagen: systemd.[3]
  - Gestor de servicios de sistema (demonios) estándar.
- Restaurar la imagen: dd o Rescuezilla.[4]
  - Rescuezilla permite restaurar un sistema incluso si está tan roto que no puede ejecutar programas de restauración.

2. Creación y automatización de copias de seguridad

Esta parte del documento es prácticamente idéntica a la actividad 11 de ISO,[5] con sólo algunas modificaciones, ya que la idea es la misma. Específicamente, repetiremos la tercera parte del punto **1.2.1** y el punto **1.2.2** completo.

2.1. Dd

El comando dd es una herramienta básica de linux que permite clonar tablas de particiones completas de un dispositivo de almacenamiento a otro. Su uso, aunque sencillo, requiere prestar mucha atención a qué dispositivos seleccionamos y en qué orden van el el comando, ya que podríamos sobreescribir el disco equivocado y perder información en lugar de asegurarla.

Para crear una imagen de sistema en un archivo .img, deberíamos de utilizar el siguiente comando:

```
dd if=/dev/vda of=/run/media/usuario/backups/backup-vda.img status=progress
```

Esto nos mostrará el progreso de la copia de seguridad, que nos devolverá a la línea de comandos cuando termine.

2.2. Configuración de temporizadores

Hasta ahora, los métodos que hemos visto para crear copias de seguridad en sistemas operativos basados en Linux han sido métodos manuales e imperativos, es decir, que sólo hacemos una copia cuando un humano accede a la terminal de comandos y le dice al sistema operativo que haga la copia de seguridad. Pese a que este es un sistema válido que puede servir en algunos escenarios, no es tan fiable a la práctica.

Es por esto que existen también formas de automatizar procesos en Linux, que permiten programar tareas para que se ejecuten sin la necesidad de intervención humana. Esto nos abre las puertas a sistemas mucho más consistentes y fiables de almacenamiento de datos y copias de seguridad. Los dos métodos principales para crear estos servicios de temporizadores (también conocidos como demonios o *daemons*) son cron y *timers* de systemd; que se pueden usar de forma conjunta en caso de ser necesario.

En este documento utilizaré temporizadores de systemd debido a que es el sistema más soportado en sistemas operativos actuales y presenta grandes ventajas como una mayor eficiencia y opciones más granulares sobre permisos y entorno de ejecución. Para procesos sencillos, da lo mismo, ya que al final ambos van a ejecutar un *script* en un momento concreto.

Temporizadores de systemd

Lo primero que haremos será crear un *script* en bash que lleve a cabo la copia de seguridad en sí. Podríamos simplemente ejecutar el comando, pero entonces cada copia sobreescribirá la anterior, por lo que vamos a usar un sencillo comando que le de un nombre distinto a cada copia que indique la fecha en la que se creó. De esta forma podremos guardar varias copias a la vez y saber exactamente cuándo se tomó.

Contenido de `/usr/local/bin/respaldo-vda.sh`:

```
#!/bin/bash
# Crea un directorio en caso de no existir
mkdir -p /backups
# Monta el disco VDB en /backups
sudo mount -t ext4 /dev/vdb /backups
# Tomamos la fecha del sistema
```

```
DATE=$(date +%F)
# Ejecutamos el comando 'dd', creando una imagen del disco vda.
sudo dd if=/dev/vda of=/backups/respaldo-vda-${DATE}.img
```

Ahora que contamos con este *script*, podemos invocarlo con el temporizador.

La forma en la que funcionan los temporizadores en systemd es mediante la creación de un servicio, que es el que detalla en profundidad la tarea que queremos ejecutar, junto con un "timer" con el mismo nombre en el que se establece la temporalización de este servicio. Esto es necesario ya que, por defecto, los servicios se ejecutan de forma constante y en segundo plano; un comportamiento que no nos interesa para este tipo de tareas de mantenimiento.

Contenido de `/etc/systemd/system/respaldo-vda.service`:

```
[Unit]
Description=Crea una copia de respaldo completa.

[Service]
# Indica que se ejecute una única vez
Type=oneshot
# Indica qué tiene que ejecutar, la dirección del binario o script
ExecStart=/usr/local/bin/respaldo-vda.sh

[Install]
# Establece una prioridad de ejecución
WantedBy=timers.target
```

Contenido de `/etc/systemd/system/respaldo-vda.timer`:

```
[Unit]
Description=Indica cuándo se ejecuta el servicio con el mismo nombre.

[Timer]
# Especifica un día de la semana y una hora. Esto hará que el servicio se
ejecute cada viernes a las ocho de la tarde.
OnCalendar=fri 20:00:00
Persistent=false
```

```
[Install]
WantedBy=timers.target
```

Con los archivos de configuración creados y copiados en su directorio correspondiente, podemos proceder a habilitar el script y servicio:

Para hacer el script ejecutable, modificamos sus permisos con chmod:

```
sudo chmod +x /usr/local/bin/respaldo-vda.sh
```

Por último, habilitamos e iniciamos el temporizador (no el servicio, que queremos dormido) que acabamos de crear:

```
sudo systemctl enable respaldo-vda.timer
```

```
sudo systemctl start respaldo-vda.timer
```

Al ejecutarse el servicio, activado de forma manual para esta prueba, podemos encontrar el archivo “respaldo-vda-2025-12-15.img” dentro del directorio `/backups`. Nuestro sistema de copias de seguridad automatizadas está terminado.

```
usuario@nyarch-asir:~$ ls -l /backups
total 39287704
-rw-r-r- 1 root root 40230502400 dic 15 13:56 respaldo-vda-2025-12-15.img
```

3. El peor de los escenarios

Existen incontables problemas que pueden provocar la pérdida de datos en cualquier clase de sistema operativo, desde accidentes físicos hasta ataques de ransomware. Para simplificar la práctica, voy a limitarme a hacer el clásico comando de destrucción de un sistema Linux:

```
sudo rm -fr / --no-preserve-root
```

Este comando elimina la raíz del sistema, por lo que perderemos todos los datos. Como dato curioso, antiguamente no existía la flag `--no-preserve-root`, por lo que cualquier error en un script podía eliminar accidentalmente el sistema entero. Esto llevó a una situación en 2015

donde Steam, una tienda y lanzador de videojuegos, eliminó accidentalmente los sistemas de varios usuarios con un script mal formado.[6]

El resultado es ver cómo el sistema elimina los archivos hasta colapsar.

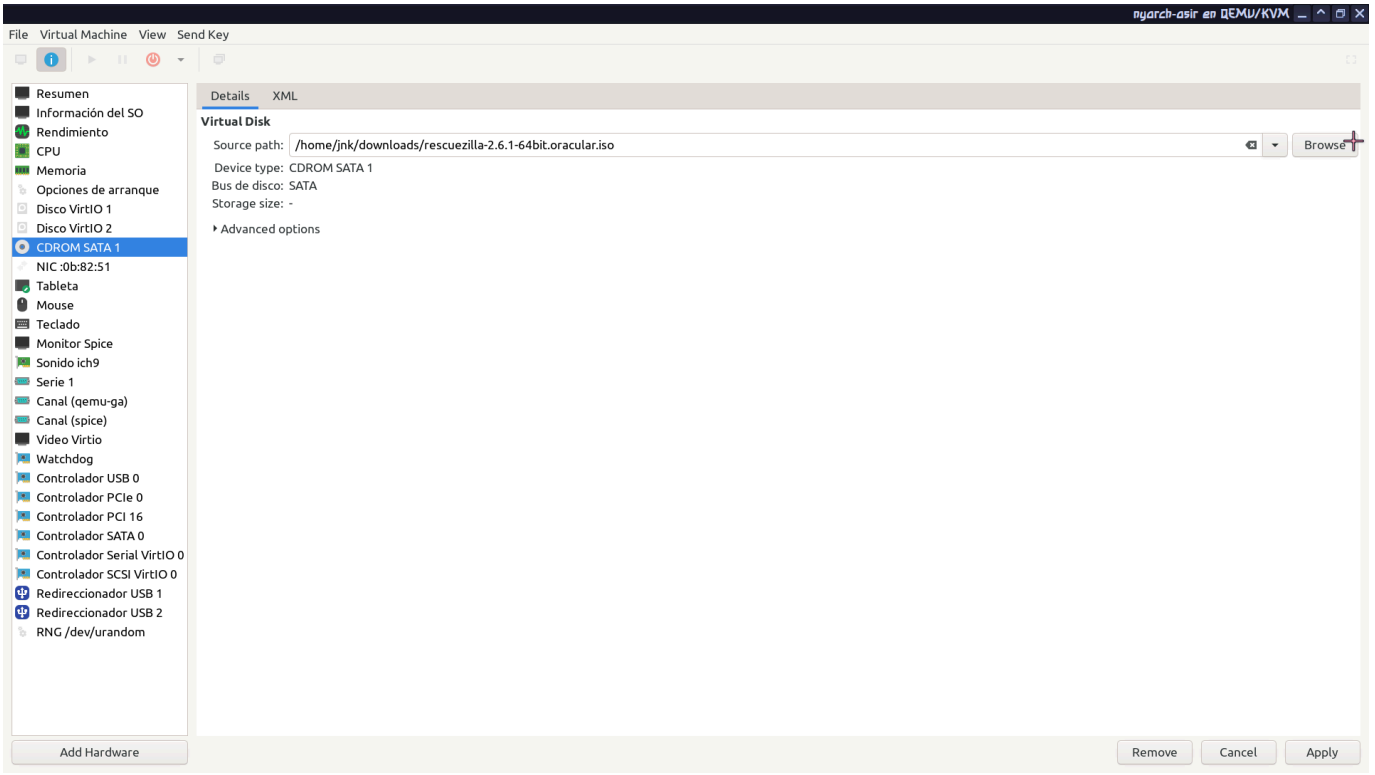
4. Restauración y conclusiones

Ahora que tenemos un sistema roto, no debería iniciar, ya que lo único que no borramos antes fueron los archivos temporales y la RAM. Eso nos ha dejado en el peor de los casos, por lo que no podemos simplemente usar la utilidad `dd` a la inversa para restaurar la imagen que creamos anteriormente. En este tipo de situaciones necesitamos usar una herramienta de rescate externa, que puede ser un “live cd” o una herramienta especializada.

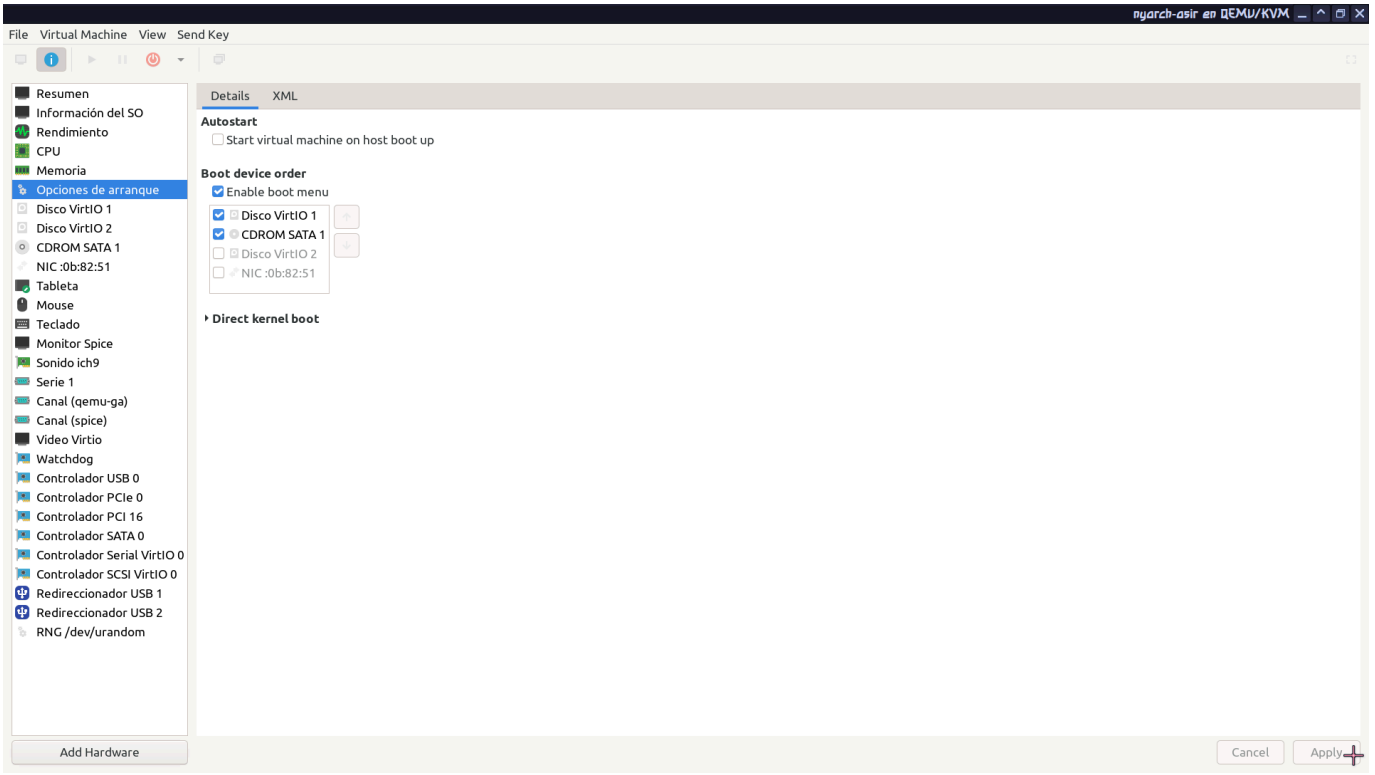
En este documento, mostraremos cómo utilizar Rescuezilla,[2] una herramienta de rescate especializada.

4.1. Configuración de la máquina virtual

Para empezar, forzaremos el cierre de la máquina virtual que acabamos de romper. Luego tenemos que asignar el archivo .iso de Rescuezilla. Para ello, nos vamos al dispositivo virtual “CDROM SATA 1”, pulsamos sobre “Browse”, y seleccionamos el archivo deseado.



Ahora que la MV tiene un CD virtual “insertado”, nos vamos al menú de opciones de arranque de la máquina y activamos ambos “Enable boot menu” y “CDROM SATA 1”. Esto nos permitirá acceder al menú de arranque de la MV y seleccionar nuestro sistema de recuperación, respectivamente. No olvides aplicar todos los cambios.



Con esto hecho, podemos arrancar la máquina.

4.2. Restauración con Rescuezilla

Pulsamos rápidamente la tecla escape (ESC) para entrar al menú de arranque y seleccionamos la opción que diga “CD”. En nuestro caso es la opción 2 (que incluso reconoce el nombre del sistema), por lo que pulsamos la tecla para iniciar Rescuezilla.

Una vez dentro seleccionamos el idioma español y continuamos con enter. En unos segundos, veremos el sistema iniciado.

Entramos en restaurar y esperamos a que escanee los discos disponibles. Seleccionamos el disco en el que se encuentra la imagen que creamos en el punto 2.1 y avanzamos.

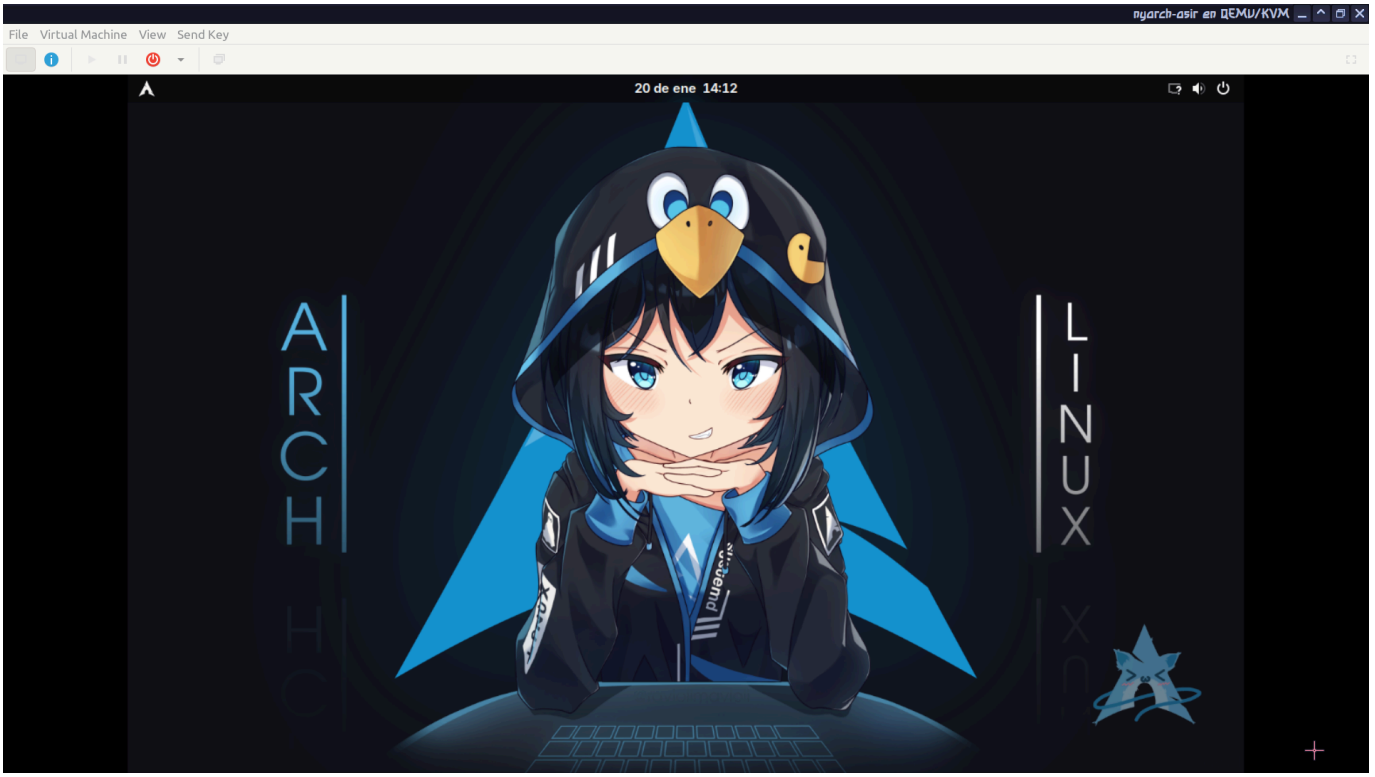
En la siguiente pantalla seleccionamos la imagen “backup-vda.img”.

A continuación seleccionamos el disco de destino, es decir, el disco que queremos reparar. En este caso, seleccionamos el primer disco; de 32GB en formato ext4, ya que se trata del disco principal de la MV.

Continuamos la siguiente pantalla, que tiene seleccionadas todas las particiones por defecto, y continuamos a la última pantalla de confirmación. Recuerda **siempre** revisar varias veces este tipo de procesos, ya que cualquier despiste o fallo tonto puede arruinar un equipo o datos importantes.

Esperamos a que finalice el proceso de recuperación.

Una vez finalice el proceso, reiniciamos la máquina y estará igual que cuando creamos la imagen de sistema.


nyarch-asir en QEMU/KVM
File Virtual Machine View Send Key
20 de ene 14:12
ARCH
LINUX

Bibliografía

[1] Dd:
https://es.wikipedia.org/wiki/Dd_(Unix)

[2] Coreutils:
https://www.gnu.org/software/coreutils/coreutils.html

[3] Systemd:
https://es.wikipedia.org/wiki/Systemd

[4] Rescuezilla:
https://rescuezilla.com

[5] Actividad 11 de ISO:
https://docs.google.com/document/d/10CHl3PK7Qu43DwGysR_TgP4AreOxhn7xH8hld4_hEy4/edit?usp=sharing

[6]Accidente de Steam:
https://hackaday.com/2024/01/20/how-a-steam-bug-once-deleted-all-of-someones-user-data/