

# Protección física en Linux

Bienvenidos a esta nueva sesión en la que vamos a trabajar la protección física de los sistemas, pensando ya en qué cosas puede ocurrir en un sistema Linux cuando no tenemos una buena protección física.

Lo que vamos a ver en esta sesión es un poco la parte de protección física o seguridad física, lo que necesitamos proteger cuando un usuario tiene acceso a un sistema.

Hablaremos de la BIOS, hablaremos de sistemas UEFI, de la evolución de las BIOS, hablaremos del SecureBoot y hablaremos de GRUB.

## Protección física

### Physical protect

- An unprotected system physically can be vulnerable to a potential attacker.
- Offline access will allow manipulation of the file system.
- To Modificate the boot system, privileged access can be gained



Singularity Hackers

0xWORD



My Public  
Inbox

Cuando hablamos de la protección física o seguridad física, hablamos de que los sistemas que no estén protegidos físicamente pueden ser vulnerables a que un usuario tenga acceso físicamente a ese equipo.

Podríamos hacer, o un usuario podría hacer muchas cosas cuando tiene acceso físico a un sistema, por ejemplo, podría arrancar un sistema operativo en memoria, si tiene acceso a la BIOS, por ejemplo, para cambiar el orden de arranque de los dispositivos, en vez de arrancar desde el dispositivo de disco duro, pues podrá arrancar desde un USB o desde una unidad óptica en el cual pueda cargar un sistema operativo y poder acceder al sistema de archivos que está en esa máquina.

Al final tenemos que entender que este tipo de ataques que se llaman ataques offline, porque el sistema operativo residente en la máquina, el sistema operativo que está instalado en la máquina, no arranca al iniciar la máquina, al iniciar el ordenador, sino que iniciamos con otro sistema operativo que cargamos en memoria.

Si no tenemos cifrado, en este caso la partición disco duro o el propio disco duro donde se encuentra el sistema operativo, los datos del sistema, pues también pueden tener acceso a ello.

El acceso tipo offline va a permitir manipular este tipo de información.

Como he comentado, si se modifica el sistema de arranque, también puede ocurrir en sistemas Linux que tengan GRUB por ejemplo, como gestor de arranque, pues también esa modificación, si el gestor de arranque no está protegido, también podría provocar que el atacante pudiera arrancar en modo privilegiado.

## Physical protect

- Protecting access to equipment physically:

- BIOS
- GRUB - Boot managers
- File system
- Partition encryption
- File encryption
- Physical security

Proteger el acceso a los equipos de forma física es algo importante. Entonces tenemos que tener en cuenta varias cosas.

El primer punto es la BIOS o el sistema UEFI que estemos utilizando en nuestro equipo, debemos tenerlo lógicamente con contraseña, debemos tenerlo con credenciales para poder editar, en este caso el estos sistemas.

Deberíamos también tener un orden de arranque que no permita arrancar un ordenador, poner un USB o poner un DVD, una unidad óptica y arrancar directamente desde ahí, sino que el orden de arranque de los sistemas debería empezar por el disco duro y luego el resto de unidades. De

forma que por lo que sea, intentamos modificar ese orden de arranque, necesitamos la contraseña de la BIOS o del sistema UEFI para poder realizar la tarea.

Además, segundo punto, hablamos del GRUB. Bueno, pues el GRUB o gestor de arranque necesitamos protegerlo. Por defecto GRUB es editable, por defecto el GRUB no tiene ningún tipo de protección para arrancar los sistemas, pero lo más crítico es que el GRUB pueda ser editable, es decir, podemos editar y arrancar una shell como root y poder tener acceso completo al sistema.

Entonces tenemos que proteger el GRUB. Es una cosa fundamental que debemos tener en cuenta.

El tercer punto habla del sistema de archivos (File System). Ya hemos comentado en la slide anterior que el sistema de archivos es importante cifrarlo a nivel de partición y a nivel de archivo. Esto ya lo hemos trabajado en otros módulos, pero volvemos a hacer hincapié, el cifrado de partición es importante porque protege un gran conjunto de datos, pero el cifrado a nivel de archivo o carpeta también. Si queremos darle una segunda capa de protección a una información que es importante o sensible para nosotros, podemos dárselo de forma que cuando arranquemos el sistema operativo, cuando arranquemos la partición de datos, podamos descifrarlo y luego pueda haber algún tipo de información aún más sensible que también esté cifrada a nivel de en el caso de Linux, por ejemplo, con un GPG, a nivel de clave simétrica o clave o incluso clave pública o clave simétrica y poder disponer de la información protegida en un segundo nivel, de forma que si me rompen ese primer nivel de cifrado, que es el cifrado de partición, todavía el atacante tendría que romper un segundo nivel de descifrado.

En el cuarto punto hablamos de seguridad física (physical security).

Aquí hablamos, como hemos comentado antes, de la importancia de, oye, no puedan manipular el hardware. Imaginaros que tenemos un equipo de sobremesa y cualquiera puede abrir ese equipo, modificar los dispositivos que hay dentro, las unidades, insertar un pendrive o insertar otro tipo de dispositivo, otro tipo de unidad, por ejemplo unidad óptica, cambiar el orden de arranque. Si podemos manipular el hardware también tenemos un problema y deberíamos tener bloqueado o con un candado protegido la manipulación del propio hardware que hay en un equipo.

## **BIOS y UEFI**

## BIOS and UEFI

- BIOS
  - It is a legacy firmware that has been widely used in computers for decades
  - Provides a basic interface for configuring and controlling computer hardware, such as system settings and boot sequence
  - It has limitations in terms of support for modern devices, storage capacity for boot data, and security



## BIOS and UEFI

- UEFI
  - It is a newer technology designed to replace BIOS
  - Offers a more advanced and extensible interface for hardware initialization and operating system boot
  - Supports modern features such as large hard drives (over 2 TB), Secure Boot, support for GUID partitions (GPT), and a more user-friendly interface



Bien, vamos a hablar de la BIOS, tras haber hecho esa primera definición, bueno, aquí vemos que la BIOS al final es un firmware legacy que llamamos, es un firmware antiguo que ha sido ampliamente utilizado en los ordenadores durante muchísimos años, nos proporciona una

interfaz para poder gestionar y controlar ese hardware que tenemos en el equipo y esa configuración del sistema y de arranque.

Y bueno, tenía una serie de limitaciones lógicamente en capacidad de almacenamiento y también en limitaciones en temas de seguridad.

En el caso de los sistemas UEFI, como vamos a ver, esta tecnología es una tecnología más reciente que la propia BIOS, es una tecnología que es una evolución, ofrece una interfaz mucho más avanzada, más extensible, que permite inicializar el hardware, permite gestionar el hardware, gestionar el arranque de los sistemas operativos, gestionar la configuración de los dispositivos también, pero hace desde un punto de vista mucho más seguro y mucho más evolucionado.

Como ventajas soporta características modernas como por ejemplo discos duros de más de 2 Tb, arranque seguro (Secure Boot) y aquí es donde decíamos que mejoramos la seguridad respecto a la BIOS. Soporta también particiones de tipo GVD, es decir, las típicas GPT y bueno pues es un uso mucho más friendly.

## SecureBoot

### SecureBoot

- It is a security feature that ensures only trusted and digitally signed software is loaded during the boot process of an operating system. It helps prevent the loading of malicious or unauthorized software during system startup



Picture source: Google DeepMind. Free use. <https://www.pexels.com/es-es/foto/abstracto-tecnologia-investigacion-digital-17485657/>



Singularity Hackers

0xWORD



My Public  
Inbox

Vamos a hablar también del Secure Boot. Es una característica de seguridad que tenemos en los equipos actuales, que está implementado también gracias a los sistemas UEFI y que nos va a permitir ejecutar software confiable. ¿Cómo funciona esto?

Al final este software confiable, vamos a fiarnos de él, está firmado digitalmente, entonces cuando vamos a ejecutar, arrancar el sistema operativo, se hace una serie de comprobaciones para evitar que el software malicioso, algún tipo de componente malicioso se pueda arrancar en el propio arranque del sistema operativo. Entonces, ¿Cómo funciona Secure Boot?

Secure Boot verifica digitalmente la integridad y la firma de los componentes de software que se quieren cargar en el arranque y si alguno de estos no se puede validar, pues entonces se evitará el arranque del sistema operativo.

Esto puede ocurrir con sistemas operativos que no tengan esa integridad o no tengan esa firma o nuestro sistema Secure Boot no conozca esa firma, entonces no lo de por bueno e impida la arranque.

Esto nos ayuda a intentar evitar los bootkits, por ejemplo, o código que se intenta ejecutar en el arranque del sistema operativo para poder aprovecharse y tomar beneficio del equipo.

## GRUB

### GRUB

- GRUB is a multiple boot loader developed by GNU.
- It is mainly used in GNU/Linux operating systems.
- It allows having one or several operating systems and booting them up easily

Bien, llegamos a la parte de GRUB. Es un cargador, un gestor de arranque múltiple, es un gestor open source de código abierto y utilizado principalmente en sistemas Unix o Linux.

Lo que hace GRUB al final es el encargado de que el del sistema operativo se ejecute, se arranque y todos los componentes necesarios para arrancar el sistema operativo. También como he dicho se utiliza principalmente en sistemas operativos GNU Linux y es una primera pieza de protección que tenemos que proteger porque al final es un cargador que toca mucha

parte sensible porque al final está ejecutando los sistemas operativos que se pueden cargar en el sistema equipo y lógicamente habrá que protegerlo para que un atacante no pueda obtener beneficios en ese arranque.

## GRUB

- By modifying the boot process, the system can be initialized with privileged rights:
  - `init=/bin/bash`
  - Mount the partition in write mode.
  - `mount -o remount,rw /dev/sdaX`
- The offline attack allows modification or access to any file



Singularity Hackers

0xWORD



My Public  
Inbox

De forma teórica, aunque luego lo vamos a ver de forma práctica, vamos a ver un ejemplo de cómo mediante la modificación del arranque en GRUB, como he dicho anteriormente, GRUB es editable, cuando vemos la pantalla de group para mostrar una imagen y vemos la pantalla de group con la letra e se puede editar.

Entonces mediante la modificación de ese arranque se podría inicializar el sistema indicando que se quiere arrancar un terminal de bash en vez de arrancar el sistema operativo normal.

Ese terminal de bash lógicamente se va a ejecutar como root y además se puede hacer un remontaje, un montado de la partición para tener incluso permiso de lectura escritura. Esto es un ataque offline porque al final el sistema no está arrancado, lo que está en ejecución es el propio gestor de arranque del sistema operativo.

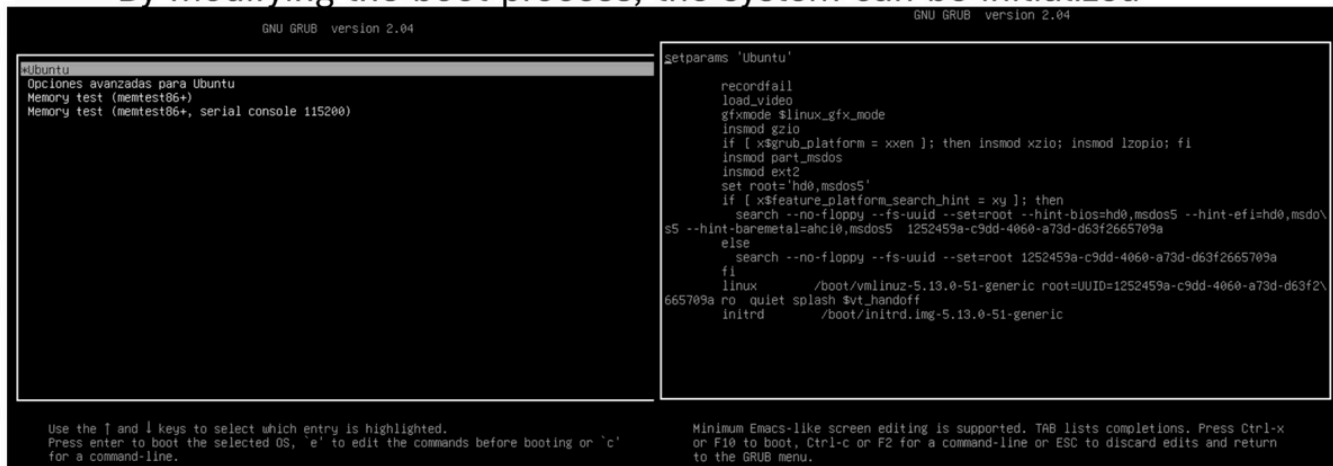
Como vemos este tipo de ataques offline nos va a permitir modificar o acceder al fichero shadow u otros ficheros.

## Ataque offline desde GRUB



# GRUB

- By modifying the boot process, the system can be initialized



The screenshot shows the GRUB version 2.04 interface. On the left, the 'Ubuntu' menu is selected, showing options: 'Opciones avanzadas para Ubuntu', 'Memory test (memtest86+)', and 'Memory test (memtest86+, serial console 115200)'. On the right, the 'setparams 'Ubuntu'' script is displayed, showing the configuration for the Ubuntu boot process, including the search for the root filesystem and the execution of the kernel and initrd. At the bottom, instructions for using the arrow keys to navigate and the 'e' key to edit are provided.

```
setparams 'Ubuntu'
recordfail
load_video
gfxmode $linux_gfx_mode
insmod gzio
if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
insmod part_msdos
insmod ext2
set root='hd0,msdos5'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos5 --hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5 1252459a-c9dd-4060-a73d-d63f2665709a
else
  search --no-floppy --fs-uuid --set=root 1252459a-c9dd-4060-a73d-d63f2665709a
fi
linux /boot/vmlinuz-5.13.0-51-generic root=UUID=1252459a-c9dd-4060-a73d-d63f2665709a ro quiet splash $vt_handoff
initrd /boot/initrd.img-5.13.0-51-generic
```



Entonces fijemonos, en la parte izquierda vemos el GRUB, si le pulsamos en la tecla E entramos en el modo edición de GRUB, en la parte de la derecha vemos el script que se va a ejecutar y ahí sí que veríamos hacia el final podemos ver cómo encontramos la línea que pone "ro quiet splash \$vt\_handoff" por ahí nosotros podemos sustituir eso porque estamos viendo además que lo que está cargando es /boot y la imagen, en este caso ubuntu y podemos sustituir eso para que en vez de arrancar esto nos arranque en lectura escritura pues un terminal de bash.

```
linux /boot/vmlinuz-3.0.0-12-generic root=UUID=2655490d-2f16-4b96-a4\
bc-d0c11d6eb17f ro quiet splash rw init=/bin/bash_
initrd /boot/initrd.img-3.0.0-12-generic
```

Vemos la línea que estamos sustituyendo, cuando nosotros le damos al f10 después de un proceso de arranque no arranca el sistema operativo, sino que lo que arranca es un terminal y aquí vamos a poder encontrarnos terminal y el prompt diciéndonos que somos root.

```
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@(none):/# _
```

A partir de aquí tenemos acceso completo al sistema, depende si estamos en modo lectura escritura, bueno, podríamos modificar cosas o no, el atacante podría modificar cosas o no, pero lo que sí va a poder es leer y va a poder extraer información del sistema como root.



## GRUB

- To protect:
- Set a password for GRUB2: This can be achieved by configuring a password in the grub.cfg file or using the grub2-mkpasswd-pbkdf2 utility
- Restrict access to the GRUB2 configuration file: Just like with GRUB, it's important to ensure that only authorized users have permission to modify the GRUB2 configuration file
- Use SecureBoot



## ¿Cómo nos protegemos de ésto?

Vamos a hablar de forma teórica, también haremos una sesión donde veremos esto de forma práctica.

Y bueno, para proteger GRUB en Linux se pueden tomar medidas similares a las mencionadas que hemos ido comentando.

Lo primero hay que establecer una contraseña maestra o un superusuario que tiene una serie de roles, meterle un superusuario o varios superusuarios que sin esa contraseña, sin ese usuario y contraseña no se pueda editar el grupo, eso es algo fundamental.

Además hay que restringir acceso al archivo de configuración o donde se almacenen los datos de los usuarios.

Luego también tenemos la posibilidad de crear usuarios para que puedan delegar o podamos delegar el arranque de ciertos sistemas, es decir, podemos proteger también a nivel de GRUB qué sistema operativo se pueda ejecutar y qué usuario puede ejecutarlo, no son usuarios del sistema, son usuarios del GRUB, son cosas diferentes.

Y luego por supuesto utilizar el SecureBoot para intentar evitar la carga de componentes de software no firmados o posiblemente maliciosos. Pero el SecureBoot no es una protección tanto para el GRUB, sino al final para el GRUB lo que nos interesa más es roles de superusuarios para indicar qué usuarios pueden editar el GRUB en caso de necesidad y usuarios rasos, usuarios normales, para ver qué usuarios pueden arrancar los sistemas operativos que está gestionando

GRUB en el caso de ser necesario, porque puede ser que no lo necesitemos, entonces no lo llevaremos a cabo así. Pero sobre todo lo que sí seguro es que necesitaremos proteger la parte de edición del GRUB.

## Conclusions

- Physical protect
- BIOS and UEFI
- SecureBoot
- GRUB

Como conclusiones, tenemos la protección física, tenemos, hemos visto la parte de sistemas de la BIOS, lo que es la BIOS, lo que es un sistema UEFI, qué cosas nos interesan para esta parte de Linux, también lo hemos estado viendo.

Hemos visto también el concepto de SecureBoot, en qué nos ayuda, qué nos protege cuando lo veamos en un sistema operativo Linux, pues lógicamente ya sabemos de qué nos están hablando. Cuando veamos un sistema UEFI, pues ya sabemos de qué están hablando.

Hemos visto también el GRUB como efecto de arranque y su potencial configuración por defecto nos trae un potencial debilidad y en el caso de protegerlo veremos cómo estamos poniendo capas de manera que ya no pueden aprovecharse ese acceso físico, esa manipulación del groove para sacar un beneficio a lo que es un atacante.