[FORT] Tema 4: Securización de Aplicaciones

Identificar y eliminar aplicaciones no utilizadas

Hay 2 tipos de aplicaciones:

- Las que tienen agujeros de seguridad
- Las que las tienen pero aún no se han descubierto

Cuando se instala una distribución de linux hay muchas aplicaciones que no son necesarias de las cuales se puede prescindir. Se pueden poner límites a las configuraciones en /etc/security/limits.conf. Estos límites son por sesión, estos límites tienen uno hard y uno soft. El límite soft se puede superar, siempre y cuando no sobrepase el límite hard. Para limitar una aplicación podemos usar:

- cpulimit: Limita el consumo de CPU de una aplicación, pero es demasiado rudimentario
- prlimit: Permite poner límites a los gastos de recursos de un proceso.

Limitar recursos de aplicaciones con cgroups

Los cgroups son jerárquicos. Para crear un cgroup hacemos lo siguiente

```
cd /sys/fs/cgroup/ #Ruta de los cgroups
mkdir prueba #Creamos nuevo cgroup
```

Los ficheros del cgroup son como los ficheros de proc. Ahora el directorio /prueba debería estar poblado por numerosos archivos. Para añadir un proceso al cgroup hacemos lo siguiente:

```
echo 2007 > cgroup.procs #Metemos el proceso 2007 en cgroup.procs
```

Para limitar el consumo de cpu hacemos:

```
echo 10000 1000000 > cpu.max #Por cada 100000 de CPU se asigna 10000 de CPU
al programa.
```

Para limitar el consumo de memoria:

```
echo 5000000 > memory.high #Se limita a 500000 de memoria el uso del
programa
```

se pueden parar todos los procesos del cegroup con:

```
echo 1 > csgroup.freeze
```

Para liberar los programas se usa:

```
 echo 0 > csgroup.freeze
```

Ejecución en jaulas chroot

El programa que se ejecuta en estas jaulas no puede subir del directorio en el que se ejecuta. Antiguamente se usaba para testear software y para servidores FTP. Se suele usar cuando se arranca un medio de instalación. Para tener un entorno ChRoot funcional se hace lo siguiente:

Entorno de virtualización

La creación de containers es muy simple, utilizamos lxc: <Code> lxc-create </code>

Se pueden ver los contenedores que se pueden crear en /usr/share/

```
lxc-create -t alpine -n NOMBRE_CONTENEDOR #En este caso alpine sería el tipo de contenedor
```

Para arrancar el container usamos el siguiente comando:

```
lxc-start -F -n NOMBRE_CONTENEDOR #el -F indica que es en primer plano y el -n el nombre del contenedor.
```

Para parar el contanier se usa:

```
lxc-stop -n NOMBRE_CONTENEDOR
```

Los containers tienen usuarios predefinidos que se suelen indicar al crear el container. Podemos ejecutar algo en el container con:

```
lxc-attach -n NOMBRE_CONTENEDOR /bin/sh #Por ejemplo, ejecutamos un shell en el container
```

Una vez conectados así al contanier le podemos poner una contraseña con el comando “passwd root” y podemos crear un usuario nuevo en esta con “useradd -m NOMBRE”. Podemos ver los container arrancados y sus ips con:

```
lxc-ls -f
```

Para entrar en un container que no se ha iniciado con -F podemos usar SSH contra su IP. lxc crea una interface llamada bridge 0 para conectar los containers mediante NAT como si fuera virtualbox. para configurar la red de un container vamos a la dirección:

```
cd /var/lib/lxc
```

Aquí hay una carpeta por container, para configurar uno vamos al que queramos y modificamos el archivo config. Dentro de este se pueden ajustar más parámetros, como el autoarranque o la memoria entre otros. Mediante NFTables podemos comunicar el container con el exterior para que pueda prestar servicios. Esto nos permite arrancar aplicaciones de forma aislada. Dentro de /var/lib/lxc/NOMBRE_CONTAINER/rootfs podemos encontrar los archivos usandos dentro del container.

Estos aparecen como pertenecientes al usuario que creó el container, a pesar de ello la máquina los puede usar, esto se debe a que comparte los identificadores con la máquina HOST

Mandatory Access Control

Hay 2 tipos:

SELinux

Usado por Fedora. Permisos por archivos, todo tiene una etiqueta (Archivos, procesos, etc...) que dice que puede acceder a que. Solo se admiten los acceso permitidos por las etiquetas. Para convertir una máquina en SELinux primero hay que instalar los siguientes paquetes:

```
apt install selinux-basics selinux-utils selinux-policy-defualt auditd
```

Tras es activamos selinux con el siguiente comando:

```
selinux activate
```

esto crea un archivo /.autorelabel para etiquetar los ficheros no etiquetados. Selinux necesita EXT4. Si se usa Selinux no se puede compartir el directorio home entre dos distros que no lo usen. Selinux tiene 2 modos:

- Modo permisivo: selinux simplemente manda warnings al log, pero no bloquea el acceso.
- Modo enforce: Bloquea accesos no autorizados, lo malo es que solo avisa la primera vez que se intenta realizar un acceso no autorizado.

También añade al grub en /boot/grub/grub.cfg el modo security = selinux. En /etc/selinux/config podemos cambiar el modo de selinux de permisive a enforce.

Para ver los avisos generados por selinux podemos usar el siguiente comando:

```
audit2why -a
```

APParmor

Usado por debuan. Permisos por aplicación indicando donde puede acceder y donde no. Para habilitar un programa con apparmor usamos el comando:

```
apparmor_parser /etc/apparmor.d/usr.bin.programa
```

Por defecto el programa se pondrá en modo enforce. Para saber el estado de las aplicaciones en apparmor usamos

```
aa-status
```

para que al usar un programa simplemente mande un warning usamos complain:

```
aa-complain /usr/bin/programa
```