Xestión da Autorización

ACL e Permisos, Control Obrigatorio e Discrecional
Atributos e Limitación de recursos

Autorización

A **autorización** e o sistema polo que se decide que accións pode e non pode levar a cabo un usuario. Se pode xestionar a varios niveis:

- Sistemas de Control Discrecional ou ***DAC:*** Atributos de arquivos (lsattr, chattr), Permisos (chmod,chown), ACL (getfacl, setfacl),
- Sistemas de Control Obrigatorio ou **MAC:** MIC, AppArmor, SELinux...
- Limites ao uso de recursos do sistema (ulimit): CPU, Memoria, Disco... etc.

Control de Acceso Discrecional (DAC)

Atributos, Permisos e ACL – sudo / runas

O ***control de acceso discrecional*** se refire a que a autorización de acceso aos recurso é unha decisión discrecional do administrador, que a establece mediante determinadas ferramentas:

Atributos de arquivos: Mediante atributos no sistema de arquivos é posible limitar ou indicar determinadas características dos ficheiros, como inmutabilidad, ou si son visibles ou ocultos. Estes atributos dependen do sistema de arquivos utilizado:

- **Linux**: Se xestionan mediante os comandos **lsattr** e **chattr**. Con + se activa un atributo e con – se elimina. Os atributos poden ser os seguintes:

a: append only,**c**: compressed, **d**: no dump,**e**: extent format,**i**: immutable,**j**: data journalling, **s**: secure deletion,**t**: no tail-merging,**u**: undeletable,**A**: no atime updates,**C**: no copy on write,**D**: synchronous directory updates,**S**: synchronous updates,**T**: top of directory hierarchy

- **Windows**: Se xestionan co comando **attr**. Con + se activa un atributo e con – se elimina. Os atributos poden ser os seguintes:

R Atributo de sólo lectura, **A** Atributo de archivo, **S** Atributo de sistema, **H** Atributo de archivo oculto.

Un control máis eficiente se pode facer mediante os **permisos**. Os permisos nos sistemas Windows se xestionan mediante listas de control de acceso (ACL), mentres que Linux asocia a cada arquivo un número binario de 10 díxitos - --- --- --- onde se especifica si é un arquivo ou directorio, e os permisos de lectura (r), escritura (w) ou execución (x) para o propietario do arquivo, o grupo ao que pertence o arquivo e para o resto de persoas respectivamente. Estes permisos se xestionan co comando **chmod**

Tanto Windows como Linux poden xestionar os permisos de xeito moi detallado mediante listas de control de accceso. En Windows, esas ACL son moi detalladas permitindo un gran control da autorización e se xestionan co comando **cacls**, mentres que en Linux son menos versátiles e se xestionan mediante os comandos ***setfacl*** e ***getfacl***. En Linux as acl postas sobre cada obxecto do disco se almacenan nos atributos estendidos dos arquivos.

Un proceso de usuario estará autorizado a acceder a un obxecto dependendo dos permisos e ACL que se establezan. ***Os procesos de usuario se executan co privilexio do usuario que os lanza***. Existen dúas formas de alterar esto en Linux:

1) Os bits SUID e GUID: Estes bits permiten executar o proceso cos permisos do seu propietario ou grupo respectivamente.
2) O uso de **sudoers**: Mediante o comando **sudo** é posible executar un conxunto de comandos cunhas condicións definidas no ficheiro */etc/*sudoers

En Windows é posible facer uso do comando ***runas*** de xeito similar a *sudo*, pero moito menos versátil.

Control de Acceso Discrecional (DAC)
Limitación de uso de recursos

Outro tipo de autorización discrecional é a limitación de uso de diferentes recursos en función da identidade do usuario propietario do proceso. Podemos distinguir dous tipos de recursos diferenciados:

- **Uso de espazo en disco:** Se xestiona mediante cuotas de disco que permiten definir o uso máximo de disco para cada usuario en cada dispositivo (espazo, número de arquivos...). En **Linux** se poden utilizar os comandos ***setquota***, ***edquota***, ***quotactl***, ***quotaon***, ***quotacheck*** e ***repquota***. En **Windows** podemos facelo nas propiedades do dispositivo.
- **Recursos do sistema:** como consumo de memoria, uso de cpu, uso da rede, prioridade máxima dos procesos... En Linux se poden xestionar mediante as utilidades **sysctl** e a **ulimit**, e os ficheiros ***/etc/sysctl.conf*** e ***/etc/security/limits.conf***. Mediante sysctl e o ficheiro sysctl.conf controlamos o usode recursos a nivel de sistema, mentres que con ulimit e limits.conf o podemos facer a nivel de usuario. Os sistemas Windows non precisan deste tipo de control ao funcionar básicamente como sistemas monousuario. Sen embargo os sistemas Server que poden funcionar en modo multiusuario con Terminal Server dispoñen de ***Windows System Resource Manager*** que permiten unha xestión similar.

Si non controlamos o consumo de recursos que poden facer os usuarios nos expoñemos a ataques de denegación de servizo (DoS) mediante consumo excesivo de recursos. Un exemplo típico son as "***fork bombs***" que permiten bloquear un sistema facendo uso dun programa recursivo que consume os recursos de memoria da máquina.

Fork Bomb para Linux

```
bomb(){
 bomb|bomb&
};bomb
```

Forks Bomb paraWindows

```
%0|%0

:s
start %0
goto s
```

Control de Acceso Obrigatorio (MAC)

Un problema de seguridade moi común é a dos usuarios que traballan no sistema con máis privilexios dos necesarios, como consecuencia os procesos que lanzan terían polo tanto a autorización para alterar o sistema de xeitos non previstos. Un xeito de limitar este problema e o ***sandboxing*** utilizando ***chroot*** ou contedores como *Dockers* ou *LXC* no caso dos servicios. Pero para o traballo do usuario se precisa un xeito de paliar este problema: O control de acceso obrigatorio.

A diferenza do control de acceso discrecional no que o administrador establece as políticas de autorización, no control de acceso obrigatorio é o sistema o que as controla a partir dun ***perfil de seguridade***. Nin siquera os procesos con permiso de administración poderán facer nada que non autorice o perfil aplicado ao sistema. Deste xeito, aínda que un usuario traballe co nivel máximo de privilexios, si o control obrigatorio non lle permite realizar certas accións, non poderá facelas.

Si o control obrigatorio autoriza a acción se aplicarán os permisos discrecionais establecidos polo administrador.

O control MAC é un intento de aplicar a **política do mínimo privilexio** que establece que *un proceso únicamente debe ter o nivel de autorización necesario para levar a cabo o seu traballo*, e nada máis.

En Windows, este tipo de control se denomina **Mandatory Integrity Control** (*MIC*) e está integrado no sistema dende Windows Vista. En Linux existen varios sistemas sendo como SELinux, RSBAC, AppArmor ou Grsecurity. Os máis estendidos son SELinux e AppArmor

```
#include <tunables/global>

/usr/sbin/tcpdump {
  #include <abstractions/base>
  #include <abstractions/nameservice>
  #include <abstractions/user-tmp>

  capability net_raw,
  capability setuid,
  capability setgid,
  capability dac_override,
  network raw,
  network packet,

  # for -D
  capability sys_module,
  @{PROC}/bus/usb/ r,
  @{PROC}/bus/usb/** r,

  # for -F and -w
  audit deny @{HOME}/.* mrwkl,
  audit deny @{HOME}/.*/ rw,
  audit deny @{HOME}/.*/** mrwkl,
  audit deny @{HOME}/bin/ rw,
  audit deny @{HOME}/bin/** mrwkl,
  @{HOME}/ r,
  @{HOME}/** rw,

  /usr/sbin/tcpdump r,
}
```

Control de Acceso Obrigatorio (MAC)

MIC (Mandatory Integrity Control)

Os sistemas Windows dende Windows Vista utilizan MAC para xestionar a autorización mediante o ***Mandatory Integrity Control (MIC)***. En Windows todos os obxectos do sistema están etiquetados cun nivel de integridade (IL) que pode ser un dos seguintes (de menor a maior nivel):

- Untrusted: E o nivel mínimo, e corresponde a procesos que non están asociados con ningún usuario. Por exemplo, o navegador Chrome
- Low: E o nivel por defecto para os procesos que se conectan a internet. Os procesos con este nivel non poden escribir no rexistro e teñen un acceso limitado aos ficheiros do usuario.
- Medium: E o nivel por defecto para os procesos de usuario.
- High: E o nivel para os procesos lanzados polo administrador do sistema.
- System: E o nivel para os procesos importantes do sistema (kernel e servizos de sistema)
- Intaller: E o nivel maior, estes procesos poden acceder a calquera parte e eliminar calquera obxecto do sistema.

Un obxecto non pode acceder a obxectos con nivel de integridade maior ao que posee

O nivel de integridade se xestiona mediante o comando ***icacls***

Control de Acceso Obrigatorio (MAC)
SELinux

SELinux é o sistema MAC empregado por defecto por algunhas distribucións Linux como RedHat. As aplicacións supervisadas por SELinux unicamente poderán acceder aos recursos descritos na ***política de seguridade do proceso***. O módulo de seguridade SELinux do kernel se encargará de que os procesos cumpran coas regras descritas na política de seguridade.

SELinux protexe todos os obxectos do sistema (ficheiros, procesos, rede, comunicacións...) asignándolles unha etiqueta ou **contexto de seguridade**. Esta etiqueta é a que se utiliza na política para determinar o nivel de acceso e contén información como a identidade do propietario, o usuario, o nivel ou o rol, almacenándose nos inodos do sistema de arquivos (nos atributos estendidos) na forma **usuario_u:rol_r:tipo_t:nivel**

- **usuario_u:** E o "usuario SELinux" que non ten por qué coincidir co usuario do sistema. Normalmente un usuario SELinux está asociado a un grupo de usuarios Linux.
- **rol_u:** Todos os obxectos do sistema teñen por defecto o rol object_r, e se pode utilizar para establecer control de acceso baseado en roles (RBAC)
- **tipo_t:** Tipo de obxecto SELinux. O tipo de obxecto determina os permisos, e dicir a política de seguridade a utilizar. Tamén se denomina ***dominio***
- **nivel*:*** So se utiliza para políticas de seguridade multinivel (MLS) ou multicategoría (MCS) que permiten un control mediante etiquetas de ***sensibilidade*** ou ***categoría.*** A sensibilidade é xerárquica. Un proceso cun nivel determinado so poderá leer nos procesos de niveles inferiores, pero poderá ler e escribir nos de niveles superiores. A categoría non é xerárquica, e permite un control fino das regras.

Exemplos:

A carpeta raíz (inodo) dunha páxina web tería o seguinte contexto de seguridade: **system_u:object_r:httpd_sys_content_t:s0**

O home dos usuarios (inodo) tería o contexto **system_u:object_r:home_user_t:s0**

Un proceso servidor como Apache ou nginx tería o contexto **system_u:system_r:httpd_t:s0**

SELinux pode estar en 3 estados: **enforcing** (habilitado), **permisive** (se rexistran alertas) ou **disabled.** A elaboración de políticas de seguridade SELinux é laboriosa e complexa, polo que cos sistemas se subministran varios perfís "estándar" que logo se poden persoalizar. En Debian temos o paquete **selinux-policy-default** e **selinux-policy-src.** Para xestionar SELinux dispoñemos de ***semodule*** e ***semanage***

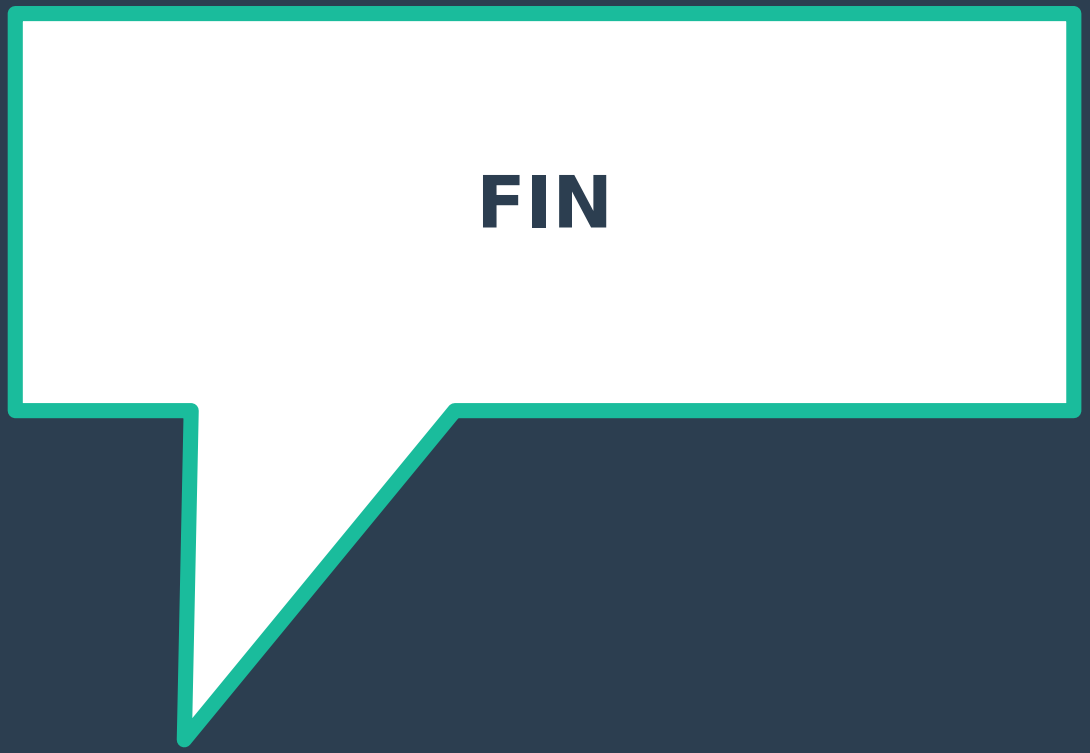
Control de Acceso Obrigatorio (MAC)

AppArmor

AppArmor é un sistema MAC moito máis simple de administrar e utilizar que SELinux desenvolto por Canonical (Ubuntu), e é o utilizado por defecto nos sistemas Debian.

Cada aplicación do sistema está restrinxida polo seu perfil de tarefa (en Debian en ***/etc/*apparmor.d**) que consiste en rutas a directorios, arquivos, dispositivos..... etc. Si unha aplicación non ten perfil, se executará sen restriccións.

De xeito similar a SELinux, AppArmor pode funcionar nos modos Enforcing ou Complain. A xestión se realiza cos comandos **aa-status**, **aa-complain**, **aa-enforce** e **apparmor-parser.**


FIN