Autorización e Acceso

Controis de Acceso Obrigatorios e Discrecionais

CONTROIS DE ACCESO OBRIGATORIOS (MAC)

Habitualmente os sistemas lanzan os procesos cos permisos do usuario que o inicia. Isto é fonte de moitos problemas de seguridade, xa que moitos servizos se lanzan co usuario ***root/administrador*** e polo tanto teñen acceso completo a todo o sistema. Si o servizo ten algunha vulnerabilidade, potencialmente compromete a todo o equipo e os servizos e usuarios en lugar de afectar unicamente ó servizo vulnerable.

Para mitigar este problema moitas veces se executan os servizos en contornos illados (*sandboxing*) como en contornas con *chroot*, *containers LXC* ou *dockers.*

Outra solución empregada hoxe en día de xeito estándar en sistemas Windows e Linux é o uso de controis de acceso obrigatorios (MAC)

> Un control de acceso obrigatorio consiste nun nivel de autorización que fixa o sistema para determinados obxectos (procesos, ficheiros ....) en lugar de facelo o administrador. Deste modo, o nivel de acceso que ten un servizo está limitado polo perfil de seguridade e non unicamente polos permisos baixo os que está funcionando o servizo.

O nivel de acceso nun sistema o xestiona en primeiro lugar o control obrigatorio. Os niveles de acceso obtidos despois de pasar polo control obrigatorio son restrinxidos logo polos niveis de acceso discrecional (DAC) que fixe o administrador.

Nos sistemas Windows se emprega un sistema de control de acceso obrigatorio implantado dende Windows Vista, denominado ***ILS (Integrity Level System).***

Windows Integrity Level System (ILS)

Nos sistemas Windows todos os obxectos do sistema están etiquetados cun ***nivel de integridade*** que pode ser un dos seguintes:

- **Untrusted:** Normalmente son procesos que non están asociados a ningún usuario en concreto e polo tanto teñen privilexios mínimos. Por exemplo o navegador Chrome...
- **Low:** E o nivel utilizado por defecto polos procesos que se conectan a internet. O Microsoft Edge ou a carpeta de arquivos temporais de internet funcionan con este nivel de acceso. Os procesos con este nivel non poden escribir no rexistro e teñen acceso limitado aos ficheiros do perfil do usuario.
- **Medium:** E o nivel de funcionamento habitual e o asignado por defecto polo sistema. Os usuarios normais teñen este nivel.
- **High:** E o nivel de integridade do Administrador do sistema. E o nivel que se obten cando executamos procesos como Administrador
- **System:** Este nivel está reservado para o sistema. O kernel e os servizos do sistema funcionan con este nivel.
- **Installer:** Este nivel é especial, e e o nivel máis alto de todos, polo que os procesos con este nivel son capaces de acceder a calquera parte e de eliminar calquera outro obxecto.

> Un obxecto so pode acceder a obxectos cun nivel de integridade igual ou menor ao nivel propio

Non todos os ficheiros e carpetas teñen que ter un nivel de integridade, pero **todos os procesos están sempre funcionando baixo un nivel de integridade. Si un proceso quere escribir dentro de outro debe ter como mínimo o mesmo nivel de integridade. Polo tanto, os procesos deben funcionar no nivel de integridade máis baixo posible.**

SELinux (Security Enhanced Linux)

SELinux é un sistema de control de acceso obrigatorio (MAC) empregado nas instalacións por defecto de algunhas distribucións Linux sobre todo en RedHat / Fedora.

Unha aplicación supervisada por SELinux pode acceder unicamente aos recursos que necesita descritos nunha ***política de seguridade do proceso.*** O acceso a procesos, portos, arquivos e directorios se controla mediante as regras definidas na política que implantará o módulo de seguridade do Kernel (SELinux). Este módulo autorizará ou denegará as operacións do sistema en base a esas políticas.

SELinux está orientado á protección de todos os obxectos do sistema (ficheiros, comunicación IPC e de rede, ... etc). Cada obxecto do sistema ten asignada unha etiqueta (***contexto de seguridade***) que utilizará a politica de seguridade para determinar o seu nivel de acceso. O contexto de seguridade conten diversa información como a identidade do propietario, o usuario, o rol, o tipo de nivel... etc) e se almacena nos inodos do sistema de arquivos (atributos extendidos). Os contextos de seguridade SELinux teñen a estrutura **usuario_u:rol_r:tipo_t:nivel**

- **Identidade ou usuario (usuario_u):** E o "usuario SELinux" que non coincide co usuario do sistema. Normalmente un usuario SELinux está asociado a un conxunto de usuarios UNIX, para distiguilo dos usuarios do sistema se lles engade o sufixo **_u**
- **Roles (rol_r):** rol de SELinux (para implementar control de acceso baseado en roles ou ***RBAC***). Todos os obxectos do sistema teñen por defecto o rol **object_r.**
- **Tipo (tipo_t):** tipo de obxecto SELinux. Determina os permisos de acceso, e dicir, a política de seguridade a utilizar. O tipo de obxecto de un proceso se denomina ***dominio***
- **Nivel (nivel)**: So se utiliza para políticas multi-nivel avanzadas. (MLS) ou multi-categoría (MCS). Permiten un control máis preciso mediante etiquetas adicionais ***sensibilidade*** e ***categoría***. A etiqueta ***sensibilidade*** é xerárquica: Un proceso dun nivel terá acceso de lectura a niveles inferiores, pero so poderá escribir nos niveles superiores ou iguais. A etiqueta ***categoría*** non é xerárquica e permite un control fino das regras.

Por exemplo:

- a carpeta root (inodo) dunha páxina web tería como contexto de seguridade **system_u:object_r:httpd_sys_content_t:s0**
- o home dos usuarios (inodo) sería **system_u:object_r:home_user_t_s0**
- un server como Apache ou nginx (dominio) sería system_u:system_r:httpd_t:s0

O sistema SELinux pode estar en tres estados:

- **Enforcing:** SELinux está habilitado. Se rexeitan os accesos non autorizados seguindo as políticas de seguridade.
- **Permisive:** Se permiten todos os accesos (so se aplica o DAC), pero se rexistran alertas
- **Disabled:** SELinux está desactivado

A elaboración de políticas de seguridade SELinux é complexa e laboriosa, polo que se subministran co sistema varias políticas "estándar" que se poden seleccionar segundo as necesidades do noso sistema. En Debian por exemplo temos o paquete ***selinux-policy-default*** e o fonte ***selinux-policy-src***

AppArmor

SELinux é un sistema DAC moi potente e configurable, pero esa versatilidade e posibilidade de xestión dos permisos resulta nunha xestión complexa. Canonical desenvolveu un sistema MAC coa idea de simplificar a súa administración e uso que é o que se emprega habitualmente nos sistemas da familia Debian: AppArmor.

Para cada aplicación do sistema se aplican as restriccións indicadas no ***perfil da tarefa*** (en Debian almacenados en **/*etc*/apparmor.d/**). O control de acceso se basa nas rutas (path) aos directorios, arquivos, dispositivos..... As aplicacións que non teñen perfil se executan sen restriccións.

AppArmor pode funcionar nos modos **Enforcing** e **Complain / Permisive** de xeito similar a SELinux, o modo **Audit** que rexistrará todos os accesos cumplan ou non con perfil.

CONTROIS DE ACCESO DISCRECIONAIS (DAC)

Unha vez que se cumpren os requisitos establecidos polo acceso obrigatorio (MAC) si existe, se deben satisfacer os requisitos establecidos polo control discrecional.

O control discrecional o configura o administrador do sistema establecendo que usuarios poden acceder a que obxectos configurando diferentes ***permisos de acceso***. Os permisos de acceso dispoñibles dependen do sistema de arquivos utilizado (por exemplo FAT non permite a xestión de permisos de acceso).

> Non se debe confundir os permisos de acceso cos **atributos dos ficheiros**. Os atributos dos ficheiros son "características" dos ficheiros e carpetas que non dependen do usuario que accede ós mesmos como "inmutable", "únicamente engadir ou "non actualizar data de acceso", "so lectura" ou "oculto".
>
> En Linux baixo ext4 se poden observar os atributos dos arquivos co comando **lsattr**, e cambialos con **chattr**, metres que en Windows se utiliza o comando **attrib**

En Windows se utiliza o sistema de arquivos NTFS e a base do seu control discrecional son as ACL (*Access Control List*). Estas listas de control asociadas aos obxectos indican os permisos de cada usuario ou grupo sobre cada obxecto do sistema e a súa herdanza nos obxectos internos. Windows posúe unha gran variedade de permisos e de grupos estándar que se poden controlar dende a liña de comandos mediante ***icacls:***

Mediante o comando wmic podemos localizar o SID dun usuario

/T *indica que a operación se realizará en todos os ficheiros e carpetas a partir de nome*
/C *indica que a operación continuará aínda que se produzan erros*
/L *indica que a operación se realiza sobre o enlace simbólico en lugar do ficheiro enlazado*
/Q *indica que non se visualicen as mensaxes de éxito*

icacls nome /*save aclfile* [/T][/C][/L][/Q] : Almacena as acls nun ficheiro de acls, que se pode restaurar con /restore
icacls directory [/substitute SidOld SidNew [...]] restore aclfile [/C][/L][/Q] : Aplica as acl do ficheiro ao directorio
icacls name /setowner user [/T][/C][/L][/Q] : Cambia o propietario
icacls name /findsid sid [/T][/C][/L][/Q] : Atopa todos os elementos que coinciden co patrón e que conteñan o SID indicado
icacls name /reset[/T][/C][/L][/Q] : Reemplaza as ACL polas herdadas
icacls name [/grant[:r] Sid:perm[...]] [/deny Sid:perm[...]][/remove[:g|:d]] Sid[...]] [/T][/C][/L][/Q] [/setintegritylevel Level:policy[...]]

/grant[:r] Sid:perm *garante que os dereitos de acceso especificados. :r indica que substitúa calquera permiso anterior en lugar de engadilo*
/deny Sid:perm *denega os dereitos de acceso especificados. Se engadira unha regra que denega os dereitos eliminando os "grant"*
/remove[:[g|d]] Sid *elimina das acl toadas as referencias ao Sid indicado, tanto os "deny" como os "grant".*

A orde nos ACE (Access Control Entries) a orde é denials explícitos, grants explícitos, denial herdados, grant herdados

Os permisos **perm** poden indicarse dunha destas dúas maneiras **:**

Unha secuencia de dereitos simples:

F – full access
M – modify access
RX – read and execute access
R – read-only access
W – write-only access

Unha lista separada por comas entre paréntese indicando dereitos específicos:

D – delete
RC – read control
WDAC – write DAC
WO – write owner
S – synchronize
AS – access system security
MA – maximum allowed
GR – generic read
GW – generic write
GE – generic execute
GA – generic all
RD – read data/list directory
WD – write data/add file
AD – append data/add subdirectory
REA – read extended attributes
WEA – write extended attributes
X – execute/traverse

DC – delete child
RA – read attributes
WA – write attributes

Se pode indicar a herdanza si os permisos se aplican sobre directorios:
(OI) – object inherit
(CI) – container inherit
(IO) – inherit only
(NP) – don't propagate inherit

EXEMPLOS:
icacls c:\windows* /save AclFile /T
icacls c:\windows\ /restore AclFile
icacls file /grant Administrator:(D,WDAC)
icacls file /grant *S-1-1-0:(D,WDAC)

En ***Linux***, se emprega habitualmente un sistema de permisos discrecionais moi simple na que se especifican os permisos de lectura(r) escritura(w) e execución(x) para o propietario e grupo. O permiso de execución aplicado a un directorio se interpreta como permiso para ingresar nel. Conceptos importantes son o ***umask***, ***SUID***, ***SGID*** , ***sudo*** e o ***Sticky Bit***.

O uso de programas SUID o SGID supon un importante problema de seguridade e o seu uso está totalmente desaconsellado.

umask establece os permisos por defecto dos novos ficheiros e directorios que crea o usuario. Se especifica con un número de catro díxitos en base 8 que comeza por 0. Os seguintes 3 díxitos son a negación dos permisos para o usuario, grupo e outros. Habitualmente os permisos umask por defecto son 0022, que crean arquivos con permisos ***rwxr-xr-x***:

umask	**permisos**
0	*7 (rwx)*
1	*6 (rw-)*
2	*5(r-x)*
3	*4(r--)*
4	*3(-wx)*
5	*2(-w-)*
6	*1(--x)*
7	*0(---)*

Cando un arquivo executable ten establecido o seu ***SUID***, se executará cos permisos do propietario do mesmo en lugar dos permisos do usuario que lanza o proceso. O mesmo se aplica ao bit ***SGID***, na que os procesos lanzados terán os permisos correspondentes ao grupo o que pertencen en lugar de os grupos aos que pertence o usuario que o lanza.

O "***sticky bit"*** cando se utiliza sobre un arquivo indica que o seu uso é moi frecuente, polo que é boa idea telo cacheado na memoria RAM. Cando se utiliza sobre directorios, indica que os arquivos que conten o directorio so poden ser borrados por ***root***, polo propietario do directorio ou polo propietario do arquivo.

```
chmod o+t  ficheiro|directorio
chmod o-t ficheiro|directorio
chmod u+s ficheiro
chmod u-s ficheiro
chmod g+s ficheiro
chmod g-s ficheiro
```

Mediante ***sudo*** é posible autorizar a usuarios concretos de xeito que podan utilizar unha serie de comandos con permisos de administración sin necesidade de coñecer máis que a propia password ou incluso sin especificar password. A configuración deses comandos se realiza mediante ***visudo*** que permite alterar e verificar a configuración en **/*etc*/sudoers.** A páxina do manual **man sudoers** proporciona documentación precisa sobre o seu formato.

Linux tamén incorpora no seu sistema ext4 ***acls POSIX***. Estas acl son moito máis simples que as de NTFS e permiten indicar os permisos de lectura, escritura e execución para cada usuario ou grupo do sistema, así como un sistema de "herdanza" especificando unha "acl por defecto". Os comandos para xestionar as acl POSIX son ***setfacl*** e ***getfacl***