Curso: (62612) Diseño de aplicaciones seguras

Fernando Tricas García

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
http://webdiis.unizar.es/~ftricas/
http://moodle.unizar.es/
ftricas@unizar.es

Tema VI: Control de acceso

Fernando Tricas García

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
http://webdiis.unizar.es/~ftricas/
http://moodle.unizar.es/
ftricas@unizar.es

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Control de acceso

- Una vez autentificados los usuarios en el sistema, éste necesita determinar a qué recursos pueden acceder.
- Hay muchos modelos de control de acceso
- Hablaremos de los modelos de Unix y Windows.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Control de acceso

- Una vez autentificados los usuarios en el sistema, éste necesita determinar a qué recursos pueden acceder.
- Hay muchos modelos de control de acceso
- Hablaremos de los modelos de Unix y Windows.

Pero antes ...

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Identificación

Identificación

Proporciona la identidad del usuario al sistema.

- Típicamente se basa en algún tipo de identificador (cadena, número, ...)
  - Representa de manera única el conjunto de datos que constituyen los aspectos relevantes para el sistema del sujeto identificado (la cuenta, 'account')
  - Cada usuario debería tener el suyo (preferentemente elegido por él).
  - Muy importante para la autentificación y la contabilidad

Autentificación

Autentificación

Es el proceso de validar la identidad del usuario.

- Naturalmente, que el usuario diga quién es no es suficiente
- Para tener acceso a los datos asociados a un determinado identificador es necesario proporcionar alguna evidencia.

`https://www.owasp.org/index.php/Authentication_Cheat_Sheet`

Credenciales

Credencial

La evidencia proporcionada por el usuario

- Algo que sabemos
  - Un secreto compartido (con matices)
- Algo que somos
  - Biometría, ...
- Algo que tenemos
  - Llave, tarjeta, ...

Auntentificación (más terminología)

Cliente ('Supplicant')

La parte en el proceso de autentificación que proporciona la identidad y la evidencia adecuada.

Servidor

La parte en el proceso que proporciona recursos al cliente y necesita estar seguro de que el cliente es quién dice ser para proporcionárselos.

Autoridad aseguradora

Almacenamiento para comprobar las credenciales del usuario.
Puede ser desde un fichero plano a sistemas mucho más complejos.

La comunicación puede ser posible entre cualquier par de ellos.

Autorización

Autorización

Proceso de determinar si un usuario **ya identificado y autentificado** tiene permiso para acceder a unos recursos de una forma determinada.

- Determinar qué operaciones pueden hacer qué usuarios
- Un usuario puede estar identificado con una identidad determinada, pero necesitar el acceso a un recurso con otra (identidad de autorización).
  - Si la identidad es de otro usuario, hablamos de representación ('impersonation')
    - Muy útil, algún servicio puede representar a sus usuarios frente a otros cuando éstos necesitan algo.

Proceso de acceso del usuario

Acceso del usuario

El proceso de identificación y autentificación se realizan casi siempre unidos: el sistema o la aplicación solicita al usuario su identificación y la credencial que lo demuestra.

- La aplicación o el sistema asocian al identificador un objeto de acceso ('access token') que se utilizará en todas las acciones realizadas por el usuario
- El proceso de acceso del usuario puede implicar también acciones no relacionadas con la seguridad (entorno de trabajo, contexto, aspecto, ...)

Contabilidad

Contabilidad

El proceso de mantener un registro de las acciones de los usuarios en el sistema. Proporciona información para determinar acciones autorizadas o no autorizadas. También intentos de acceso a recursos y autentificaciones exitosas y fallidas.

- Información sobre acciones autorizadas o no
- Intentos de acceso a recursos
- Autentificaciones exitosas y fallidas
- Acciones exitosas y fallidas
- Registro de actividad

El modelo de Unix

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

El modelo de Unix

Usuarios

- Cada usuario tiene:
  - Un nombre único con el que le identifica el sistema
  - Un entero único que le representa (UID).
  - Uno o varios enteros que representan a grupos de usuarios (GID)
    - Grupo primario ('login group') y grupos secundarios

root

El usuario con UID 0 puede hacer potencialmente cualquier cosa en el sistema.

El modelo de Unix

Ficheros y directorios

- Interfaz común y simple para interactuar con los ficheros

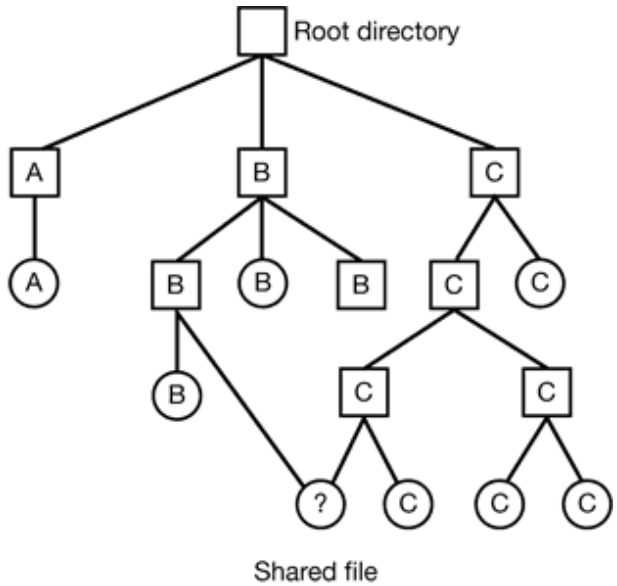
  escribir

  leer mover
- La misma abstracción para otros objetos (tuberías -'pipes'-, dispositivos, conexiones)
- Cada objeto del sistema: UID y GID

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

El modelo de Unix. Ficheros y directorios

- Organizados en una estructura jerárquica: un fichero se identifica mediante su nombre y localización.
  - Varios sistemas de ficheros (y situaciones diversas -montado/demontado-)
  - Un mismo fichero (o directorio) en distintos lugares (enlaces) $\longrightarrow$ Distintos nombres



"Tanenbaum. Sistemas Operativos. Diseño e Implementación"

El modelo de Unix

Procesos

- Un programa es un fichero ejecutable
- Un proceso es una instancia de un programa ejecutándose en un sistema
- Cada proceso tiene un identificador único (PID) y se ejecuta con los permisos de un usuario particular, el usuario efectivo.
  - Los permisos del **usuario** determinan lo que puede hacer el **proceso**
  - Algunos procesos pueden **cambiar** su usuario efectivo.

El modelo de Unix

Procesos

Asociado a un **proceso**:

- ID real ('Real User ID')
- ID almacenada ('Saved set-user ID')
- ID efectiva ('Effective user ID')

¿Y los grupos?

- ID de grupo real ('Real group ID')
- ID de grupo almacenada ('Saved set-group ID')
- ID de grupo efectiva ('Effective set-group ID')
- ID de grupo suplementaria ('Supplemental groups ID')
  - Se tienen en cuenta para acceder a los objetos

El modelo de Unix

- Al ejecutar un proceso, se le asigna la UID del que lo ejecutó (a sus hijos también)
- Acceso a los objetos a través de procesos
  - Se comprueba el EUID y el EGID (E de efectivo)
  - Habitualmente EUID==UID, y EGID==GID
    Pueden ser distintos en los llamados programas 'setuid' y 'setgid'.
    - Si hay fallos, cualquiera puede hacer todo lo que pueda hacer el propietario del programa

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

El modelo de Unix. Los ficheros.

- Hay permisos de acceso asociados a cada objeto:

Acción	
leer	r
escribir	w
ejecutar	x

Recordar:

El usuario 0 puede hacer cualquier cosa, independientemente de los permisos.

Con **cuidado**: si un grupo puede escribir en un fichero pero el propietario no puede, el propietario tendrá que cambiar los permisos antes de escribir.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Funcionamiento de los permisos

¿Qué es leer, escribir, ejecutar?

- Ficheros, está claro (escribir también es borrar)
- Para directorios
  - Leer: Listar directorios
  - Ejecutar: Entrar y usar sus ficheros
  - Escribir:
    - añadir
    - cambiar
    - borrar ficheros del directorio

    Muchos sistemas impiden modificar ficheros dentro de un directorio del que no se sea propietario 'sticky bit'

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Funcionamiento de los permisos

- Están distruidos en tres bloques:

Tipo	usuario	grupo	resto

Ejemplos:

-	rwx	rwx	rwx
d	rwx	r-x	r-x
d	rwx	rwx	rwt

Además

- En la primera posición, indicativos de directorios (d), enlaces (l), ficheros 'normales' (-), sockets (s), ...
- En la tercera de cada bloque puede ir: Setuid, setgid (s)
- Sticky bit (t) al final
- - en general, no.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Los permisos

Representación

Binario	Decimal	Explicación
100 000 000 000	4000	Set user ID on execution (SUID)
010 000 000 000	2000	Set group ID on execution (SGID)
001 000 000 000	1000	"Sticky bit"
000 100 000 000	0400	Read by owner
...	0200	Write by owner
	0100	Execute by owner
	0040	Read by group
	0020	Write by group
	0010	Execute by group
	0004	Read by other
	0002	Write by other
	0001	Execute by other

Funcionamiento de los permisos

- root (0) puede leer y escribir todo
  - Si alguien puede ejecutar un fichero, root también. Si no, no puede.
- Más permisos
  - setuid: si el ejecutable puede cambiar UID, EUID
  - setgid: si el ejecutable puede cambiar GID, EGID

Gestión desde los programas

`chmod()`, `fchmod()`

```
int chmod(const char *path, mode_t mode);
int fchmod(int fd, mode_t mode);
```

- Usaremos la segunda, para evitar posibles **condiciones de carrera** (hablaremos).

Dos parámetros: descriptor de fichero y `mode_t`

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

fchmod

Se trata de un OR de

S_ISUID	*04000*	*set user ID on execution*
S_ISGID	*02000*	*set group ID on execution*
S_ISVTX	*01000*	*sticky bit*
S_IRUSR (S_IREAD)	*00400*	*read by owner*
S_IWUSR (S_IWRITE)	*00200*	*write by owner*
S_IXUSR (S_IEXEC)	*00100*	*execute/search by owner*
S_IRGRP	*00040*	*read by group*
S_IWGRP	*00020*	*write by group*
S_IXGRP	*00010*	*execute/search by group*
S_IROTH	*00004*	*read by others*
S_IWOTH	*00002*	*write by others*
S_IXOTH	*00001*	*execute/search by others*

```
fchmod(fd, S_IRUSR | S_IWUSR )
```

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Los grupos

`chown()`, `fchown()`

```
int chown(const char *path, uid_t owner, gid_t group);
int fchown(int fd, uid_t owner, gid_t group);
```

- Sólo la segunda!
- Tres parámetros:
  - Descriptor del fichero
  - UID (`uid_t`)
  - GID (`gid_t`)

Si pasamos -1 a cualquiera de los dos últimos no se cambia ese parámetro.

```
fchown(fd, -1, 100)
```

Si fallan se devuelve -1 y se activa `errno`

Seguridad en directorios

- Si creamos un fichero en el que cualquiera puede escribir hay que tener cuidado
- Si el directorio es 'sticky' hay que tener cuidado al crearlo (no podrán borrarlo, ni renombrarlo)
- Si el directorio es de otro usuario, hay que tener en cuenta que puede cambiarlo todo
- En el directorio actual, o también en sus padres...

Cuando necesitamos más. Programación setuid

- Los bits setuid y setgid permiten acceder a ficheros o servicios para los que el usuario que ejecuta el proceso no tiene permisos.
  - Ejemplo: un registro de actividad del programa para todos los usuarios, pero que nadie pueda estropearlo.
- setuid y setgid permiten ejecutar procesos con los permisos del propietario del ejecutable (además, permiten cambiar UID, EUID, GID, EGID, ...)

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Cuando necesitamos más. Programación setuid

- Los bits setuid y setgid permiten acceder a ficheros o servicios para los que el usuario que ejecuta el proceso no tiene permisos.
  - Ejemplo: un registro de actividad del programa para todos los usuarios, pero que nadie pueda estropearlo.
- setuid y setgid permiten ejecutar procesos con los permisos del propietario del ejecutable (además, permiten cambiar UID, EUID, GID, EGID, ...)

- Especialmente peligroso para UID 0!
- Una forma habitual de ataque:
  - Entrar a través de alguno de los usuarios (agujeros del sistema, claves malas, ...)
  - Ejecutar algún programa setuid del usuario 0 (root)
  - Romperlo, para obtener sus privilegios

Programación setuid

- Un programa setuid que no es de root puede:
  - Intercambiar la UID y la EUID
    - `ftricas` corre un programa setuid de `arronategui`
    - El programa empieza con UID `ftricas` y EUID `arronategui`
    - Tiene los permisos de EUID y puede modificar ficheros de `arronategui`
    - Para poder modificar ficheros de `ftricas`, debe intercambiarlos

Cambiando UID's

- `seteuid()`
  int seteuid (uid_t euid);
  - Parámetros: uno
    - `uid_t`
  - La EUID deseada
    (Siempre que sea posible: usuario con privilegios altos, o usuario con alguna UID guardada).
- `setreuid()`
  int setreuid (uid_t ruid , uid_t euid);
  - Parámetros: dos
    - `ruid_t`, la primera la UID,
    - `euid_t`, la segunda la EUID
  - -1 deja el valor sin cambiar
  - Necesitamos `getuid()` y `geteuid()` para conocerlas

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Cambiando UID's

- Es conveniente cambiar de UID tan pronto como se pueda y sólo volver a la del otro usuario cuando sea necesario
  - Nuestra aplicación esta algo más protegida
  - Sobre todo, de nosotros

```
uid_mia = getuid(); uid_prop = geteuid()
setreuid(uid_mia, uid_prop); /* Lo primero de todo */
.....
setreuid(uid_prop, uid_mia);
```

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

setuid root

- A veces hacen falta privilegios que sólo tiene el root
  - Utilización de puertos ≤ 1024
  - `chown()`, `chroot()`, administración, acceso a los dispositivos
- Utilizar permisos especiales al principio, y abandonarlos tan pronto como sea posible.
  Ejemplo: primero usar los permisos y luego ...
  setuid(getuid()); /* *Dejar los privilegios de root* */
- Si no es posible, modularizar (por ejemplo, un demonio que corre como root, y con el que nos comunicamos desde el programa).

setuid root

- Los hijos heredan UID y EUID (¡cuidado!)
- Esto hace especialmente peligrosos a los scripts setuid
- Si hace falta, crear usuarios propios para las aplicaciones

Con grupos, parecido: `setgid()`, `setregid()`, ...

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

¿Y las credenciales?

Discutiremos algunas aproximaciones habituales...

- Cifrado de las claves

  `crypt()`

  Parámetros:
  - Clave
  - Semilla (2 caracteres aleatorios para 'cambiar' el DES).
- Se almacena en:
  `/etc/passwd`
  `/etc/group`
  O, en sistemas más modernos:
  `/etc/shadow`
  `/etc/gshadow`

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Las credenciales

- Los dos primeros caracteres contendrán la 'semilla' de forma que se pueda utilizar después para comprobar la clave.
- La semilla garantiza que dos claves iguales no se verán iguales

Ejemplo

Si la clave es

```
password123
```

para dos usuarios diferentes, se vería:

```
user1:r/2WTCYrcT6Eg:13030:0:99999:7:::
user2:40EQinNC6Or/.:13030:0:99999:7:::
```

- En la actualidad muchos sistemas permiten utilizar MD5 (hash) en lugar del DES.
- MD5 permite claves más largas (en realidad, utilizar más de los 8 primeros caracteres).

El fichero /etc/passwd

Ejemplo

```
root:x:0:0::/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/log:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/:/bin/false
news:x:9:13:news:/usr/lib/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucppublic:/bin/false
operator:x:11:0:operator:/root:/bin/bash
games:x:12:100:games:/usr/games:/bin/false
ftp:x:14:50::/home/ftp:/bin/false
smmsp:x:25:25:smmsp:/var/spool/clientmqueue:/bin/false
mysql:x:27:27:MySQL:/var/lib/mysql:/bin/false
rpc:x:32:32:RPC portmap user:/:/bin/false
sshd:x:33:33:sshd:/:/bin/false
gdm:x:42:42:GDM:/var/state/gdm:/bin/bash
apache:x:80:80:User for Apache:/srv/httpd:/bin/false
pop:x:90:90:POP:/:/bin/false
nobody:x:99:99:nobody:/:/bin/false
user1:x:500:100:User 1:/home/user1:/bin/bash
```

- login - nombre de usuario
- password - clave cifrada.
  * (cuenta no usada, cambiar clave)
  x (la clave no está aquí)
- UID - Identificador del Usuario
- GID - Identificador del Grupo
- Inf. Usuario -
- Directorio de trabajo
- Intérprete de instrucciones.

Típicamente legible para todos los usuarios.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

El fichero /etc/group

Ejemplo

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:
tty:x:5:
disk:x:6:
lp:x:7:
wwwadmin:x:8:
kmem:x:9:
wheel:x:10:
mail:x:12:cyrus
news:x:13:news
```

- Nombre del grupo
- Clave. En caso de que se permita el acceso a usuarios no listados.
- GID
- Lista de usuarios

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

El fichero /etc/shadow

```
login:password:Daysince:Daysafter:Daysmust:dayswarn:daysexpire:daysince:reserved
juan:$1$.QKDPc5E$SWlkjRWexrXYgc98F.:11956:0:90:5:30:12197:
```

- login - login name
- password - password in encrypted form, which is 13 to 24 characters long.
- Daysince - Days since Jan 1, 1970 that the password was changed
- Daysafter - Days before the password may be changed
- Daysmust - Days after which the password must be changed
- dayswarn - Days before the password will expire (A warning to the user)
- daysexpire - Days after the password expires that the account is disabled
- daysince - Days since Jan1, 1970 that the account is disabled.
- reserved - Reserved field.

El fichero /etc/gshadow

```
general:!!:shelley:juan,bob
```

- Nombre del grupo
- Clave cifrada
- Administradores del grupo
- Miembros del grupo

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Hay más

`/etc/publickey`
`/etc/cram-md5.pwd`
`htpasswd`
/etc/smb.d/smbpasswd
Kerberos, ...

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Atención a ...

- Nombres proporcionados por el usuario ('../../', NUL)
- Sitios 'peligrosos'
  - Directorios públicos y/o temporales (/tmp, /var/tmp)
  - Directorios proporcionados por el usuario
  - Ficheros y directorios nuevos (creación)
  - Ficheros y directorios de otros usuarios
- Ficheros 'interesantes'
  - Ficheros de configuración (sistema, usuarios, web, ...)
  - Ficheros de registro ('log')
  - Ficheros de dispositivos ('dev')
  - Tuberías con nombre ('named pipes')
  - El sistema de ficheros `proc` .

El modelo de Windows

D. Todorov
'Mechanics of User Identification and Authentication. Fundamentals of Identity Management'. Auerbach Publications.

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Control de acceso en Windows NT, 2000, XP, ...

- Primera diferencia: mayor granularidad
  - Ejemplo: derecho de transferir la propiedad de un fichero vs propiedad del fichero
- Permisos basados en capacidades

En Windows

- La unidad fundamental de abstracción en este caso son los objetos:
  - Tipo: p.ej. fichero
  - Se crean instancias: p.ej. `C:kk.txt`
- Lista de objetos 'asegurables'
  - Objetos de servicios de directorio
  - Objetos de mapeo de ficheros
  - Objetos de sincronización de procesos
  - Objetos 'Job'
  - Tuberías con nombre y anónimas
  - Elementos compartidos por la red
  - Ficheros y directorios NTFS
  - Impresoras
  - Procesos y threads
  - Claves de registro (pero no sus valores)
  - Servicios
  - Objetos de gestión de ventanas (pero no las ventanas)

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Windows NT, 2000, XP, Vista

Espacios de nombres

- Los objetos sin nombre se pueden compartir sólo mediante el identificador ('Handle').
- Los objetos con nombre se almacenan jerárquicamente.
  - Hay que tener cuidado al crearlos (que no estén ya creados y sean de otros).
  - En Vista hay espacios de nombres privados (para aligerar el problema).

Windows NT, 2000, XP, Vista

Identificadores de objetos ('Object handles')

- Se accede a ellos a través del tipo de datos HANDLE.
- El acceso se controla mediante una DACL ('Discretionary Access Control List')
  - ¡Independientemente de que tenga nombre o no!
- Devolución inconsistente (a veces -1, otras NULL, ...)
- Herencia de identificadores: un hijo no tiene acceso a los objetos del padre salvo que los herede

Sesiones

Forma de encapsular los datos relativos a una entrada ('logon') al sistema.

- Es el mecanismo para encapsular datos relativos a la entrada al sistema
  - Derechos de acceso
  - Datos accesibles en la sesión
  - Comportamiento de los procesos en la sesión

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Identificadores de seguridad (Security ID's)

La identidad de una entidad a la hora de permitirle acceso a recursos.

- Nivel de revisión, identificador de autoridad, subautoridad, e identificador relativo ('RID').
  - Una SID:
    S-1-5-32-545
    S Siempre lo mismo
    1 Versión de la especificación de SID. Común a todas las versiones de Windows hasta ahora
    5 identificador de autoridad de SECURITY_NT_AUTHORITY
    32 subautoridad de cuentas del sistema ('built-in acc.s')
    545 Grupo de usuarios

Usuarios y grupos

- Dos tipos de cuentas de usuarios
  - Predefinidas
  - Definidas para el sistema
- Tipos de grupos
  - Grupos del sistema: se crean automáticamente y de manera dinámica. Las gestiona el sistema operativo.
  - Predefinidas
  - Definidas por el usuario

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Usuarios y grupos

- La parte del dominio se genera aleatoriamente durante la instalación
- Todos los usuarios y grupos tienen el mismo identificador de dominio
- No se reutilizan los RIDs, para evitar que alguien tenga acceso a información de usuarios anteriores

Ejemplo de grupos del sistema

- SID: S-1-0 Name: Null Authority Description: An identifier authority.
- SID: S-1-0-0 Name: Nobody Description: No security principal.
- SID: S-1-1 Name: World Authority Description: An identifier authority.
- SID: S-1-1-0 Name: Everyone Description: A group that includes all users, even anonymous users and guests. Membership is controlled by the operating system.

  Note By default, the Everyone group no longer includes anonymous users on a computer that is running Windows XP Service Pack 2 (SP2).
- SID: S-1-2 Name: Local Authority Description: An identifier authority.
- SID: S-1-3 Name: Creator Authority Description: An identifier authority.
- SID: S-1-3-0 Name: Creator Owner Description: A placeholder in an inheritable access control entry (ACE). When the ACE is inherited, the system replaces this SID with the SID for the object's creator.

`http://support.microsoft.com/kb/243330`

Cuentas predefinidas, grupos y SIDs

- 500 - Administrator S-1-5-21—-500
- 501 - Guest S-1-5-21—-502
- 502 - KRBTGT S-1-5-21—-502
- 512 - Domain Admins S-1-5-21—-512
- 513 - Domain Users S-1-5-21—-513
- 514 - Domain Guest S-1-5-21—-514
- 515 - Domain Computers S-1-5-21—-515
- 516 - Domain Controllers S-1-5-21—-516
- 517 - Cert Publishers S-1-5-21—-517
- 518 - Schema Admins S-1-5-21—-518
- 519 - Enterprise Admins S-1-5-21—-519
- 520 - Group Policy Creator Owners S-1-5-21—-520
- 533 - RAS and IAS Servers S-1-5-21—-533

`http://www.winzero.ca/WEllKnownSIDs.htm`

Derechos de acceso ('logon rights')

Determinan si un usuario puede establecer una conexión de entrada y qué tipo de sesión es la permitida

Constant/value	Description
SE_BATCH_LOGON_NAME TEXT("SeBatchLogonRight")	Req. for an acc. to log on using the batch logon type.
SE_DENY_BATCH_LOGON_NAME TEXT("SeDenyBatchLogonRight")	Exp. denies an acc. the right to log on using the batch logon type.
SE_DENY_interac._LOGON_NAME TEXT("SeDenyinterac.LogonRight")	Exp. denies an acc. the right to log on using the interac. logon type.
SE_DENY_NETWORK_LOGON_NAME TEXT("SeDenyNetworkLogonRight")	Exp. denies an acc. the right to log on using the network logon type.
SE_DENY_REMOTE_interac._LOGON_NAME TEXT("SeDenyRemoteinterac.LogonRight")	Exp. denies an acc. the right to log on rem. the interac. logon type.
SE_DENY_SERVICE_LOGON_NAME TEXT("SeDenyServiceLogonRight")	Exp. denies an acc. the right to log on using the service logon type.
SE_interac._LOGON_NAME TEXT("Seinterac.LogonRight")	Req. for an acc. to log on using the interac. logon type.
SE_NETWORK_LOGON_NAME TEXT("SeNetworkLogonRight")	Req. for an acc. to log on using the network logon type.
SE_REMOTE_interac._LOGON_NAME TEXT("SeRemoteinterac.LogonRight")	Req. for an acc. to log on rem. using the interac. logon type.
SE_SERVICE_LOGON_NAME TEXT("SeServiceLogonRight")	Req. for an acc. to log on using the service logon type.

`http://msdn2.microsoft.com/en-us/library/bb545671(VS.85).aspx`

Tokens de acceso

- El contexto de seguridad de un proceso
  - The security identifier (SID) for the user's account
  - SIDs for the groups of which the user is a member
  - A logon SID that identifies the current logon session
  - A list of the privileges held by either the user or the user's groups
  - An owner SID
  - The SID for the primary group
  - The default DACL that the system uses when the user creates a securable object without specifying a security descriptor
  - The source of the access token
  - Whether the token is a primary or impersonation token
  - An optional list of restricting SIDs
  - Current impersonation levels
  - Other statistics

`http://msdn2.microsoft.com/en-us/library/aa374909.aspx`

Se crea cada vez que alguien se conecta y se **copia** a cada proceso de la sesión

Windows NT, 2000, XP, Vista

Privilegios

- SeAssignPrimaryTokenPrivilege
- SeAuditPrivilege
- SeBackupPrivilege
- SeChangeNotifyPrivilege
- ...

`http://msdn2.microsoft.com/en-us/library/bb530716(VS.85).aspx`

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Windows NT, 2000, XP, Vista

Lista de grupos

- Se comprueban los DACL's de la lista de grupos a los que pertenece.
- Se genera al entrar al sistema y no se puede actualizar durante la sesión.
  SE_GROUP_ENABLED
  SE_GROUP_ENABLED_BY_DEFAULT
  SE_GROUP_INTEGRITY
  SE_GROUP_INTEGRITY_ENABLED ...

`http://msdn2.microsoft.com/en-us/library/aa379624(VS.85).aspx`

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Windows NT, 2000, XP, Vista

Además para añadir mas restricciones:

- Restricted Tokens (token que tiene una lista restringida no vacía de SID)
- Software Restriction Policies (SAFER)
- Cambios de contexto
- Representación (‘impersonation’)

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Windows NT, 2000, XP, Vista

Descriptores de seguridad (asociados a los objetos asegurables). Se componen de:

- SID del propietario
- SID del grupo
- DACL
- SACL

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Windows NT, 2000, XP, Vista

Máscaras de acceso (bits que describen qué debe tener el SID solicitante)

- Derechos de acceso estándar (8 bits reservados, 5 usados. Se aplican a cualquier tipo de objeto)

DELETE	The right to delete the object.
READ_CONTROL	The right to read the information in the object's security descriptor, not including the information in the SACL.
SYNCHRONIZE	The right to use the object for synchronization. This enables a thread to wait until the object is in the signaled state. Some object types do not support this access right.
WRITE_DAC	The right to modify the DACL in the object's security descriptor.
WRITE_OWNER	The right to change the owner in the object's security descriptor.

- Derechos de acceso específicos (depende del tipo de objeto)
- Derechos de acceso genéricos (se aplican a todos los objetos, pero su significado depende del objeto)

GENERIC_ALL	Read, write, and execute access
GENERIC_EXECUTE	Execute access
GENERIC_READ	Read access
GENERIC_WRITE	Write access

`http://msdn2.microsoft.com/en-us/library/aa446632(VS.85).aspx`

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Windows NT, 2000, XP, Vista

Herencia de ACLs

- Hay que tener cuidado con la herencia (se puede configurar qué heredarán los hijos)

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Windows NT, 2000, XP, Vista

Acceso a los interfaces de programación de los descriptores de seguridad

- Control a bajo nivel de ACLs
  `AddAce`, `AddAccessAllowedAce`, `AddAccessDeniedAce`, `GetAce`
- Cadenas de descriptores de seguridad
  Permiten el manejo de los datos con un formato mas cómodo, basado en texto

Entonces ...

- ACL nula: todo el mundo puede hacer lo que quiera con el objeto
- Orden ACE's ('Access Control Entry'): un ACL es una lista de ACE's y el orden es importante.

Windows NT, 2000, XP, Vista

Procesos e hilos

- Un proceso no se ejecuta. Simplemente es un contenedor de hilos ('threads').
- Todos los hilos asociados a un proceso comparten el espacio de direcciones y el contorno de seguridad ('security boundary'). Todos tienen acceso sin restricciones a cualquiera de los del mismo proceso.
- Carga de un proceso (cuidado con los espacios en blanco, trata de abrir ejecutables sin mas).
  `CreateProcess(NULL, "D:\\Program Files\\My Application\\ my app.exe", ...)`
  1. `D:\Program.exe`
  2. `D:\Program Files\My.exe`
  3. ...
- `ShellExecute()` y `ShellExecuteEx()` ejecutan a través de la Windows Explorer API (no tienen por qué ejecutar el fichero, si no abrirlo con la aplicación predefinida).

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Windows NT, 2000, XP, Vista

- Carga de DLLs
  1. Directorio de la aplicación
  2. System32
  3. Directorio del sistema
  4. Windows o WINNT
  5. Directorio actual (paso 2 antes de XP)
  6. PATH

  Además: `SetDllDirectory()`
- Redirección DLLs
  Para evitar 'DLL hell' (pero tiene problemas con que da preferencia al directorio actual o al especificado. Se puede evitar con el registro).
- Manifiesto de la aplicación ('Application Manifest') (carga de la aplicación, bibliotecas, módulos, ...)

Windows NT, 2000, XP, Vista

Acceso a ficheros

- Son objetos y sus permisos definen los permisos para el fichero 'físico'.

FILE_ADD_FILE
FILE_ADD_SUBDIRECTORY
FILE_ALL_ACCESS
FILE_APPEND_DATA
FILE_CREATE_PIPE_INSTANCE
...

`http://msdn2.microsoft.com/en-us/library/aa364399(VS.85).aspx`

- El acceso es a través de identificadores de ficheros, no de nombres.
  Cuidado con `CreateFile()` con nombres ya usados.
- Cuidado con los nombres (más adelante mas)
- Cuidado con los objetos que son como ficheros y también con los dispositivos

Windows NT, 2000, XP, Vista

El registro

- Base de datos centralizada de configuraciones
- Organizado en forma de árbol, por claves, con subclaves y valores en las hojas.
- Las claves son objetos asegurables
- Como cualquier objeto con nombre, están sujetos a la 'ocupación' previa.

Los permisos

- Listas de Control de Acceso (*Access Control Lists - ACL*)
  - Cada entidad tiene asociada su ACL
  - Contiene una lista de usuarios y grupos y sus capacidades asociadas
  - Con herencia

Las credenciales

- Las claves están guardadas con un 'hash' y no son fácilmente accesibles
- El fichero SAM (Security Accounts Manager) tiene dos hashes de las claves
  - LAN Manager hash. **Compatibilidad hacia atrás**
  - Hash de NT

En ambos casos se utiliza DES para cifrar el hash que se obtiene.

- En LM se utiliza también DES para obtener el hash con una cadena fija de tamaño fijo. Tiene más 'defectos': se pasa a mayúsculas, ...
- En NT se utiliza MD4.

Los algoritmos no están publicados.

Algunas generalidades para terminar

Compartimentalización

- Las jaulas `chroot`
- La caja de arena *sandboxing*

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Mayor granularidad

- Algunos sistemas tipo Unix ofrecen soluciones para obtener mayor granularidad
- Trusted Solaris, HP
- También existe una tecnología basada en capacidades (*capabilities*)
- Acceso de control obligatorio (*mandatory access control*), para evitar la propagación de permisos...