



GRUPO DE SEGURIDAD INFORMÁTICA

---

# Fundamentos de la Seguridad Informática

## Seguridad en Sistemas Windows

### control de acceso



**GSI - Facultad de Ingeniería**



GRUPO DE SEGURIDAD INFORMÁTICA

# Control de Acceso: Plan

---

A continuación vamos a ver:

- *Principals, Sujetos y Objetos*
- Algoritmo de decisión
- Dónde y como son evaluadas en W2K



# Principals

- Son las entidades activas en las políticas de seguridad, esto es, objetos sobre los que se puede permitir o denegar el acceso
- Por ejemplo: usuarios, alias, usuario de dominio, grupos o máquinas
- Asocian <username, **Security ID**entifier>
- Hay distintos “tipos”: Local / Domain / Universal



# Principals (2)

## Grupos globales

---

- Un *grupo* (global) es una colección de **SIDs** manejada por el DC
- Tiene su propio **SID**, para que puedan anidarse
- Constituyen una **capa de control intermedia**, esto es:  
*se da permisos a un grupo para acceder a un objeto, y luego se da permisos a los usuarios sobre el objeto agregando usuarios al grupo*



# Principals (3)

## Alias (grupos locales)

---

- Son un conjunto de SIDs de usuarios y grupos manejados por el DC o por el LSA (Local Security Authority)
- No pueden ser anidados
- Se utilizan para crear roles locales



GRUPO DE SEGURIDAD INFORMÁTICA

# Principals (4) formato SID

- formato de un SID: **S-R-I-SA-SA-S-A-N**
  - S: la letra S
  - R: el número de revisión
  - I: identifier authority (48-bit)
  - SA: subauthority (32-bit)
  - N: identificador relativo, único (**RID**) en el espacio de nombres de la autoridad



# Principals (5) Ejemplos

- Everyone: **S-1-1-0**
- SYSTEM: **S-1-5-18**  
Principal con el que ejecuta localmente el Sistema Operativo en una maquina
- Administrator: **S-1-5-21-<local authority>-500**  
Cuenta de usuario creada durante la instalación del S.O.
- Administrators: **S-1-5-32-544**  
Grupo predefinido (built-in) del sistema con privilegio de administrador. Inicialmente integrado por la cuenta *Administrator*



GRUPO DE SEGURIDAD INFORMÁTICA

# Principals (6) Ejemplos

- Domain Administrators:  
**S-1-5-21-<domain authority>-512**  
Grupo global, miembro del alias  
Administrators en todas las maquinas del  
dominio
- Cuenta Guest  
**S-1-5-21-<authority>-501**





GRUPO DE SEGURIDAD INFORMÁTICA

# Principals

Podemos ver información de los principals con los siguientes comandos:

Usuarios locales y alias:

- > *net user*
- > *net localgroup*

Usuarios, grupos y alias de dominio:

- > *net user /domain*
- > *net group /domain*
- > *net localgroup /domain*

Miembros de un grupo:

- > *net group "UK Employees" /domain*

Información de un usuario:

- > *net user diego /domain*



GRUPO DE SEGURIDAD INFORMÁTICA

# Principals

- Podemos también ver los SID de los usuarios de un equipo:

**Start** | **run** | *regedt32.exe*

Ir a **HKEY\_USERS**



# Sujetos

- Son las entidades activas en el Sistema Operativo:  
**threads o procesos**
- Las credenciales de seguridad para un sujeto se guardan en access tokens
- SIDs sirven como atributos de identidad y autorización
- Contiene la Unión de todos los privilegios asignados a los SIDs

Access Token:

<b>User SID</b>
<b>Group / Alias SIDs</b>
<b>Privilegios</b>
<b>Defaults for New Objects</b>
<b>Miscellaneous</b>




# Sujetos: privilegios

- Controlan el acceso a los recursos del sistema
- Se asignan a usuarios, grupos y alias por máquina
- Son diferentes a los derechos de acceso (*access rights*)
- Ejemplos:
  - backup de archivos y directorios
  - generar auditorías de seguridad
  - apagar el sistema



# Sujetos: Autenticación Usuarios

- Se encarga la *winlogon.exe* y GINA DLL, iniciado por la secuencia de atención segura (CTRL+ALT+DEL)
  - Se pasa la información al LSA (*lsass.exe*)
  - LSA invoca paquete de autenticación que retorna el SID del usuario y los SID de los grupos a los que pertenece
- 
- LSA genera access token y se lo pasa el logon process

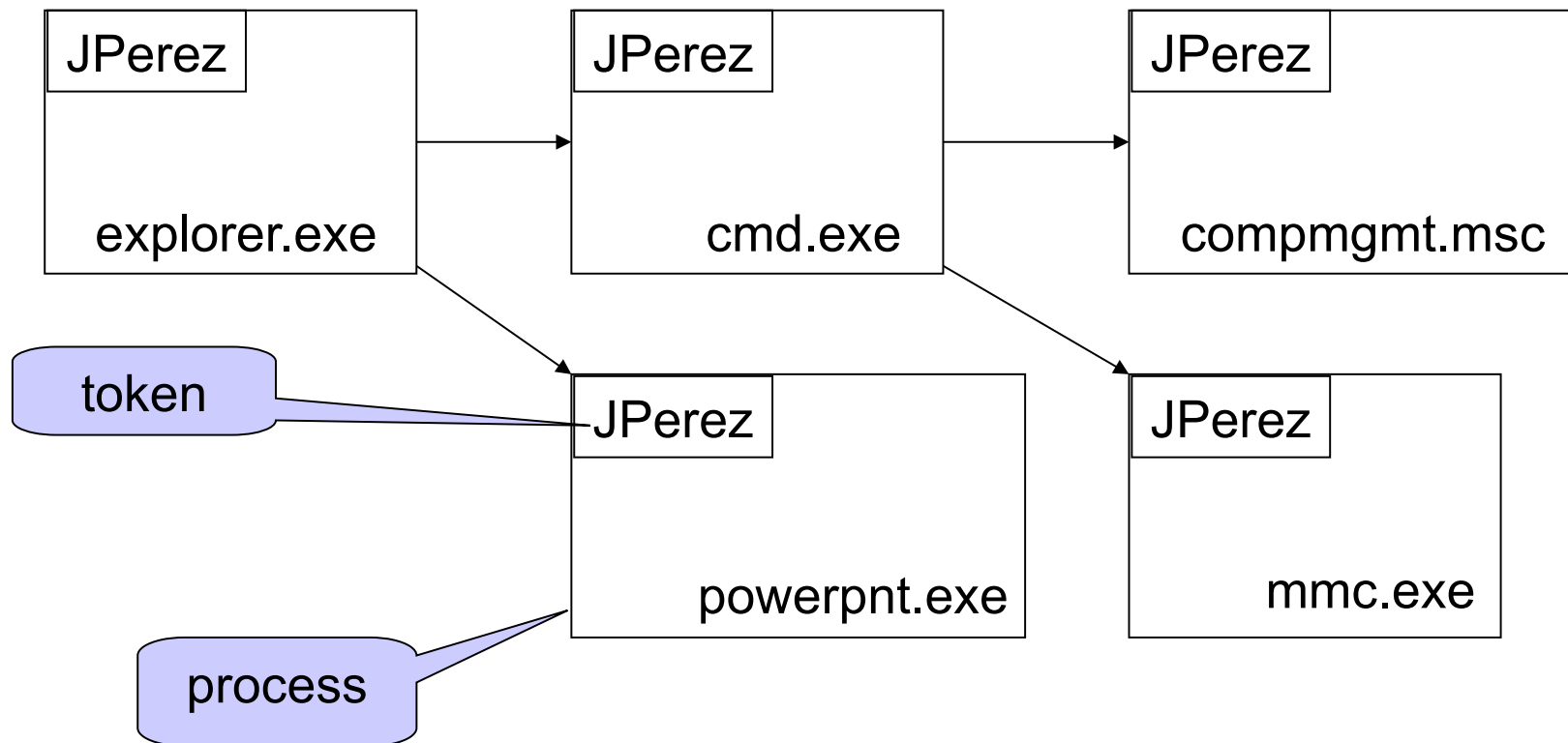


# Creación de Sujetos

- En el próximo paso, el proceso de logon inicia una shell (explorer.exe)
- Esto se hace en una nueva sesión bajo el usuario (*principal*) que se ha autenticado
- Estos procesos son los sujetos para el propósito del control de acceso
- Un proceso crea nuevos procesos (locales) invocando a *CreateProcess*
- Cada proceso obtiene una copia del *token* del proceso padre

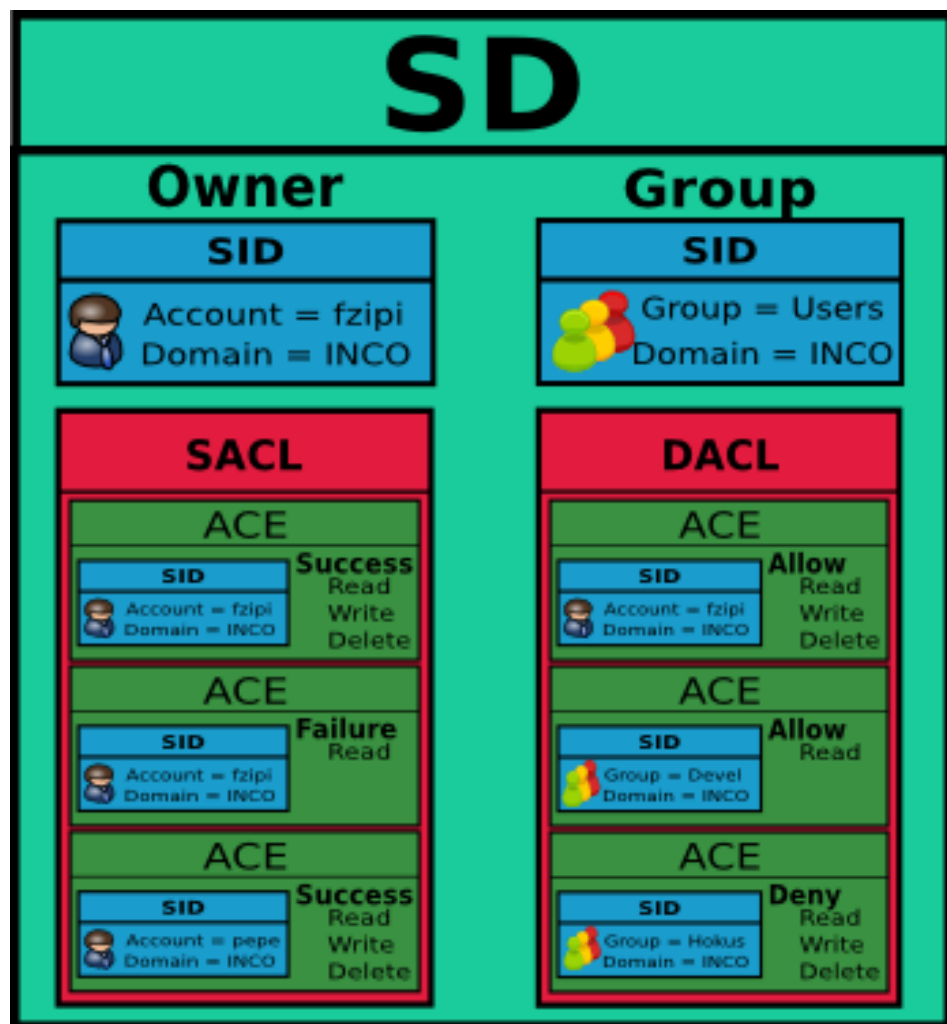


# Creación de sujetos





# Objetos Security Descriptors



- Son entidades **pasivas** en las operaciones de acceso
- Hay al menos 2 tipos de objetos: *ejecutables* (proc/threads) *filesystem* (archivos/dirs.)
- La **DACL** (*Discretionary ACL*) determina quién tiene acceso
- La **SACL** (*System ACL*) define la *audit policy*





# Objetos: permisos / access rights

---

- Un **permiso o access right**, es una autorización para hacer una operación particular sobre un objeto
  - Los permisos se codifican en 32 bits (access mask)
  - Permisos estándar
    - **DELETE**: eliminar el objeto
    - **READ\_CONTROL**: acceso de lectura sobre el owner, group o DACL del security descriptor
    - **WRITE\_DAC**: acceso de escritura a la DACL
    - **WRITE\_OWNER**: permiso de escritura sobre dueño
-



# Objetos: access rights

- Permisos genéricos
  - **GENERIC\_READ**
  - **GENERIC\_WRITE**
  - **GENERIC\_EXECUTE**
  - **GENERIC\_ALL**
- Cada clase de objetos, mapea los permisos genéricos a permisos reales



# Objetos: owner

- El dueño de un objeto se especifica con el SID del principal
- Es un principal :)
- Se asignan cuando el objeto es creado
- El dueño siempre tiene permiso de READ\_CONTROL y WRITE\_DAC
- Puedo “hacerme” dueño de un objeto si tengo el privilegio ‘Take ownership of files and other objects’ (SeTakeOwnershipPrivilege)



# Decisiones sobre acceso (1)

- Cada tipo de objeto tiene un manejador que se encarga de **crearlos** y **verificar** que un proceso tenga el derecho de usar dicho objeto (Policy Enforcement Point, PEP)
- Control Acceso -> se hace invocando una función de decisión en el Security Reference Monitor (SRM) (Policy Decision Point, PDP)
- Las decisiones de control de acceso toman como INPUT: el sujeto pidiendo acceso, el objeto al cual se quiere acceder y el tipo de acceso requerido (access mask)



# Algoritmo de decisión (1)

## **Entrada del algoritmo:**

el token del sujeto, la ACL del objeto y la “access mask” solicitada (los permisos requeridos)

A) Si no hay DACL (NULL DACL) el acceso es otorgado

B) **Si** el sujeto es el dueño del objeto, **entonces**

**si** la access mask contiene Read\_Control o Write\_DAC, el acceso es otorgado

**sino** se construye una “access mask” de permisos otorgados o garantizados y se busca en las ACE de la DACL



# Algoritmo de decisión (2)

C) Para cada ACE de la DACL se compara el SID del sujeto con el de la ACE, pudiéndose dar 3 casos:

1. La ACE no contiene un SID que corresponda, entonces se saltea
2. La ACE contiene un SID que corresponde y especificando *access denied*, entonces se niega el acceso y no se continúa el proceso
3. La ACE contiene un SID que corresponde y especificando *access allowed*, **entonces si** la access mask en la ACE, junto a las access mask de todas las anteriores ACEs que matchearon, contiene los permisos solicitados **entonces** se permite el acceso y se termina, **sino** se continúa buscando



# Algoritmo de decisión (3)

- Para que las ACEs negativas tomen precedencia sobre las positivas, deben ir al principio de la DACL