Diseño de Sistemas Operativos

Capítulo 9
Seguridad y Protección en Sistemas Operativos

Extraído de "*Sistemas operativos: una visión aplicada*"

Términos y Conceptos

- Sujetos (usuarios o tareas)
- Objetos (recursos, datos, etc.)
- Políticas de seguridad
- Mecanismos de seguridad
- Principio de mínimo privilegio
  - Sólo otorgar el privilegio realmente requerido

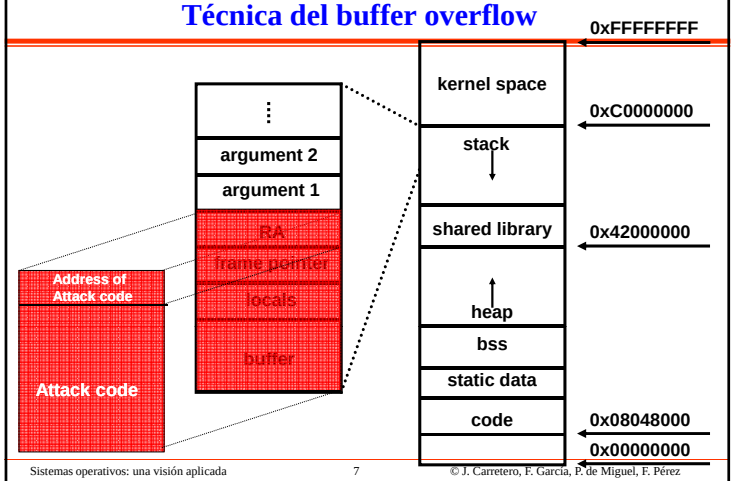
Tipos de amenazas

- Tipos de amenazas:
  - Exploits
  - Caballo de Troya.
  - Puertas traseras.
  - Virus.
  - Gusanos.
  - Ataques por denegación de servicio

Buffer overflow

- ***Exploits***: programas que aprovechan un error en otro programa para violar la política de seguridad
- Se popularizó en 1997 pero sigue vigente
- Idea fundamental: corromper la pila de un programa escribiendo más allá de los límites de un array
- Especialmente peligroso en programas con *setuid*

Técnica del buffer overflow



Virus

- Secuencia de código que se inserta en un ejecutable
- Etapas de un virus
  - *Fase latente*: El virus está dormido y se despierta por un evento
  - *Fase de propagación*: El virus inserta copias de sí mismo en otros programas
  - *Fase de activación*: El virus se activa para realiza las funciones para las que fue concebido
  - *Fase de ejecución*: La función en cuestión se realiza

Estrategias de los antivirus

- Prevención & Detección
  - Aumento en tamaño de ejecutables
  - Su firma (secuencia de instrucciones, aunque puede cambiar al propagarse: mutar)
  - Integridad de ejecutables (almacenar *checksums*)
  - Detectar operaciones potencialmente peligrosas
- Si la eliminación no es posible debemos deshacernos del programa infectado
- Estrategias más sofisticadas:
  - Descifrado genérico

Descifrado genérico

- *Generic decryption* (GD)
- Permite la detección de virus polimórficos con altas velocidades
- Todos los ficheros se recorren con un escáner GD que contiene
  - Emulador de CPU: ordenador virtual basado en software
  - Escáner de firma de virus: recorre el código buscando firmas de virus
  - Módulo de control de emulación: controla la ejecución del código a analizar
- Dificultad: Cuánto tiempo se tiene que ejecutar una interpretación

Gusanos

- Características básicas:
  - Es un código malicioso cuya principal misión es reenviarse a sí mismo
  - No afectan a la información de los sitios que contagian o se comportan con un virus
  - Consumen amplios recursos de los sistemas y los usan para infectar a otros equipos
- El "Gusano de Internet" (1988):
  - Se basaba en errores en servidores (*fingerd*, *sendmail*)
  - No involucraba ninguna operación perjudicial
  - Dejó fuera de servicio a miles de máquinas
  - Su propagación "agresiva" colapsaba las máquinas
  - Enorme publicidad
  - Provocó la creación del CERT (Computer Emergency Response Team)

Conejos o bacterias

- No dañan al sistema
- Se reproducen hasta que la cantidad de recursos consumidos se convierte en una negación de servicio para el sistema afectado

```
main(){
    while(1){
        malloc(1024);
        fork();
    }
}
```

- Solución: Utilidades del *kernel* para limitar recursos de los usuarios

Caballos de Troya (troyanos)

- Programa útil que además hace cosas no autorizadas
- El usuario ejecuta voluntariamente el programa malicioso
- *Trojan mule* o mula de Troya: es el falso programa de *login*

```
luisa:~$ cat trojan
clear
printf "‘uname -n‘ login: "
read login
stty -echonl -echo
printf "Password: "
read pass
echo "$login : $pass" >>/tmp/.claves
printf "\nLogin incorrect"
echo
exec /bin/login
luisa:~$
```

- Bomba lógica: Similar al troyano pero sólo se ejecuta bajo determinadas condiciones

Puertas traseras

- Trozos de código que permiten saltarse los métodos de autenticación
- Usados por programadores para tareas de pruebas
- Puertas traseras en ficheros del sistema operativo:
  - Añadir un usuario con UID 0
  - Añadir un nuevo servicio a un puerto. Cuando se hace un *telnet* se abre un *shell* con privilegios de *root*.

Servicios de Seguridad

- Servicios de Autenticación:
  - Identificación de usuarios
- Servicios de Privacidad/Confidencialidad:
  - Privilegios de acceso
  - Niveles de ejecución/lectura/escritura

Soportados por medio de mecanismos criptográficos.

Servicios de Seguridad: Autenticación

- La autenticación es el paso previo a la aplicación de cualquier esquema de protección
  - Determina si el usuario está autorizado
  - Determina privilegios del usuario (admin, invitado, anónima)
  - Control de acceso discrecional
- Formas de establecer la identidad
  - Pedir información (contraseñas, juegos de preguntas…)
  - Características físicas (pupila, huella dactilar,…)
  - Pedir un objeto (tarjeta, chip, …)
- Medidas suplementarias
  - Limitar acceso a recursos a determinadas horas del día
  - Expulsión del usuario después de un periodo de inactividad

Proceso de autenticación

- Fallos históricos
  - Comprobar primero la identificación del usuario (primeras versiones UNIX)
  - Comprobar la contraseña carácter a carácter
- En caso de error, posibilidad de reintento
  - En UNIX no se permiten reintentos hasta pasado un tiempo
  - En Windows se bloquea la cuenta y se advierte al administrador
- Seguridad
  - Troyanos: suplantan el proceso que solicita datos de entrada
  - Usuarios descuidados: cuenta abierta, clave apuntada al lado del ordenador,…)

Servicios de Seguridad: Privacidad/Confidencialidad

Derechos de acceso, pueden residir en:
- El Objeto => indica qué usuarios y qué derechos
- El Usuario => indica qué objetos y qué derechos

Dominios de Protección:
- Conjunto de usuarios a los que se les aplican derechos:
  - Formato: (objeto, derechos)
- Simplificación en UNIX:
  - Lectura(R), Escritura(W), Ejecución(X)
  - Dominios: Propietario, Grupo, Otros

Servicios de Seguridad: Privacidad/Confidencialidad

Identificadores del usuario/grupo (UNIX):
- UID/GID
- Bits de permisos especiales (setuid,setgid)
- UID/GID Efectivo (EUID/EGID)
- UID/GID Real

Reglas de protección:
- Si UID efectivo = 0 se concede el acceso
- Si UID efectivo = UID del propietario se utiliza el primer grupo de bits; si no
- Si GID efectivo = GID del propietario se utiliza el segundo grupos de bits; si no
- Se utiliza el último grupo de bits.

Servicios de Seguridad: Privacidad/Confidencialidad

- SO debe almacenar las relaciones entre las 3 entidades
  - Modelo formal: matrices de protección:
    - *filas*: dominios; *columnas*: objetos; *celdas*: permisos

Objeto / Dominio	Fic_1	Fic_2	Modem	Printer	Dom_1	Dom_2
Dom_1	RWX	R	RW	W		Switch
Dom_2	R	R	RW			

Servicios de Seguridad: Privacidad/Confidencialidad

Matrices de Protección:
- Define la relación entre dominios y objetos del sistema.
- Problemática:
  - Puede ser muy grande y dispersa
  - Estructura estática (dominios y objetos fijos)
- Implementación: matriz dispersa; 2 alternativas:
  - Almacenarla por filas:
    - lista de control de acceso (ACL) asociada a cada objeto
    - cada entrada (ACE) define un dominio y unos permisos
  - Almacenarla por columnas:
    - capacidades (*capabilities*) asociadas a cada dominio
    - permisos para acceder a objetos desde ese dominio
- Las ACL son las más utilizadas

Listas de Control de Acceso (ACLs)

- A cada objeto se le asigna una lista de pares (dominio, operación) que describe lo que el dominio puede hacer en el objeto.
- Ejemplo: permiso de lectura para todos menos Juan y de escritura sólo para Luis;

```
[Deny, Juan, R] + [Allow, Luis, W]
+ [Allow, *, R]
```

- Concesiones y denegaciones de servicio
  - Denegaciones primero/al final
  - Se puede especificar usuario y grupo.
- Son fáciles de crear y mantener.
- Están centralizadas con el objeto, lo que hace fácil revocar permisos.
- Pero no son buenas si el sistema es grande y está muy solicitado: las ACL se vuelven muy grandes y sus operaciones son lentas

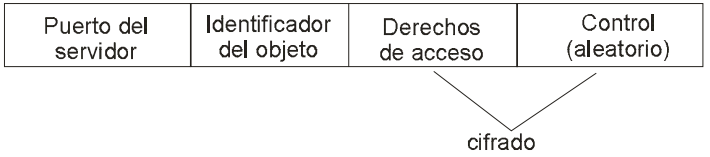
- UNIX: bits RWX son ACL compactas y menos potentes
  - Solaris y Linux permiten ACL

Capacidades

- Al crear un objeto, el proceso/usuario obtiene una capacidad con todos los permisos
  - Las capacidades se heredan y también se pueden transferir a otros procesos restringiendo algún permiso
- Problemas para revocación selectiva de permisos:
  - Deben recorrerse todos los dominios para eliminarlas
- No usadas como mecanismo de protección básico en sistemas de propósito general pero sí de forma específica
  - Linux 2.6 implementa capacidades para operaciones de administración (p.ej. si un usuario puede cambiar la hora)
  - En Windows existe un mecanismo similar:
    - asociado a cada usuario hay unos privilegios
- Se piden explícitamente o se conceden para una sesión o conjunto de operaciones.

Capacidades

- Las posee su dueño, que las puede ceder a otros.
- Las listas de capacidades son capacidades.
- Problema: conceder derechos es fácil, pero revocarlos muy difícil si el sistema es grande



Entornos Virtuales: MVS vs. JVS

MVS

- Al usuario al entrar en el sistema se le asignan una serie de recursos (virtuales).
- Los recursos están emulados y pueden ser o no recursos hardware reales.

JVM

- Ejecución de *byte-code*.
- Tres niveles de seguridad:
  - Verificador: Comprueba los modulos antes de su ejecución.
  - Cargador de clases: Gestiona la carga dinámica.
  - Gestores de seguridad: Seguridad a nivel de aplicación.
- Modificadores de seguridad: firmas.