

# Seguridad en SD 9

**Sistemas Operativos y  
Distribuidos**

**Prof. Javier Echaiz  
D.C.I.C. – U.N.S.**

**<http://cs.uns.edu.ar/~jechaiz>**

**[je@cs.uns.edu.ar](mailto:je@cs.uns.edu.ar)**



# Conceptos de Protección y Seguridad

- **Protección:** Evitar que se haga un uso indebido de los recursos que están dentro del ámbito del SO. Mecanismos y políticas que aseguren que los usuarios sólo acceden a sus propios recursos (archivos, zonas de memoria, etc.)
- **Seguridad:** Es un concepto mucho más amplio y está dirigida a cuatro requisitos básicos:
  - *Confidencialidad:* La información sólo es accesible por las partes autorizadas
  - *Integridad:* Los contenidos sólo podrán modificarse por las partes autorizadas
  - *Disponibilidad:* Los componentes de un sistema informático sólo están disponibles por las partes autorizadas
  - *Autenticación:* Capacidad de verificar la identidad de los usuarios

# Tipos de Peligros

- **Interrupción:** Se destruye un componente del sistema o se encuentra no disponible o utilizable (ataque a la disponibilidad)
  - Destrucción de disco duro, eliminación del sistema gestor de ficheros
- **Intercepción:** Ataque contra la confidencialidad
  - Escucha de canal de comunicaciones, copia ilícita de programas
- **Modificación:** Capacidad de modificar un componente (ataque hacia la integridad)
  - Alterar un programa, modificar el contenido de los mensajes
- **Fabricación:** un elemento no autorizado inserta objetos extraños en un sistema (ataque contra la autenticación)
  - Inclusión de un registro en un fichero, inserción de mensajes en una red

# Problemas de Seguridad

<b>Elemento</b>	<b>Confidencialidad</b>	<b>Integridad</b>	<b>Disponibilidad</b>
<b>Hardware</b>	Robado Copiado	Destruido Pinchado Falsificado	Sobrecargado Fallido Robado Destruido No disponible
<b>Software</b>	Robado Copiado Pirateado	Caballo de Troya Modificado Falsificado	Borrado Mal instalado Caducado
<b>Datos</b>	Descubiertos Espiadados Inferidos	Dañados Error HW Error SW Error usuario	Borrados Mal instalados Destruidos

# Ataques en líneas de comunicación

- *Ataques pasivos*: espionaje o monitorización de transmisiones
  - Lectura del contenido de los mensajes
  - Análisis de tráfico (cifrado)
- *Ataques activos*: modificaciones o creaciones de flujos de datos
  - Enmascaramiento: Un elemento se hace pasar por otro diferente
  - Reenvío: Captura de una unidad de datos y su posterior retransmisión
  - Modificación de mensajes: Se altera parte de un mensaje válido
  - Denegación de servicio: Previene o imposibilita el uso normal o la gestión de las instalaciones de comunicación

# Problemas de Seguridad

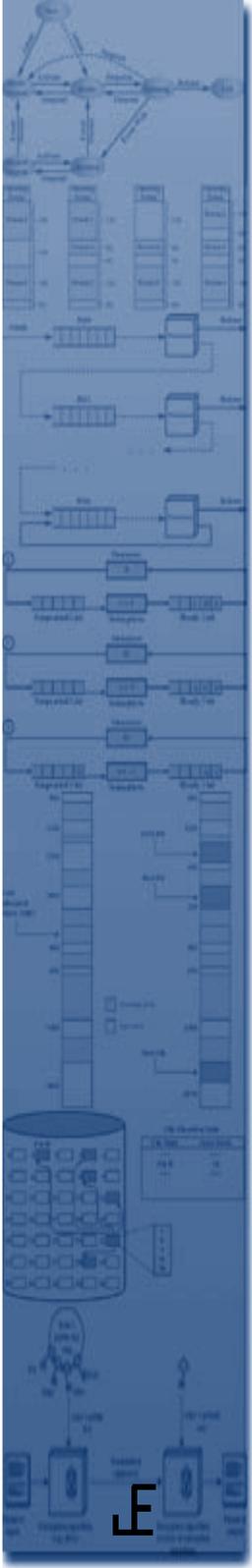
## (I)

- Uso indebido o malicioso de programas
  - *Caballo de Troya (Trojanos)*: Programa que hace cosas no autorizadas
    - Programa de *login* modificado, editor de texto que hace copias a otros directorios
  - *Puerta trasera*: Crear un agujero de seguridad al sistema a través de un programa privilegiado
    - Identificación reservada a un programa, parches a un compilador para que añada código no autorizado
- Usuarios inexpertos o descuidados
- Usuarios no autorizados (intrusos)
- Virus: Es un pequeño programa capaz de reproducirse a sí mismo, infectando cualquier tipo de archivo ejecutable, sin conocimiento del usuario

# Problemas de Seguridad

## (II)

- Gusanos: Es un código maligno cuya principal misión es reenviarse a sí mismo.
  - No afectan a la información de los sitios que contagian
  - Consumen amplios recursos de los sistemas y los usan para infectar a otros equipos
- Rompedores de sistemas de protección: Programas que tratan de romper la seguridad para ejecutar accesos ilegales (Satan)
- Ataques de denegación de servicio: Bombardeo masivo con peticiones de servicio. Los atacantes se enmascaran con identidades de otros usuarios (*spoofing*)
- *Phising*: Es la capacidad de duplicar una página web para hacer creer al visitante que se encuentra en la página original en lugar de en la copiada



# Cifrado

# Cifrado

- Cifrado de la información con clave  $K_C$ :
  - $C(\text{Info}, K_C) \rightarrow \text{InfoCif.}$
  - Notación típica:  $\{\text{Info}\}_{K_C}$
- Descifrado con clave  $K_D$ :
  - $D(\text{InfoCif}, K_D) \rightarrow \text{Info. original}$
- 2 alternativas:
  - sistemas de clave secreta
  - sistemas de clave pública

# Tipos de Sistemas de Cifrado

- Sistemas de clave secreta:
  - $K_C = K_D$
  - Emisor y receptor comparten la clave
  - Ejemplo: DES
  - Problema de la distribución de la clave
- Sistemas de clave pública:
  - $K_C \neq K_D$
  - $K_D$  es la clave secreta del servidor
  - $K_C$  es la clave pública del servidor
  - $K_D$  no puede deducirse a partir de  $K_C$
  - Menos eficientes que sistemas de clave secreta
  - Ejemplo: RSA

# Kerberos

- Sistemas de autenticación desarrollado en M.I.T. (mediados 80)
  - Basado en sistemas de clave secreta
- Usado en muchos sistemas; AFS, RPC de Sun, Windows 2000
- Autenticación de cliente y servidor basado en tercer componente
  - Reside en máquina “segura”
  - Conoce claves secretas de clientes y servidores

# Kerberos

## Características:

- Claves de clientes y servidores no se transmiten por la red
- Claves no se almacenan mucho tiempo en clientes
- *Timestamps* para detectar retransmisiones maliciosas
- Claves con plazo de expiración
- Permite autenticación entre dominios
- Puede usarse también para cifrar los datos

# *Tickets y Autenticadores*

- Objetos básicos en la autenticación
- *Ticket*:
  - Registro que el cliente incluye en el mensaje que permite a un servidor verificar su identidad
  - Está cifrado con clave del servidor e incluye entre otros:
    - identidad del cliente, clave para la sesión, plazo de expiración
- *Autenticador*:
  - Registro que el cliente incluye en el mensaje que asegura que el mensaje se ha generado recientemente y no se ha modificado
  - Está cifrado con clave de sesión e incluye entre otros:
    - tiempo actual y *checksum*

# Componentes

- 2 componentes implementados normalmente como una entidad
- Servidor de autenticación (AS):
  - Proporciona el ticket inicial:
    - Ticket de concesión de tickets (TGT)
  - Normalmente se solicita TGT en el *login* del usuario
  - El TGT se usa para comunicarse con el TGS
- Servidor de concesión de tickets (TGS)
  - Cuando un cliente necesita comunicarse con un nuevo servidor solicita al TGS, usando un TGT, un ticket para identificarse
- AS serviría para obtener tickets para identificarse ante cualquier servidor pero normalmente sólo se usa para comunicarse con TGS

# Protocolo con AS

- Secuencia de mensajes (C cliente; S servidor):
  - C→AS:
    - {C,S,...} sin cifrar
  - AS genera una clave de sesión aleatoria  $K_{SES}$  y envía mensaje:
    - $\{K_{SES}, S, T_{expiración}, \dots\} K_C$
    - $\{Ticket\} K_S$  que contiene  $\{K_{SES}, C, T_{expiración}, \dots\} K_S$
  - C→S: envía mensaje con ticket y autenticador:
    - $\{Ticket\} K_S + \{marca\ de\ tiempo, checksum, \dots\} K_{SES}$
    - Servidor comprueba que no se ha modificado mensaje y que la marca de tiempo es reciente
  - S→C:
    - $\{marca\ de\ tiempo\} K_{SES}$
- Si se precisa integridad de datos pueden cifrarse con  $K_{SES}$

# Necesidad del TGS

- AS es suficiente pero genera un problema de seguridad:
  - Clave del cliente (generada normalmente a partir de contraseña) disponible todo el tiempo en nodo cliente
  - O se pide contraseña periódicamente a usuario o se almacena en la máquina
  - Ambas soluciones inadecuadas
- Solución: Uso de TGS
  - En *login* se obtiene de AS el ticket para TGS (TGT)
  - *Tickets* para nuevos servidores se piden a TGS
  - TGT es diferente en cada *login* y tiene una vida corta

# Protocolo Completo

- Protocolo en *login* (ya analizado):
  - C solicita a AS ticket para comunicarse con TGS (TGT)
  - AS devuelve  $\{TGT\}K_{TGS}$  y clave de sesión con TGS ( $K_{SESTGS}$ )
- Protocolo con TGS: C quiere autenticación con nuevo servidor S
  - C→TGS: solicita ticket para identificarse ante S
    - $\{TGT\}K_{TGS} + \{\text{marca de tiempo, checksum, ...}\} K_{SESTGS} + \{S\}$
  - TGS→C envía mensaje con clave de sesión y ticket:
    - $\{K_{SES}, S, \text{Texpiración, ...}\}K_{SESTGS} + \{\text{Ticket}\} K_S$
  - C→S: envía mensaje con ticket y autenticador:
    - $\{\text{Ticket}\} K_S + \{\text{marca de tiempo, checksum, ...}\} K_{SES}$
  - S→C:
    - $\{\text{marca de tiempo}\} K_{SES}$

**Coming  
Next**

**Web  
Services**

