Día de software libre

Seguridad en sistemas GNU/Linux

Roberto Gómez Cárdenas

rogomez@campus.cem.itesm.mx

http://campus.cem.itesm.mx/dia/ac/rogomez

¿Es Linux seguro?

> date	> original site	> archive	> attacked by	> OS	> comments	> nmap	> class-C
14/05/2002	www.vicavia.de	mirror	hax0rs lab	Linux	none	view	history
14/05/2002	www.usfam.com	mirror	Script-Kiddie-Crew	IRIX	none	view	history
14/05/2002	www.pinkolby.info	mirror	Script-Kiddie-Crew	Linux	none	view	history
14/05/2002	www.mp3-blues.org	mirror	Data ChaOs	Linux	none	view	history
14/05/2002	www.irf.org.tw	mirror	Virtual Hell	Windows	none	view	none
14/05/2002	www.autopia.com	mirror	Virtual Hell	Windows	none	view	history
14/05/2002	www.dub-beautiful.org	mirror	Virtual Hell	Windows	none	view	history
14/05/2002	www.artnovella.com.ar	mirror	CyberCrime	Linux	none	view	history
14/05/2002	www.haryobasuki.com	mirror	AntiHiddenLine	FreeBSD	none	view	history
14/05/2002	www.thebucket.org	mirror	Data ChaOs	Linux	Massdefacement	view	history
14/05/2002	www.triarchy.com.tw	mirror	hax0rs lab	Linux	none	view	none
14/05/2002	www.akm.de	mirror	hax0rs lab	Linux	none	view	history
14/05/2002	www.trade-telecom.de	mirror	hax0rs lab	Linux	none	view	none
14/05/2002	www.goldenfuture24.de	mirror	Rooting Sabotage Forced	Linux	none	view	none
14/05/2002	mail.alibamicrosystema.com	mirror	Unknown	Windows	none	view	none
14/05/2002	gpayment.ceberbj.com.cn	mirror	shazam	Solaris	none	view	history
14/05/2002	mail.cast.org.cn	mirror	shazam	Solaris	none	view	history
14/05/2002	fyzikus.fmph.uniba.sk	mirror	BHS	Linux	none	view	none
14/05/2002	www.ada-forum.de	mirror	BHS	Linux	none	view	none
14/05/2002	www.zanima.net	mirror	BHS	Linux	none	view	none
14/05/2002	www.smue.ch	mirror	Data ChaOs	Linux	none	view	none
14/05/2002	www.julix.de	mirror	Data ChaOs	Linux	none	view	history
14/05/2002	www.transitionsoflife.co.uk	mirror	Otacon	Windows	none	view	history
14/05/2002	www.property.co.il	mirror	nerf	FreeBSD	none	view	history
14/05/2002	www.kevsign-co.com	mirror	CriminalS	Windows	none	view	history
14/05/2002	www.art-in-canada.com	mirror	Otacon	Windows	none	view	none
14/05/2002	www.fabulousfoods.com	mirror	nerf	Linux	none	view	none
14/05/2002	www.andeanoc.org.br	mirror	Otacon	Windows	none	view	history
14/05/2002	www.seram.es	mirror	Evil Angelica	Unknown	none	view	none
14/05/2002	www.kindorko.com	mirror	Otacon	Windows	none	view	history

z C.

Otros sistemas operativos

> OS Statistics for "31948" defaced Websites.

> 18 different OS's since 04/2000		
18584 time(s)	a "Windows"	Host has been defaced, which is 58.17% of all archived defacements
6984 time(s)	a "Linux"	Host has been defaced, which is 21.861% of all archived defacements
2762 time(s)	a "Unknown"	Host has been defaced, which is 8.645% of all archived defacements
1242 time(s)	a "Solaris"	Host has been defaced, which is 3.888% of all archived defacements
876 time(s)	a "FreeBSD"	Host has been defaced, which is 2.742% of all archived defacements
620 time(s)	a "Irix"	Host has been defaced, which is 1.941% of all archived defacements
386 time(s)	a "BSDI"	Host has been defaced, which is 1.208% of all archived defacements
231 time(s)	a "AIX"	Host has been defaced, which is 0.723% of all archived defacements
135 time(s)	a "SCO"	Host has been defaced, which is 0.423% of all archived defacements
31 time(s)	a "NetBSD"	Host has been defaced, which is 0.097% of all archived defacements
27 time(s)	a "HP-UX"	Host has been defaced, which is 0.085% of all archived defacements
24 time(s)	a "OpenBSD"	Host has been defaced, which is 0.075% of all archived defacements
23 time(s)	a "Tru64 UNIX"	Host has been defaced, which is 0.072% of all archived defacements
17 time(s)	a "MacOS"	Host has been defaced, which is 0.053% of all archived defacements
3 time(s)	a "Novell"	Host has been defaced, which is 0.009% of all archived defacements
1 time(s)	a "Ultrix"	Host has been defaced, which is 0.003% of all archived defacements

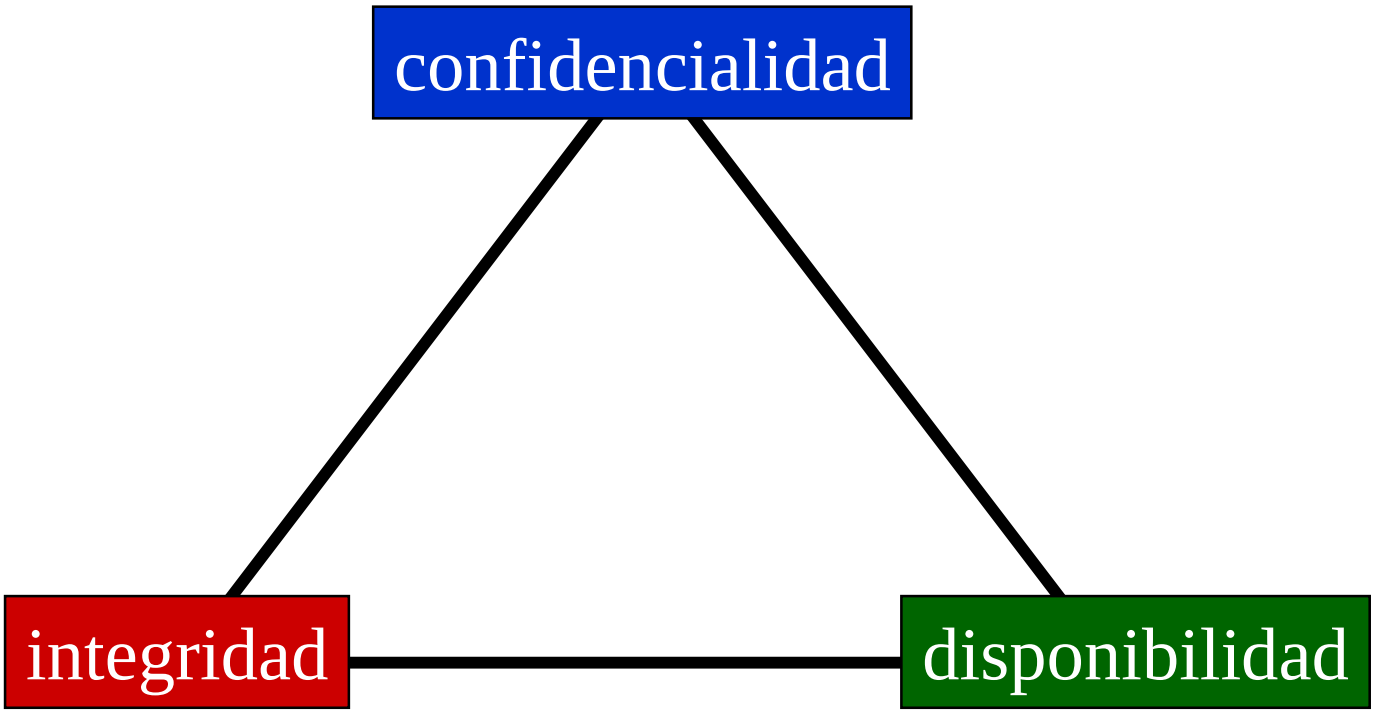
http://www.alldas.org

Seguridad Computacional

El conjunto de políticas y mecanismos que nos permiten garantizar la ***confidencialidad***, la ***integridad*** y la ***disponibilidad*** de los recursos de un sistema.

Roberto Gómez C.

El triangulo de la seguridad



Roberto Gómez C.

¿Qué es un ataque?

- Acción o acciones que tienen por objetivo el que cualquier parte de un sistema de información automatizado, deje de funcionar de acuerdo con su propósito definido.
- Esto incluye cualquier acción que causa la destrucción, modificación o retraso del servicio no autorizado.

Vulnerabilidad

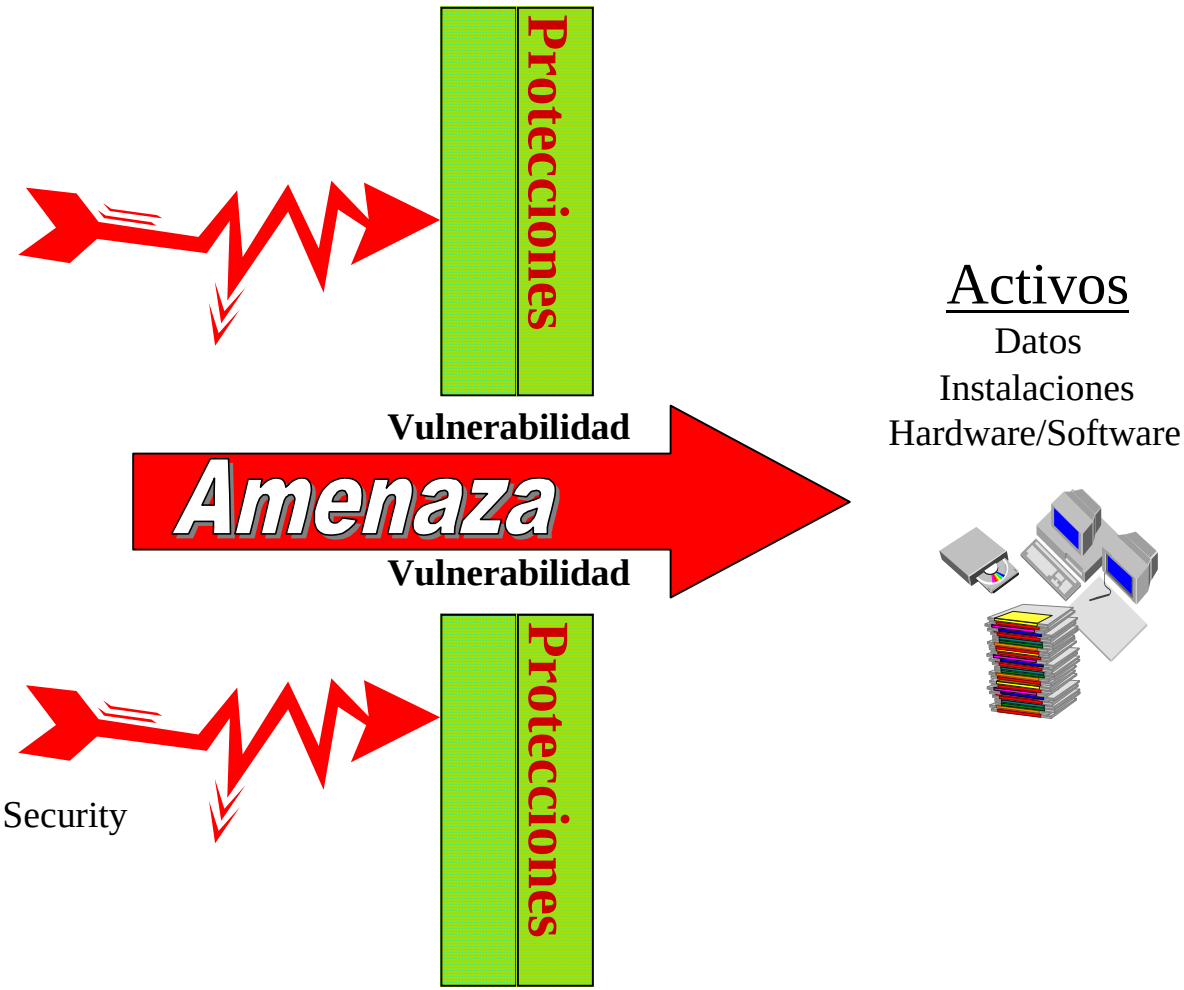
- Debilidad de un sistema informático que permite que sus propiedades de sistema seguro sean violadas.
- La debilidad puede originarse en el diseño, la implementación o en los procedimientos para operar y administrar el sistema.
- En el argot de la seguridad computacional una vulnerabilidad también es conocida como un *hoyo*.

Amenaza

- Circunstancia o evento que puede causar daño violando la confidencialidad, integridad o disponibilidad
- Frecuentemente aprovecha una vulnerabilidad

Roberto Gómez C.

Vulnerabilidad vs amenaza



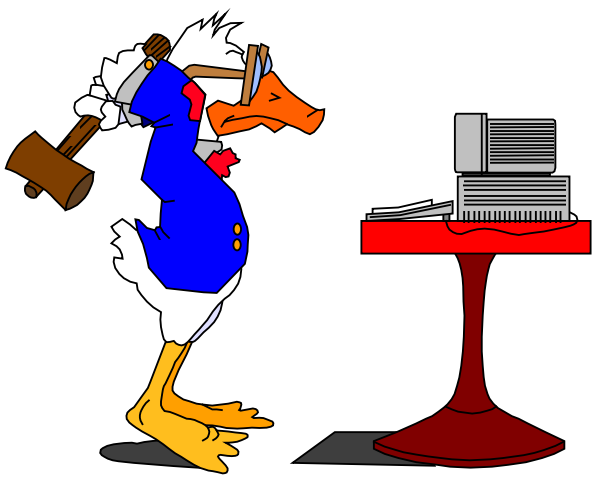
Activos

Datos
Instalaciones
Hardware/Software

Source:
An Introduction to Computer Security
The NIST Handbook
NIST- Serial
Publication 800-12

Aclaración ataque

- No es un ataque físico (aunque puede ser).
- Un ataque no se realiza en un solo paso.
- Depende de los objetivos del atacante.
- Puede consistir de varios pasos antes de llegar a su objetivo.



Roberto Gómez C.

Asegurando el sistema

- Objetivo
  - minimizar los riesgos potenciales de seguridad
- Análisis de riesgos
  - análisis amenazas potenciales que se pueden sufrir,
  - las pérdidas que se pueden generar
  - y la probabilidad de su ocurrencia
- Diseño política de seguridad
  - definir responsabilidades y reglas a seguir para evitar tales amenazas o
  - minimizar sus efectos en caso de que se produzcan
- Implementación
  - usar mecanismos de seguridad para implementar lo anterior

Roberto Gómez C.

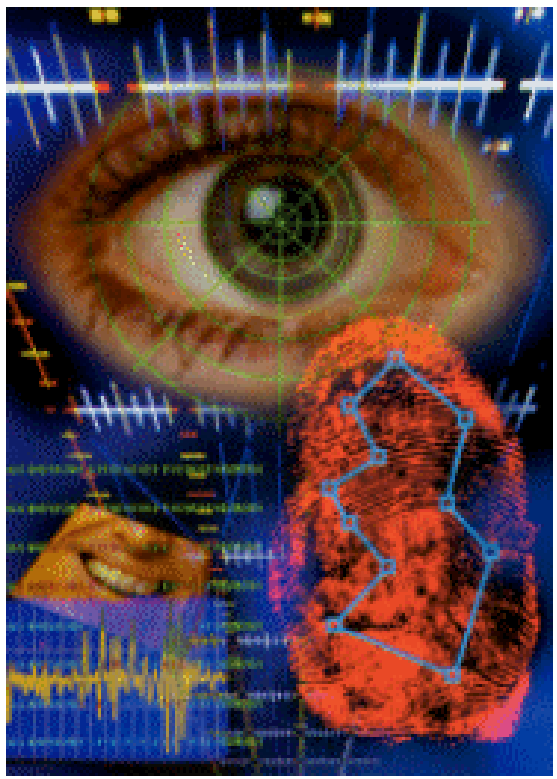
Mecanismos de seguridad

- Son la parte más visible de un sistema de seguridad.
- Se convierten en la herramienta básica para garantizar la protección de los sistemas o de la propia red.
- Se dividen en:
  - prevención
  - detección
  - recuperación

Roberto Gómez C.

Mecanismos prevención

- Aumentan la seguridad de un sistema durante el funcionamiento normal de éste.
- Previenen la ocurrencia de violaciones a la seguridad
- Ejemplos mecanismos:
  - encripción durante la transmisión de datos
  - passwords difíciles
  - firewalls
  - biométricos

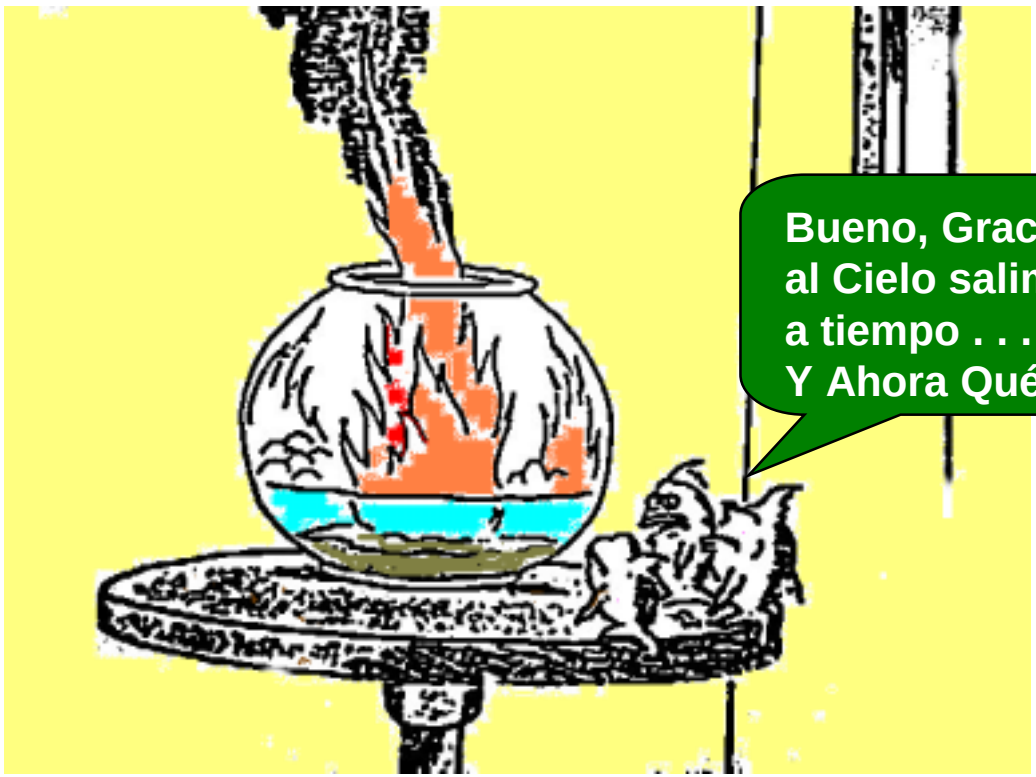


Mecanismos detección

- Son aquellos que se utilizan para detectar violaciones de la seguridad o intentos de violación.
- Ejemplos de estos mecanismos
  - IDS
    - Tripwire
    - Snort
  - Detectores de vulnerabilidades
    - Nessus
    - ISS

Mecanismos de recuperación

- Son aquellos que se aplican cuando una violación del sistema se ha detectado, para retornar a éste su funcionamiento correcto.
- Ejemplos
  - respaldos
  - redundancia
  - BCP
  - DRP
- Subgrupo
  - mecanismos de análisis forense



Mecanismos prevención más habituales en Unix

- Mecanismos de autenticación
  - hacen posible identificar entidades del sistema de una forma única
  - posteriormente, una vez identificadas, son autenticadas (comprobar que la entidad es quién dice ser)
- Mecanismos de control de acceso
  - usados para proteger objetos del sistema (archivos, recursos..)
  - controlan todos los tipos de acceso sobre el objeto por parte de cualquier entidad del sistema
  - dos tipos discrecional (DAC) y mandatorio/obligatorio (MAC)

Mecanismos prevención …

- Mecanismos de separación
  - permiten separar los objetos dentro de cada nivel
  - evitar el flujo de información entre objetos y entidades de diferentes niveles siempre que no exista una autorización expresa del mecanismo de control de acceso
  - se dividen en cinco grandes grupos, en función de como separan a los objetos
    - separación física
    - separación temporal
    - separación lógica (o aislamiento)
    - separación criptográfica
    - fragmentación

Mecanismos prevención …

- Mecanismos de seguridad en las comunicaciones
  - protegen la integridad y privacidad de los datos cuando se transmiten a través de la red
  - la mayor parte se basan en la criptografía
    - encripción de llave pública y de llave privada
    - firmas digitales
  - uso de protocolos seguros

Mecanismos autenticación

- Basados en algo que se sabe
  - passwords, frases y números de identificación personal, NIP
  - siguen siendo el sistema de autenticación más usado hoy en día.
- Basados en algo que se es
  - biométricas y comportamiento
  - se realiza una medición física y se compora con un perfil almacenado con anterioridad,
- Basadas en algo que se tiene
  - usar un objeto físico que llevan consigo y que de alguna forma comprueba la identidad del portador
  - tokens, tarjetas inteligentes y pases.

Basados en lo se tiene

8 bits (6805), 16 bits and
32 bits RISC (ARM 7)

128 Bytes to
1 KB

CPU

RAM

ROM

2KB to 64 KB

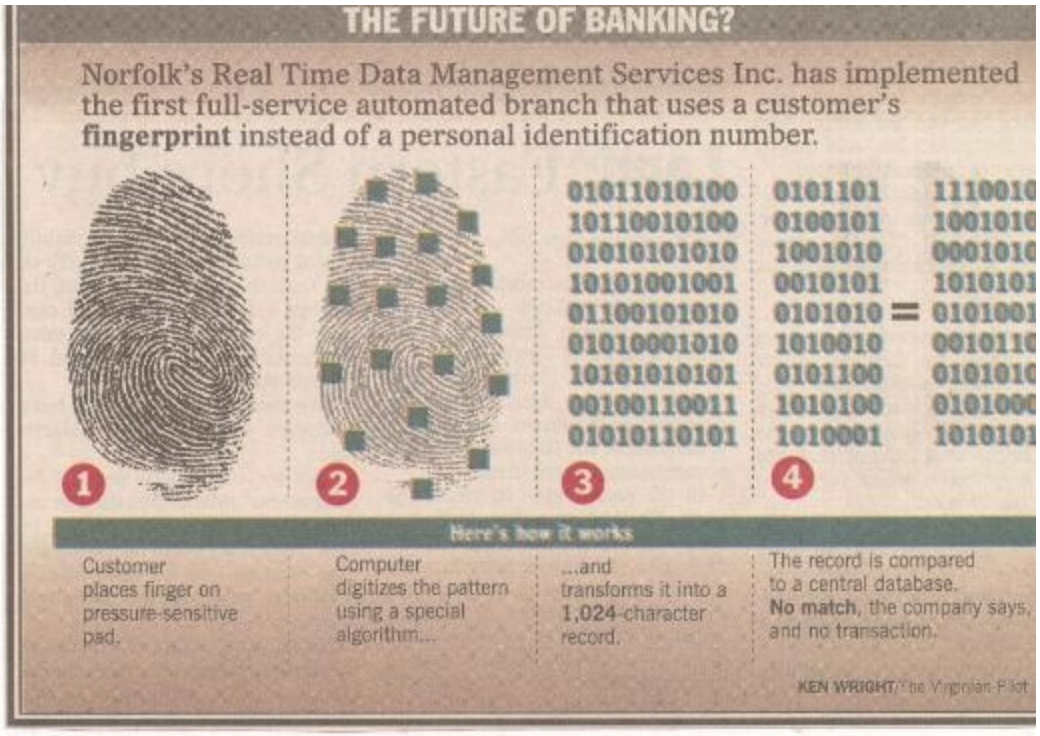
Persistent
Memory

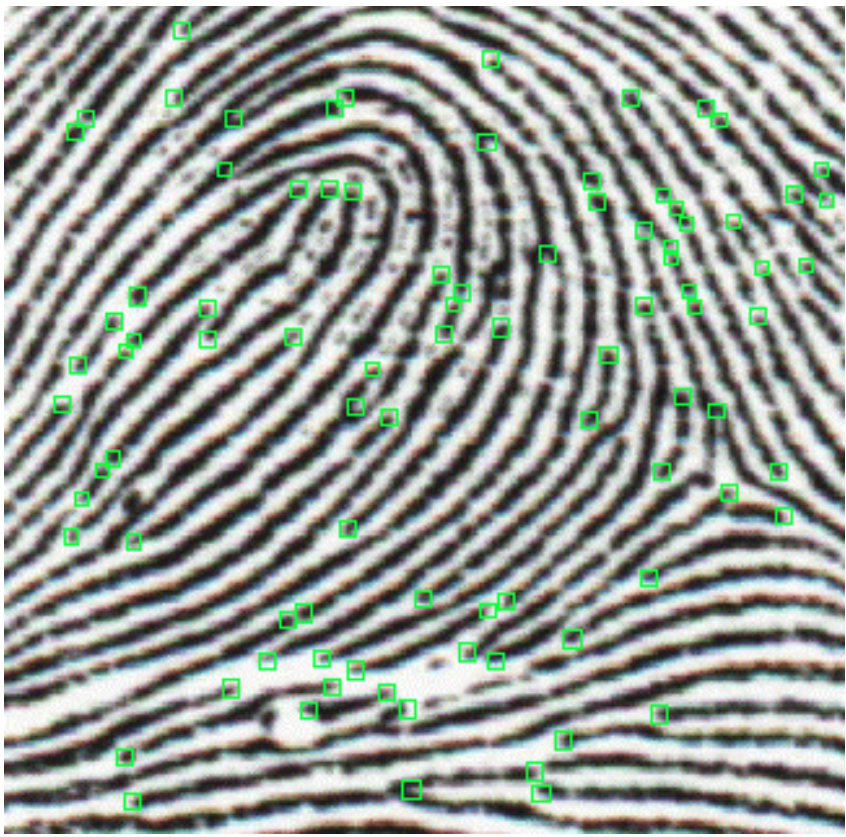
Input/
Output

Security
Block

Maximum 64 KB

Basdos en lo que és





Los passwords de Linux

- Archivo ASCII manipulable con un editor
- Debe poder ser leído por todos los usuarios para ciertos comandos
- A cada usuario le corresponde una entrada
- Los programas realizan una búsqueda secuencial de las entradas (no vale la pena ordenar las entradas)
- Los campos de cada entrada están separados por carácter de dos puntos (:)
- Se almacenan en archivo /etc/passwd

Ejemplo archivo /etc/passwd

etaboada:ypK2awu1hBqGs:1326:41:Eunice Taboada:/home/dacs/etaboada:/bin/csh
dgonzale:dU8MloKM7Af8Y:10106:41:John Derick Lucien:/home/dacs/dgonzale:/bin/csh
abermude:Fe5/I/SHg53HM:2404:43:Adriana Diaz B:/home/prepa/abermude:/bin/csh
sa448020:iqC7X.6SUEASE:1832:215:David Bernal:/home/sap/sa448020:/bin/csh
rcaballe:j3KODtAuQ8uEQ:8773:41:Ricardo Caballero:/home/dacs/rcaballe:/bin/csh
csanchez:YYoHlXDeYHanM:1212:43:Concepcion Sanchez:/home/prepa/csanchez:/bin/csh
sduenas:1ube95PeMQZOQ:10140:41:Sergio Francisco:/home/dacs/sduenas:/bin/csh
rperrin:rKWggQip3DIHQ:10021:44:Rafael Fausto:/home/dae/rperrin:/bin/csh
gperrin:Bj87cqMfSXzmc:10012:44:Graciela Patricia:/home/dae/gperrin:/bin/csh
rvilla:4McraxhY8AVB6:8839:43:Rafael Villa:/home/prepa/rvilla:/bin/csh
lvelio:IifTeZS98v/H.:1248:41:Lucrecia Velio-mejia:/home/dacs/lvelio:/bin/csh
tpacheco:UNbyYZ.dNCY3.:10275:510:Tito Omar:/home/dia/tpacheco:/bin/csh
marmoral:iNTdxOLbUnz1o:1448:45:Marisol

Archivo de passwords Shadow

- Los passwords encriptados no son almacenados en el archivo *passwd* sino en un archivo llamado *shadow*
- En el archivo de passwords, se reemplaza la contraseña por una ‘x’, lo cual le indica al sistema que verifique dicha contraseña contra el archivo shadow
- Al no tener acceso a los passwords encriptados, un atacante tendrá mayor oposición para descifrar un password.
- Desafortunadamente, en ocasiones se olvida prevenir que un archivo shadow sea públicamente accesible.

Ejemplo /etc/shadow

```
root:lLOTWOUA.YC2.o:107113:0::7:7::
bin:*:10713:0::7:7::
daemon:*:10713:0::7:7::
adm:*:10713:0::7:7::
lp:*:10713:0::7:7::
sync:*:10713:0::7:7::
shutdown:*:10713:0::7:7::
halt:*:10713:0::7:7::
mail:*:10713:0::7:7::
news:*:10713:0::7:7::
uucp:*:10713:0::7:7::
operator:*:10713:0::7:7::
games:*:10713:0::7:7::
gopher:*:10713:0::7:7::
ftp:*:10713:0::7:7::
man:*:10713:0::7:7::
majordom:*:10713:0::7:7::
```

¿Y qué tengo que hacer?

- Cambiar el archivo de passwords
  - tarea relativamente fácil
- Asegurarse que todos los programas tienen soporte para passwords con shadow
  - recompilar todos los programas para que verificasen la contraseña (login, ftpd, etc, etc)
  - si la distribución soporta PAM (p.e. Red Hat), autentificación todo lo que se necesita es añadir un módulo PAM que lo entienda y editar el fichero de configuración para cualquier programa (digamos el login) permitiéndole que use ese módulo para hacer la autentificación
  - algunas distribuciones incorporan dicha opción durante la instalación

Unas palabras sobre los módulos PAM

- PAM: Puggable Authentication Modules
- Permiten alterar la forma en la que aplicaciones Linux realizan autenticación sin reescribir estas ni compilarlas.
- En distribuciones recientes, PAM ha sido integrado en los procedimientos de login y en otros procedimientos que requieren de autenticación
- Propociona opciones para administración de autenticación, de cuentas de sesión y de passwords.

Problemas seguridad archivo /etc/password

- Cuentas sin contraseña.
  - intruso que conozca la cuenta puede introducirse sin necesidad de password.
- Cuentas con passwords fáciles.
  - mismo problema punto anterior
- Cuentas predeterminadas.
  - cuentas con passwords determinados en algunos sistemas, inrtrusos las pueden usar
- Cuentas que ejecutan un solo comando.
  - en lugar de lanzar un shell ejecutan un comando
- Permisos

Problemas seguridad archivo /etc/password

- Permisos
  - archivo de lectura pública para todos los usuarios, pero sólo el superusuario debe poder modificar su contenido.
  - el propietario debe ser el super-usuario.
- UID repetidos
  - sistema no identifica por el nombre de cuenta, sino por el número de usuario (UID).
  - si dos usuarios comparten el mismo UID el sistema los tratará como si fueran el mismo usurario.
  - todos usaurios deben tener un UID diferente, si es necesario duplicar dos o más, sería mejor crear un grupo.

Resumiendo

- Si el sistema lo soporta usar un archivo de passwords tipo shadow.
- Poner fechas de expiración en las cuentas.
- Controlar las cuentas de los invitados.
- Eliminar cuentas compartidas.
- Establecer reglas y políticas para operar como super-usuario.
- Si es posible activar opción MD5 para encripción de passwords

Mecanismos de control de acceso

- La autenticación pretende establecer quién eres.
- La autorización (o control de accesos) establece qué puedes hacer con el sistema.
- Dos modelos: DAC y MAC
- Control de acceso discrecional (DAC),
  - un usuario bien identificado (típicamente, el creador o 'propietario' del recurso) decide cómo protegerlo estableciendo cómo compartirlo, mediante controles de acceso impuestos por el sistema.
- Control acceso mandatorio (MAC)
  - es el sistema quién protege los recursos.
  - todo recurso del sistema, y todo usuario tiene una etiqueta de seguridad.

¿Y en Linux?

- Objeto
  - recursos que el usuario utiliza,
  - puede ser un archivo, impresora, etc
- Usuario
  - no siempre es un individuo
  - es una entidad que puede ejecutar un programa o guardar un programa, conectarse al sistema, editar archivos, lanzar programas y usar el sistema de forma clásica
  - usuario son reconocidos por su número de identificacación (user id = uid)

Tipos de usuarios

Nombre	*UID*	*Versión*	*Rol*
root	0	todas	superusuario
daemon	1	todas	usuario ficticio demonios
bin	2	Sistema V	propietario de /bin y de /usr/bin
	3	BSD	
sys	3	Sistema V	usuario del sistema
adm	4	Sistema V	propietario archivos
	3	BSD	contabilidad
uucp	5	Sistema V	usuario por conexión
	66	BSD 4.3	distante fuera internet
lp	71	Sistema V	administrador del spooler de impresión
	9	HP-UX	

El superusuario o root

- Es el primer usuario del sistema
- Tiene por número de usuario 0 y número de grupo 0
- Tiene todos los accesos posibles dentro del sistema:

  creación y sobretodo *destrucción*
- Es el único usuario conectado cuando el sistema esta en mono-usuario

Operaciones restringidas a root

- Montar y desmontar sistemas de archivos
- Cambiar el directorio root de un proceso
- Crear archivos de tipo periférico
- Ajustar el tiempo del reloj
- Cambiar propietarios de los archivos, (en algunos sistemas)
- Utilizar hasta el límite los recursos del sistema
- Asignar el nombre del host del sistema
- Interfaz de configuración de red
- Dar de baja el sistema

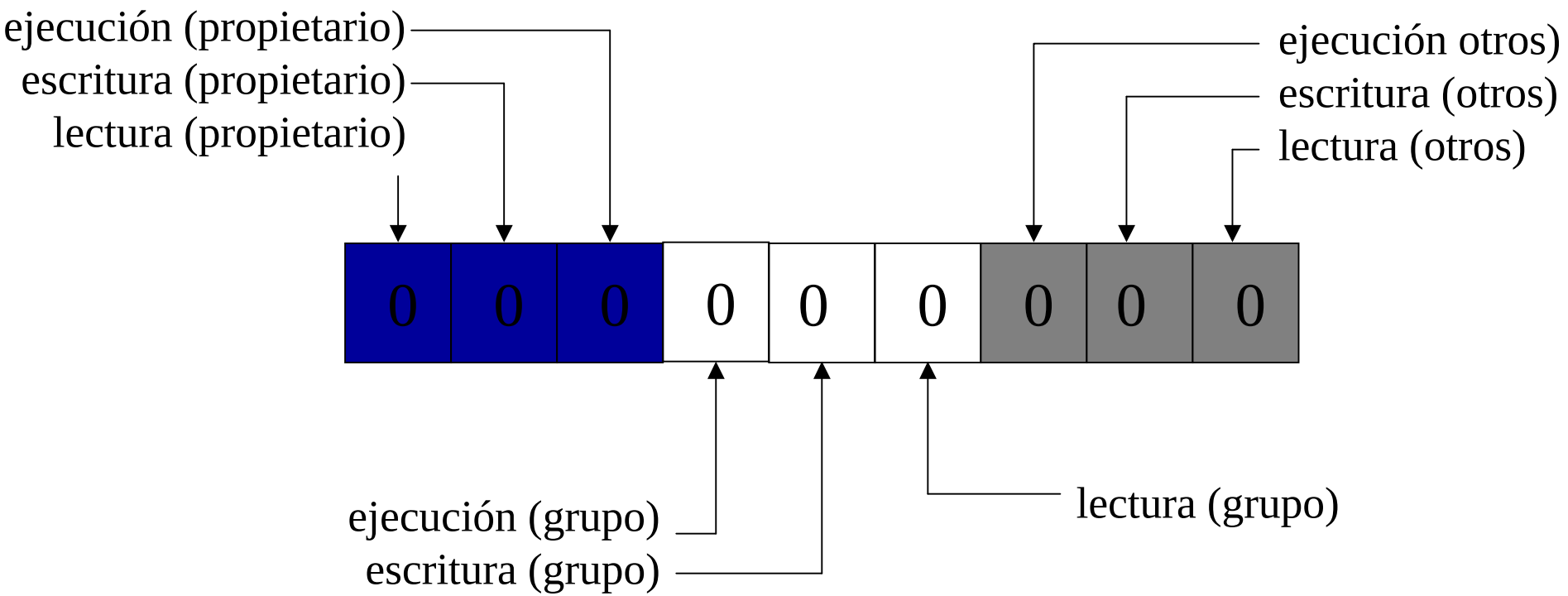
El comando sudo

- Los privilegios de super-user no pueden subdividirse:
  - problemas para hacer respaldos, (debe hacerse como root),
  - sin darle la libre funcionalidad de todo el sistema.
- Solución: comando sudo.
  - toma como argumento un comando entero que será ejecutado como super-usuario.
  - sudo verifica archivo */usr/local/adm/sudoers* que es una lista de la gente autorizada a usar sudo y que programas pueden usar.
  - si el comando a ejecutar es permitido sudo pide el password del usuario y ejecuta el programa como root.

  - después se pueden ejecutar comandos durante un período sin tener que reteclear el password.
- También guarda un historial de todos los comandos que fueron manejados, la gente que los solicitó y la hora a la que se ejecutaron.

Los permisos en Linux

- Archivos cuentan con permisos, el significado varia un poco entre archivos y directorios



Significado permisos en directorios

- **r:**
  - autorización de leer el directorio (comando **ls**)
- **w**:
  - autorización de escribir en el directorio
  - (creación, modificación o supresión de archivos)
- **x**:
  - autorización para posesionarse en el directorio (comando **cd**)

umask: permisos por default

- Comado umask
  - abreviación de user file creation mode mask
- Permite controlar los permisos que por default se le asignan a todo archivo
  - archivos permiso 666 y directorios 777
- Cuando se establece una máscara, se le indica al sistema los permisos que no desea que tenga un archivo, más que los que si desea que tenga.
- El permiso se expresa como un número octal de cuatro digitos.
- Los valores más comunes son 022, 027 y 077

Roberto Gómez C.

Los usuarios y los procesos

- Procesos pertenecen a un solo y único usuario
- El propietario es el que lanzó el proceso
  - puede enviarle señales y, en consecuencia, matarlo
- Para lanzarlo debe poseer los permisos de ejecución del archivo que contiene el código binario

- La “propiedad” del archivo del código no influye en la del proceso
  - usuario toto ejecuta código de un archivo que pertenece a cachafas
  - el proceso pertenece a usuario toto
- Esto es limitativo
  - se desea permitir a un usuario modificar el contenido de un archivo sin darle derecho de escritura en él
  - ejemplo archivo /etc/passwd, un usuario debe poder cambiar su password sin poder modificar el archivo que lo contiene

El bit Set UID (SUID)

- Derecho complementario de un proceso que condiciona la propiedad del proceso que ejecuta su código
- Retomando el ejemplo anterior:
  - si usuario toto activa el bit SUID del archivo
  - el usuario toto es el propietario del archivo, pero el propietario efectivo es cachafas
  - cachafas adquiere los derechos de toto durante el tiempo que dure la ejecución del proceso

Cuidados del bit SUID

- El bit SUID puede representar un hoyo en la seguridad del sistema
- Es necesario minimizar el número de archivos que pertenezcan al super-usuario y que tengan activado el bit SUID
- Algunas versiones de Unix ignoran el bit SUID y SGID en scripts, solo programas compilados pueden tenerlo activo

El bit Set Group ID (SGID)

- Mismo principio que SUID pero para grupos
- Ejecutar un archivo con bit SGID activo asigna el ID de grupo del usuario al mismo que el del archivo ejecutado, durante el tiempo que dura la ejecución de este
- Archivos con SGID o SUID activo pierden sus propiedades especiales cuando son copiados

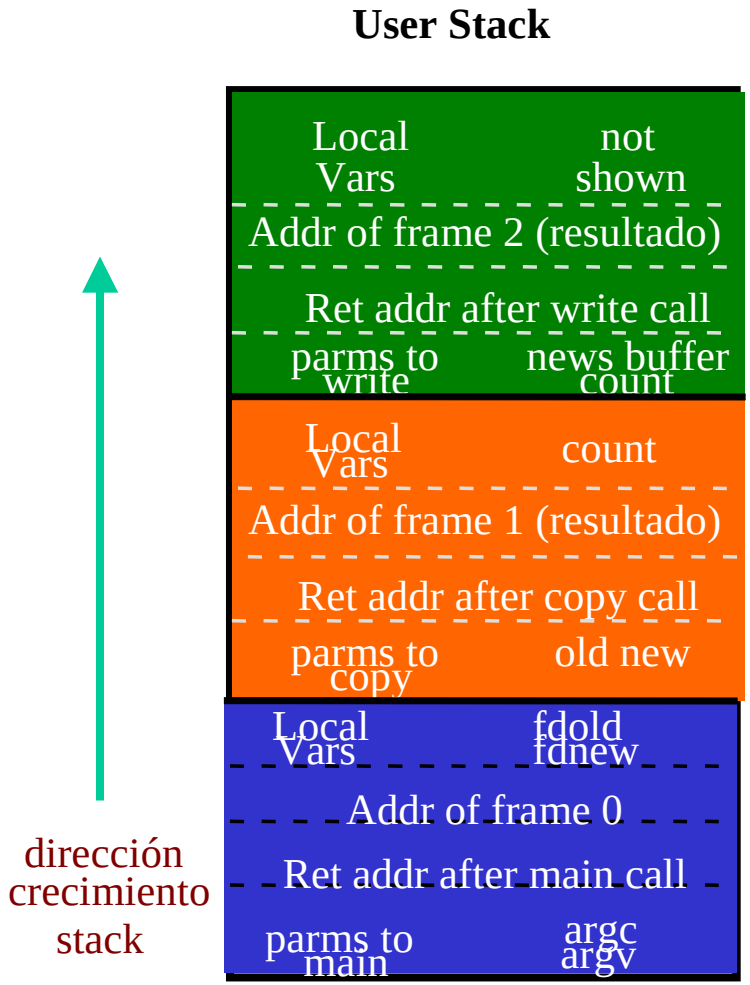
Ejemplo bits SUID y SGID

```
rogomez@armagnac:3>ls -l /usr/bin/passwd  /usr/bin/login
                                  /usr/bin/mailx /etc/passwd
-rw-r--r--   1 root         752 Oct 22  1998 /etc/passwd
-r-sr-xr-x   1 root       29192 Jul 15  1997 /usr/bin/login*
-r-x--s--x   1 bin       127540 Jul 15  1997 /usr/bin/mailx*
-r-sr-sr-x   3 root       96796 Jul 15  1997 /usr/bin/passwd*
rogomez@armagnac:4>
```

Aprovechandose del UID

- Buffer overflow
  - consiste en introducir un parámetro en una función que direccione nuestro programa a una dirección en el segmento, donde hemos introducido datos, como un apuntador a la ejecucion de un shell, añadir campos en el archivo de contraseñas
  - el programa afectado debe contar con el SUID de root

Ejemplo stack




```
copy (int old, int new)
{
        int count;
        while ( (count = read(old, buffer, sizeof(buffer))) > 0 )
          write(new, buffer, count);
}

main(argc, argv)
{
        int fdold, fdnew;
        fdold = open(argv[1], O_RDONLY]);
        fdnew = open(argv[2], 0666);
        copy (fdold, fdnew);
        exit(0);
}
```

Un primer ejemplo

toto@cachafas:1> cat prog1

```
int main(int argv,char **argc) {
      char buf[25];

      strcpy(buf,argc[1]);
 }
```

toto@cachafas:2> gcc prog1.c -o prog1

toto@cachafas:3> prog1 ‘esto es una prueba de un buffer overflow’

????????????????????????????

¿¿qué pasa si en lugar de strcpy() se usa strncpy()??

Un segundo ejemplo

```
void function(int a, int b, int c)
{
   char buffer1[5];
   char buffer2[10];
   int *ret;
   ret = buffer1 + 12;
   (*ret) += 8;
}
void main( ) {
  int x;
  x = 0;
  function(1,2,3);
  x = 1;
  printf("%d\n",x);
}
```

```
toto@cachafas:4> gcc prog2.c -o prog2
toto@cachafas:5> prog2
0
toto@cachafas:6>
```

Mecanismos de separación: firewalls

- Un firewall es una colección de componentes colocados entre dos redes, que en conjunto poseen las siguientes propiedades:
  - todo el tráfico de afuera hacia adentro, y viceversa, debe pasar por el firewall.
  - sólo tráfico autorizado, como establecido previamente en las políticas de la organización, puede pasar a través del firewall
  - algun tipo de tráfico puede requerir de cierto tipo de procesamiento:
    - autenticación usuarios
    - auteticación servicio
    - encripción

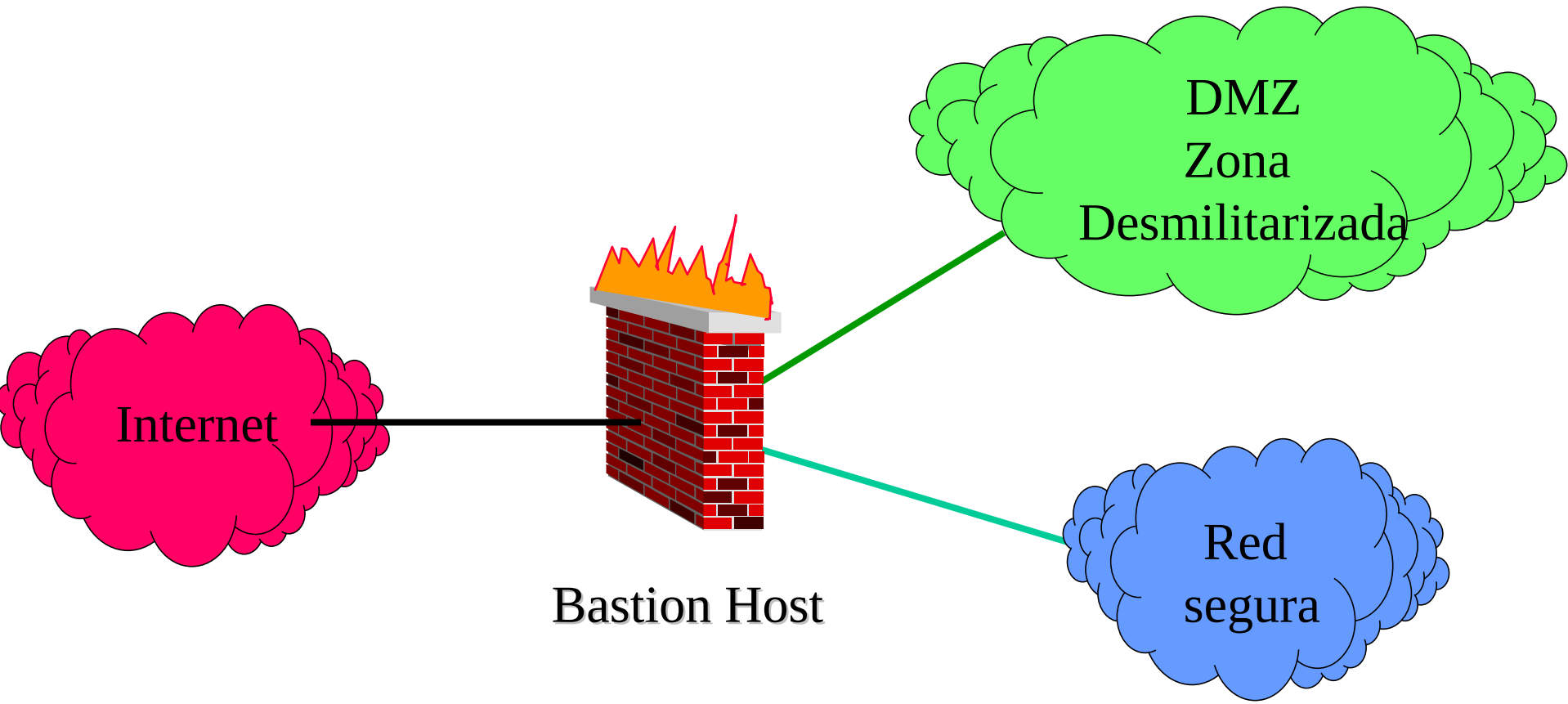
Linux y Firewalls

- Un firewall es una colección de componentes colocados entre dos redes, que en conjunto poseen las siguientes propiedades:
  - todo el tráfico de afuera hacia adentro, y viceversa, debe pasar por el firewall.
  - sólo tráfico autorizado, como establecido previamente en las políticas de la organización, puede pasar a través del firewall.

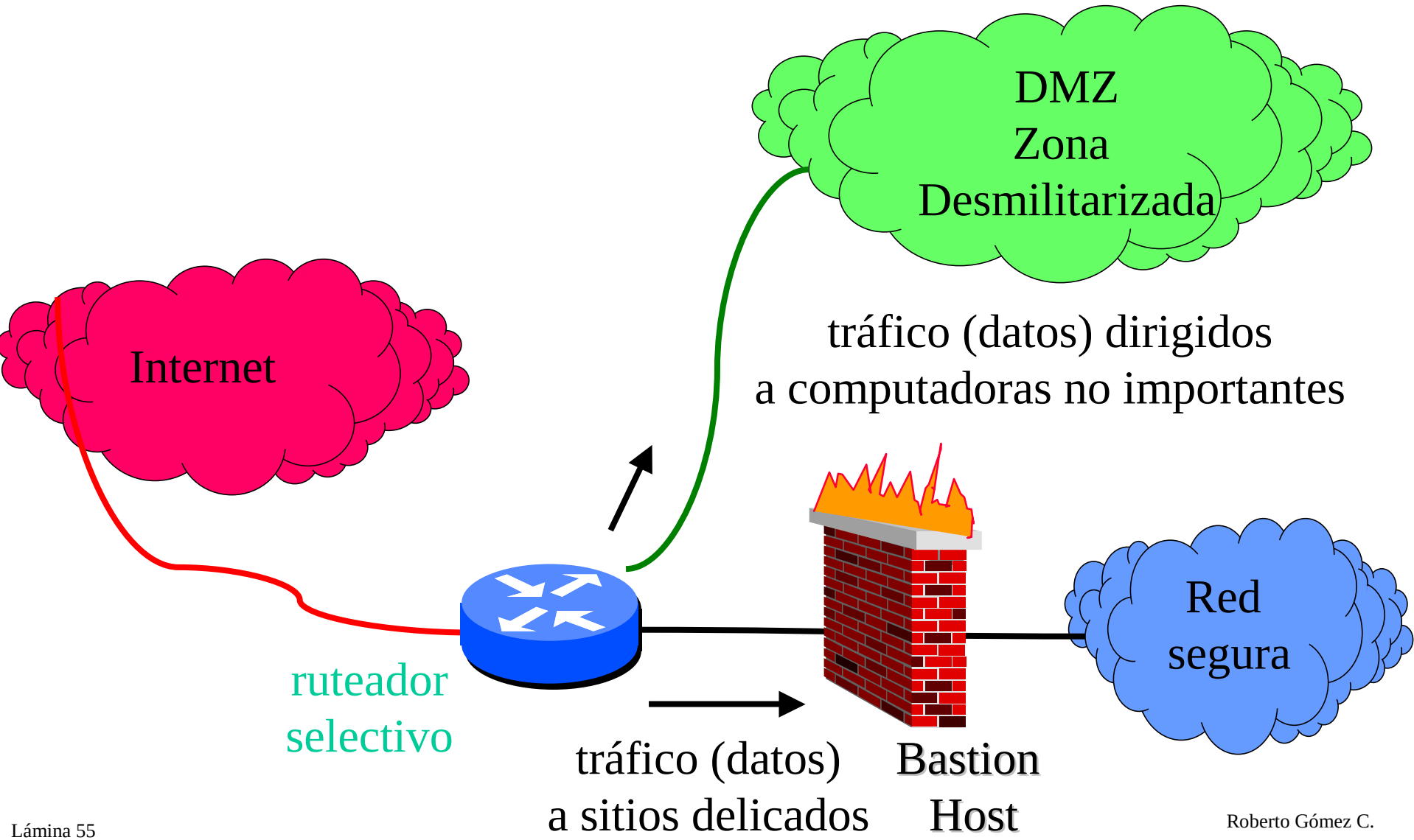
Elementos y configuraciones firewalls

- Ruteador Selectivo
  - puede ser un ruteador comercial con capacidad de filtración de paquetes.
  - bloquea el tráfico entre dos redes o servidores específicos.
- Bastion Host
  - contiene la mayor parte del software del firewall.
- Configuraciones
  - Gateway “Dual-Homed”
  - Screened host gateway

Gateway “Dual-Homed”



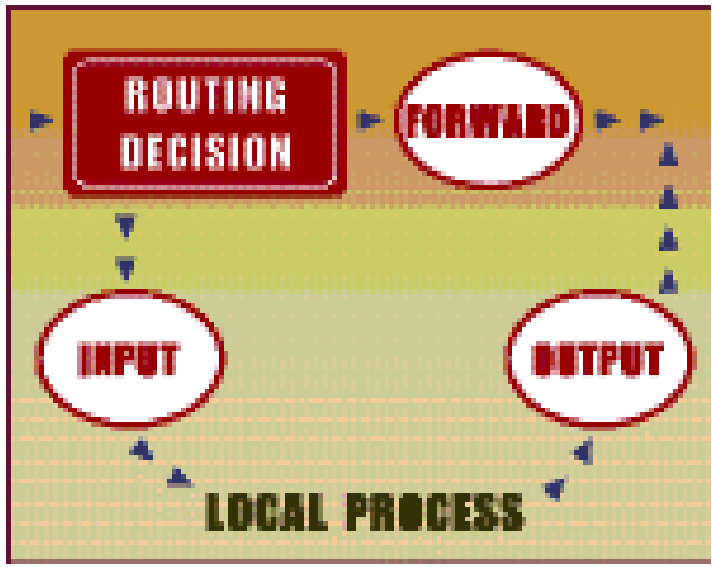
Screened host gateway



Roberto Gómez C.

Los ipchains y las iptables

- La serie 2.2.x del kernel de Linux usaba un sistema de firewalling llamado ipchains.
- Las iptables reemplazan al ipchains en 2.4x debido a cambios internos en los módulos del kernel (netfilter)



- Iptables (tambien conocido como netfilter) nos permite configurar un Firewall de forma que tengamos controlado quien entra, sale y/o enruta a través de una máquina Linux.

Otras opciones de filtrado

- TCPWRAPPER
  - sistema de filtrado que viene en todas las distribuciones
  - este sistema de filtrado es algo mas pobre que un firewall como ipchains, ya que solo permite filtrar por ip y puerto
- Proxies
  - un host con un servidor proxy instalado puede servir como un servidor y un cliente a la vez.
  - los gateway de aplicación escuchan los pedidos de los clientes, cuales serán devueltos a su destino.
  - los gateway de aplicación mantienen dos conexiones separadas entre los sistemas externos y internos.
  - analizan los datos que pasa a través de ellos

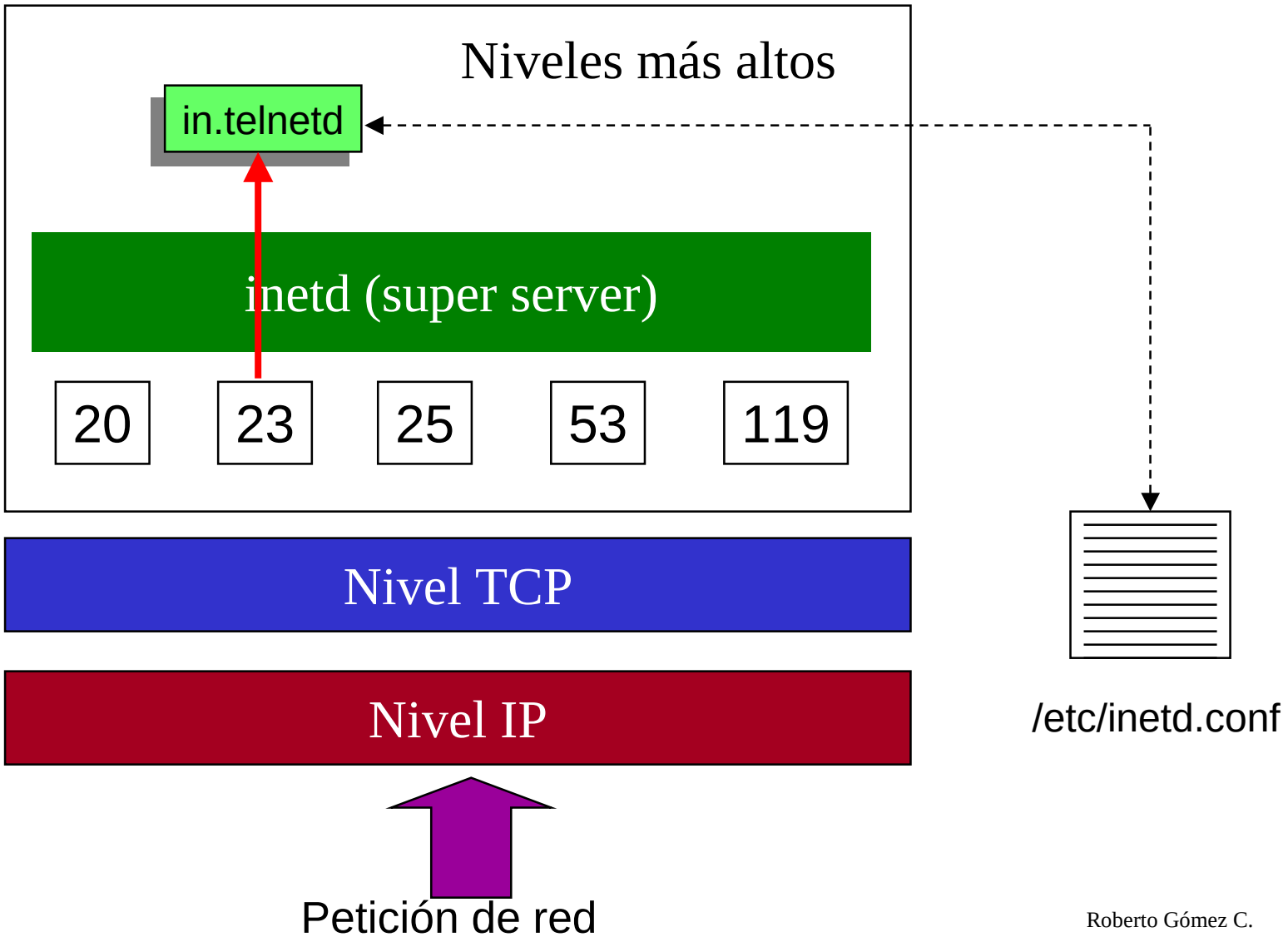
Roberto Gómez C.

Eliminando otros riesgos

- Boot
  - alguien puede resetear el sistema el sistema desde un disco con un backdoor ejecutable
  - configurar el boot para primero bootear del disco duro
- LILO
  - posible bootear en monousuario (INIT 1),
  - salta forma normal autenticación y presenta shell root
  - agregar un password al archivo de configuración LILO
- Eliminando servicios
  - es un SO poderoso que ejecuta varios servicios útiles
  - la mayoría de estos servicios son innecesarios y representan un riego potencial a la seguridad del sistema

Esquema servicios

Ejemplo petición de una sesión Telnet



Mecanismos de seguridad en las comunicaciones

- Criptografia
  - conjunto de técnicas o procedimientos que alteran los símbolos de información sin alterar el contenido,
  - la información es modificada en un conjunto de símbolos sin contenido para las partes que no disponen de las técnicas.
- Criptografia clásica
  - por substitución
  - por transposición
- Criptografía moderna
  - critpografía simétrica (llave privada)
  - criptografia asimétrca (llave pública)

El principio de llave

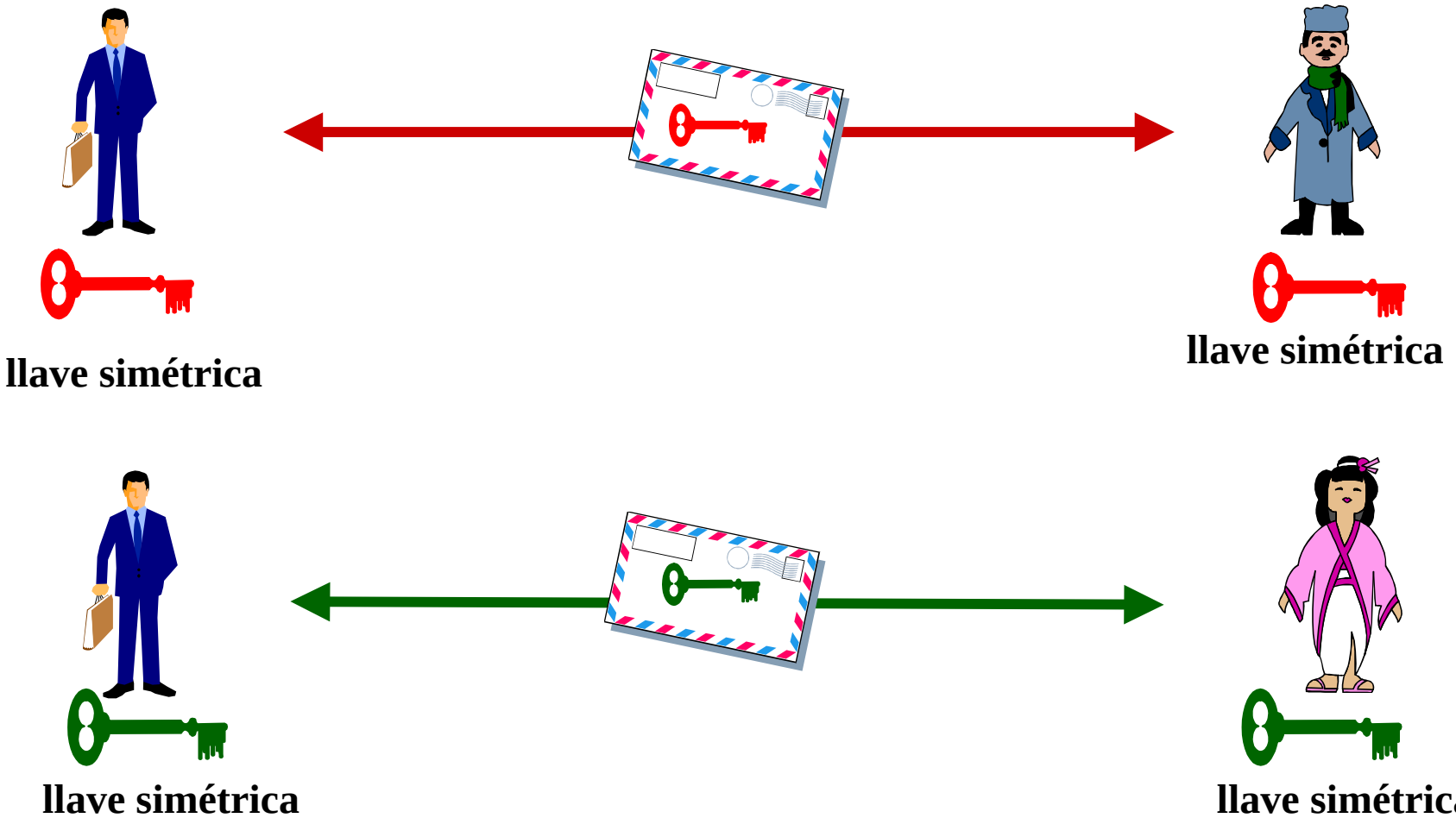
- Elemento fundamental de la criptografia.
- Usado para transformar el texto claro en criptograma
- Por ejemplo supongamos el uso de la llave DAVID.

DAVID = 1000100 1000001 1010110 1001001 1000100

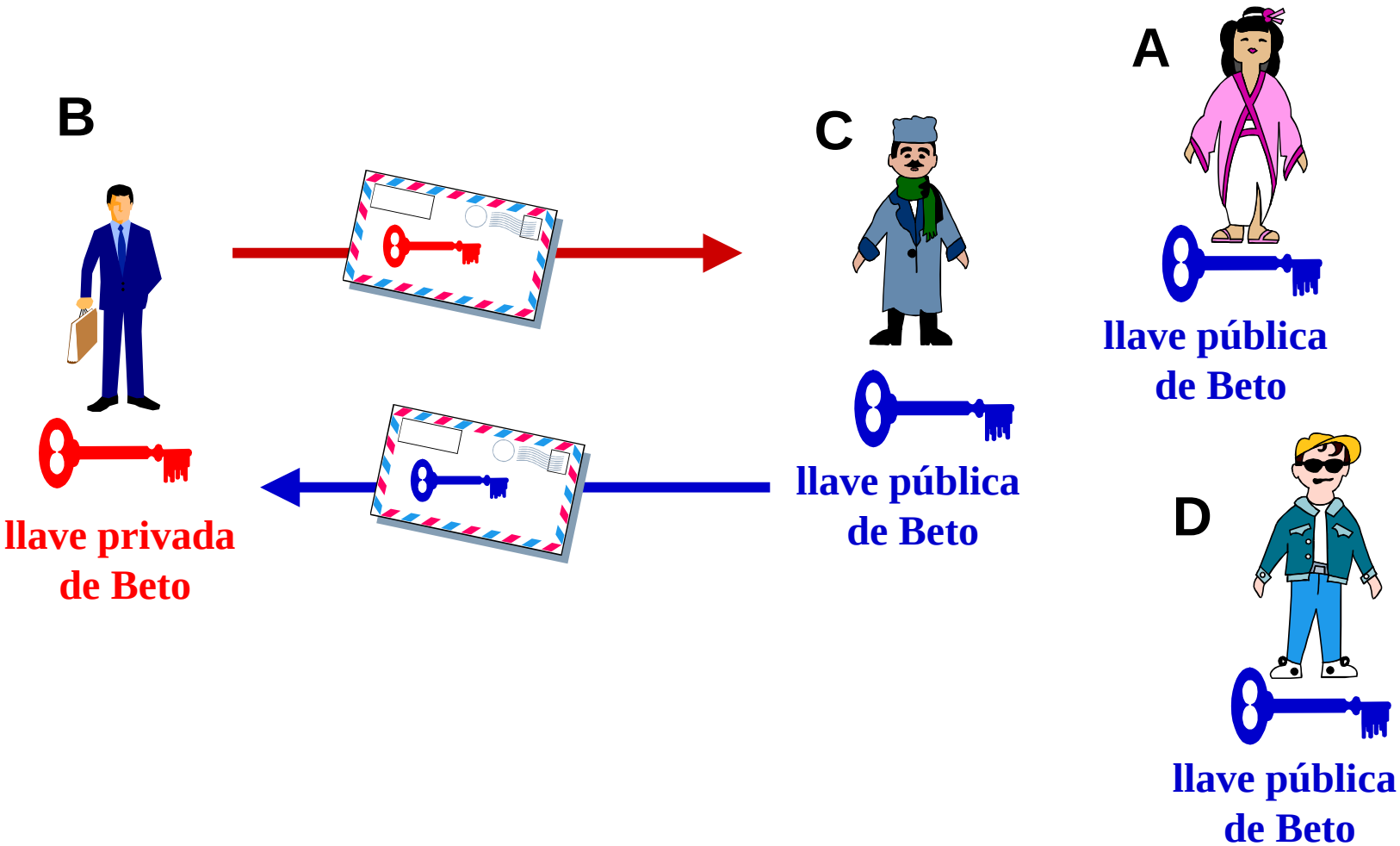
- Para encriptar/decriptar sumamos la llave al mensaje original, (suma binaria: xor)

Texto claro:	**HELLO**
Texto ASCII:	**1001000100010110011001001001111**
Llave:	**1000100100000110101101001011000100**
Criptograma:	**0001100000010000110100000101001011**

Criptografia de llave secreta (simétrica)



Criptografia llave pública (asimétrico)



SSL, PCT y TCL

- Protocolos criptográfico de propósito general para asegurar canales de comunicación bidireccionales
- Se utilizan comúnmente junto con el protocolo TCP/IP
- Sistema encripción usado por navegadores Netscape e Internet Explorer

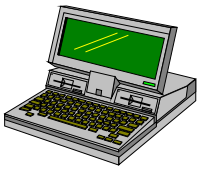
SSL, PCT y TLS

- 1994: SSL V 2.0 (Netscape)

  microsoft descubre un problema en SSL
- 1995: PCT V 1.0
- 1996: SSL V 3.0
- 1997: PCT V 4.

  se decide terminar con la pelea: Microsoft y Netscape deciden sacar un protocolo en común
- 1999: TLS V 1.0

¿Cómo funciona?



Cliente

Servidor

Hola

Hola

Como estas

Muy bien

No hay autenticación ni privacidad, ni encripción

Quiero comprar

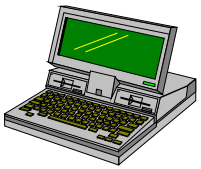
Toma mi llave pública

Te envio una llave nueva encriptada con tu llave pública

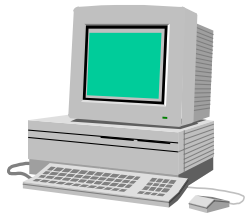
Hablemos en forma segura

Comunicación encriptada con la llave enviada por el cliente

Otro posible escenario



Cliente

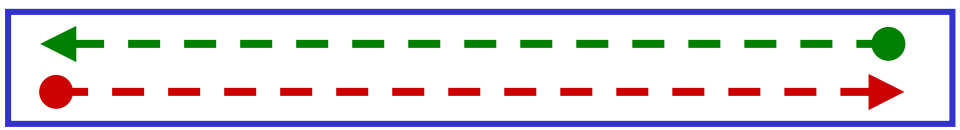


Servidor

Hablemos de forma segura, aquí están los protocolos y criptogramas que manejo

Escogo este protocolo y criptograma. Aquí esta mi llave pública, un certificado digital y un número random

Usando tu llave pública encripte una llave simétrica aleatoria



Comunicación encriptada con la llave enviada por el cliente y un hash para autenticación de mensajes

Otros protocolos

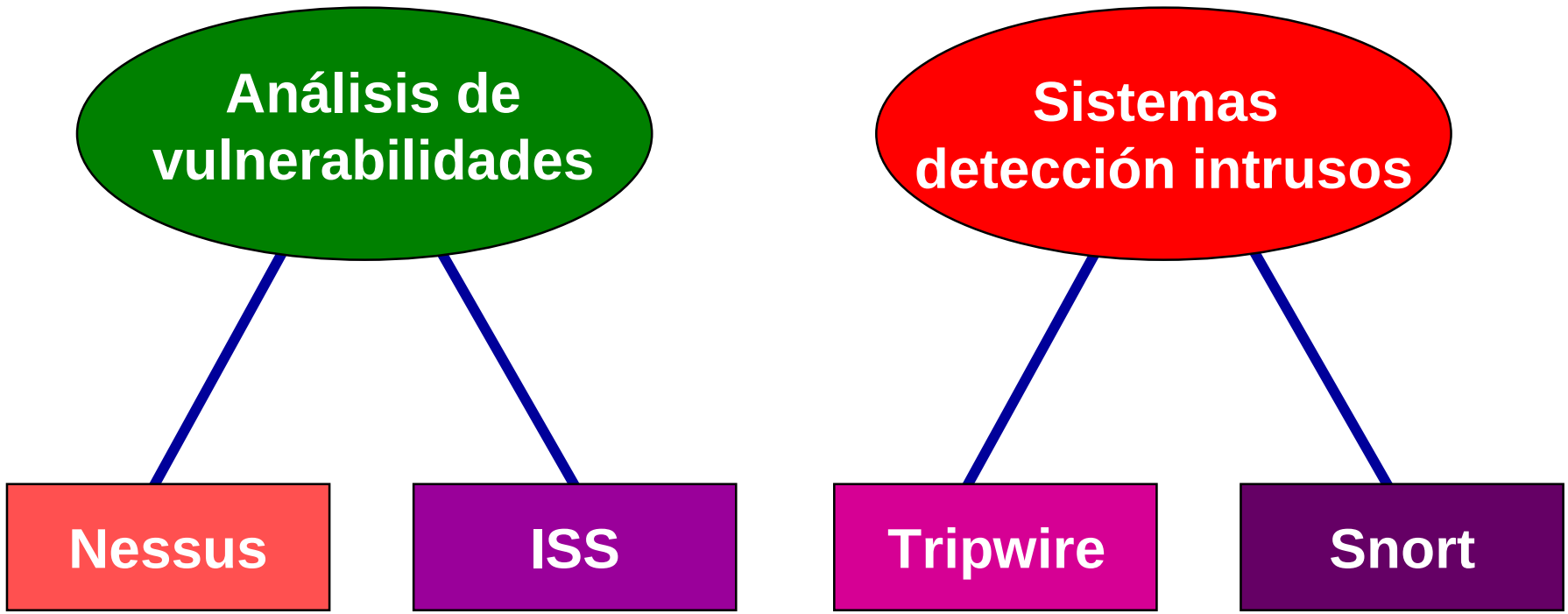
- SSL
- PCT
- TLS
- S-HHTP
- Ipsec e IPv6
- SSH
- PGP

- S/MIME
- iKP
- SET
- CyberCash/CyberCoin
- DNSEC
- Kerberos
- S/Key

Mecanismos detección

- IDS
  - Tripwire
  - Snort
- Detectores de vulnerabilidades
  - Nessus
  - ISS

Mecanismos detección



Los analizadores de vulnerabilidades

- Son herramientas para ayudar a los administradores a auditar sus redes para valorar y/o incrementar el nivel de seguridad
- Encargadas de encontrar de forma automatica vulnerabilidades de los sistemas
- Pequeño problema:
  - también las puede usar el atacante
- Ejemplos
  - Nessus
  - SATAN
  - SAINT
  - nmap

Un ejemplo scanner: nessus

- Escáner remoto de vulnerabilidades y debilidades de sistemas.
- Escrito por Renaud Deraison (a los 18 años, París)
- Actualización en base de plugins
- Incorpora ataques basados en Web
- Distribuido bajo la licencia GNU,
- Open Source, lo cual elimina el riesgo de que ejecute código malicioso.
- Cuenta con su propio lenguaje de programación (NASL, Nessus Atack Scripting Language) optimizado para pruebas de seguridad.

IDS

- Intrusion Detection Systems.
- Busca automatizar la detección y eliminación de intrusos.
- Se define un intento de intrusión a la posibilidad, no autorizada, de
  - acceder información,
  - manipular información,
  - dejar un sistema fuera de alcance o sin posibilidad de uso.

Posibles clasificaciones IDS

- fuente de datos
  - hosts
  - red
- donde se coloca
  - detras firewall
  - enfrente firewall
- parámetros detección
  - reglas
  - esta-dísticas
- momento análisis de datos
  - por lotes
  - en línea

Ejemplos de IDS en Linux

- Tripwire
  - comprobador de integridad para archivos y directorios de sistemas Unix: compara un conjunto de estos objetos con la información sobre los mismos almacenada previamente en una base de datos, y alerta al administrador en caso de que algo haya cambiado
- Snort

  - Sniffer basado en libpcap que puede ser utilizado como NIDS ya que cuenta con capacidad de analizar, a través de reglas, el contenido completo de cada paquete que circula por la red.
  - Utiliza reglas definidas por el administrador de seguridad para buscar patrones y detectar actividad hostil

Mecanismos de recuperación

DRP

BCP

Bitácoras

Respaldos

Computo forense

Los respaldos

- Es una copia de los datos escrita en cinta u otro medio de alamcenamiento duradero.
- De manera rutinaria se recuerda a los usuarios de computadoras que respalden su trabajo con frecuencia.
- Los administradores de sitios pueden tener la responsabilidad de respaldar docenas o incluso de cientos de máquinas

Accountabilty: loggin: bitacoras

- Se refiere al procedimiento a través del cual un sistema operativo registra eventos conforme van ocurriendo y los preserva para un uso posterior.
- Es posible configurar los sistemas de tal forma que los eventos:
  - se escriban en uno o en distintos archivos,
  - se envien a través de la red a otra computadora,
  - se transmitan a algún dipositivo.
- Algunos comandos en linux
  - lastlog
  - last

Otras opciones

- BCP
  - capacidad para mantener la continuidad de las operaciones
  - dirigido a situaciones catastróficas (no problemas rutinarios)
- DRP
  - recuperar la operación de los servicios computacionales y de telecomunicaciones después de un desastre
  - desastre es un evento no planeado que ocasiona la “no disponibilidad” de los servicios informáticos por un periodo de tiempo tal que, para restablecer estos servicios, es necesario utilizar facilidades alternas de cómputo y telecomunicaciones en otra localidad

Otras opciones …

- Computo forense
  - Se refiere al proceso de aplicar técnicas científicas y analíticas a infraestructura de cómputo, para identificar, preservar, analizar y presentar evidencia de manera que sea aceptable en un procedimiento legal
  - se trata de reconstruir que pasó, que lo ocasionó y deslindar responsabilidades

Una última sugerencia

- Actualización de parches.
- Estra pendientes de las últimas vulnerabilidades que se presentan y actuar en consecuencia.
- Hay que asegurarse que se instalen los parches de seguridad recomendados
- Se recomiendo usar una máquina auxiliar para bajar los parches y mantener el nuevo sistema aislado.
- Posibles fuentes:
  - bugtraq@securityfocus.com
  - redhat-watch-list-request@redhat.com

Conclusiones

- Es importante asegurar el sistema operativo
- Es la base de otras aplicaciones y mecanismos
- La seguridad nunca es negra o blanca y el contexto cuenta más que la tecnología.
- No porque un sistema operativo no protega contra granadas de mano, este no sirve
  - solo implica que no podemos deshacernos de nuestras paredes, ventanas y puertas
- Diferentes tecnologías de seguridad tienen lugares importantes en una solución general de seguridad.

Referencias

- Network Intrusión Detection; Northcutt, Ed. New Riders, 2da. edición
- Network Security; Kaufman, Perlman y Speciner, Ed. Prentice Hall
- Applied Cryptography Protocols, Algorithms and Source in C; B. Schneier, John Wiley & Sons
- Maximum Linux Security, Anonymous, SAMS, 2000
- Secure-Programs-HOWTO
- Security-HOWTO

Algunas ligas interesantes

- http://www.securityfocus.com
- http://www.cert.org
- http://www.sans.org
- http://www.kriptograma.org
- http://www.packetstorm.com
- http://www.snort.org
- http://www.tripwire.com
- http://www.linux.org
- http://linux.security.com

Un sistema operativo es tan seguro como su administrador y tan inseguro como incapaz sea el administador.

¡Gracias por su atención!

Día de software libre

Seguridad en sistemas GNU/Linux

Roberto Gómez Cárdenas

rogomez@campus.cem.itesm.mx

http://webdia.cem.itesm.mx/dia/ac/rogomez

La invencibilidad depende de uno mismo; la vulnerabilidad del enemigo, de él.

La invencibilidad reside en la defensa; la posibilidad de la victoria en el ataque.

Sun Tzu, "El arte de la guerra"