

# *Criptografía y Seguridad de Datos*

## **Intrusos y virus**

Carlos Figueira.

Universidad Simón Bolívar

*Basado en láminas del Profesor*

*Henric Johnson (<http://www.its.bth.se/staff/hjo/>)*

*[henric.johnson@bth.se](mailto:henric.johnson@bth.se)*

# Contenido

- Intrusos
  - Técnicas de intrusión
  - Protección de claves de acceso (*password*)
  - Estrategias para selección de claves de acceso
  - Detección de intrusiones
- Virus y amenazas relacionadas
  - Programas maliciosos
  - La naturaleza de los virus
  - Enfoques de antivirus
  - Técnicas avanzadas de antivirus

# Intrusos

- Tres clases de intrusos (*hackers* o *crackers*):
  - **Suplantador:** toma identidad de usuario legítimo
  - **Usuario fraudulento:** usuario legítimo, hace accesos no autorizados o uso fraudulento de sus privilegios
  - **Usuarios clandestinos:** toma id. de supervisor para borrar trazas

# Control de acceso

- El sistema mantiene un archivo que asocia clave de acceso con usuarios autorizados
- Puede ser protegido:
  - Guardando claves de accesos con cifrados de una sola vía (resumen criptográfico)
  - Control de acceso del archivo

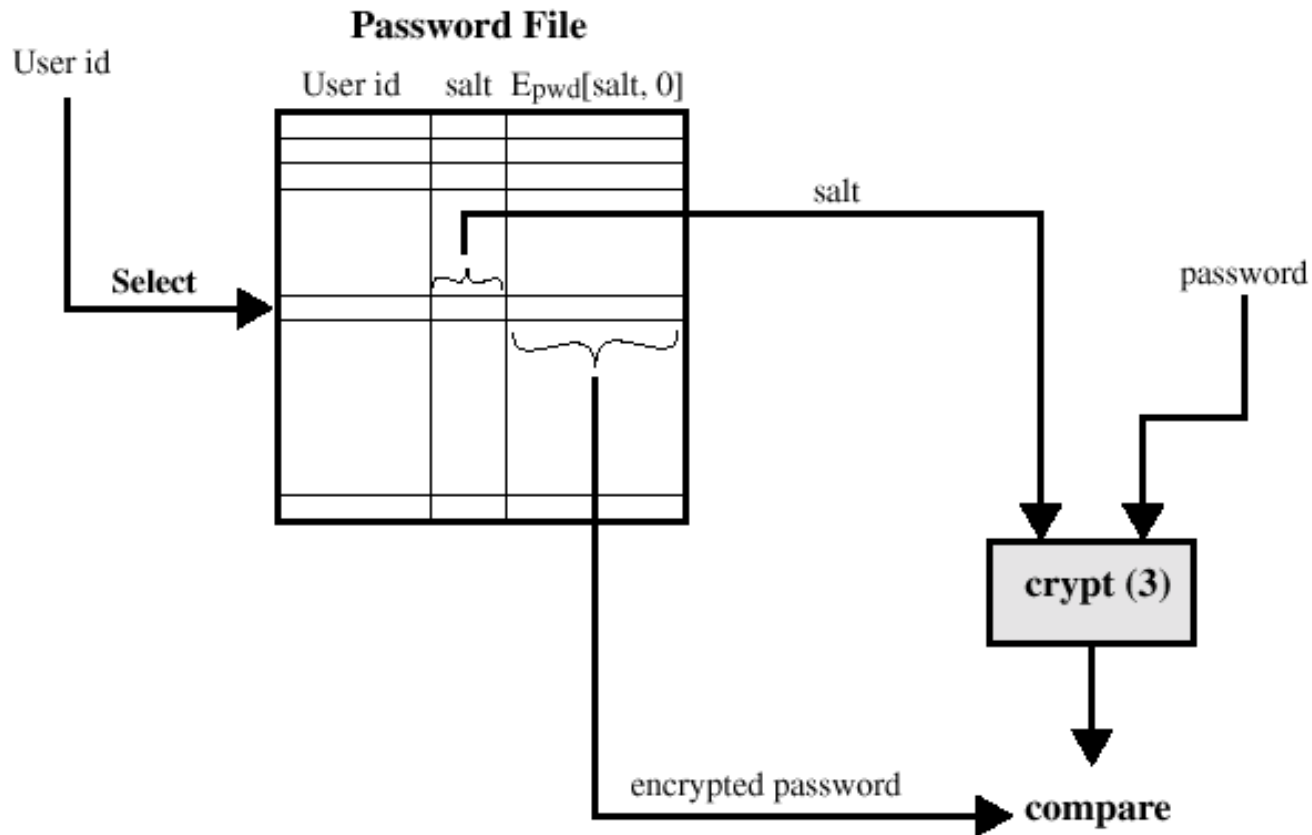
# Técnicas de Intrusión

- Técnicas para averiguar *claves de acceso*:
  - Probar clave de acceso por defecto.
  - Probar todas las palabras cortas: 1-3 car.
  - Todas las palabras de diccionario (60,000).
  - Recoger info del usuario: hobbies, nombre de familiares, fechas de cumpleaños, número de tel., cédula, calle donde vive, etc.
  - Probar todas las placas de carro
  - Usar un Caballo de Troya (ver más adelante)
  - Espiar la línea entre usuario remoto y anfitrión

Prevención: forzar selección de claves de acceso buenos (Ej.:lj4Gf4Se%f#)



# Esquema claves de acceso UNIX



Verificando clave de acceso

# Almacén de *claves de acceso* UNIX

- Se guardaban en un archivo con permisos de lectura para todos `/etc/passwd`.
- Ahora se mantienen en un directorio *“shadow”*, únicamente visible por *“root”*.

# ***“Salt”***

- El uso de la “sal” tiene tres objetivos:
  - Previene contra claves de acceso duplicadas
  - Aumenta la longitud efectiva de la clave de acceso
  - Evita el uso de implementaciones en hardware de DES

# Estrategias de selección de claves

- Educación del usuario
- Claves de acceso generadas automáticamente
- Verificación de claves de acceso:
  - *Reactiva*: administrador corre *cracker*
  - *Proactiva*: se verifica en el momento de crearlo

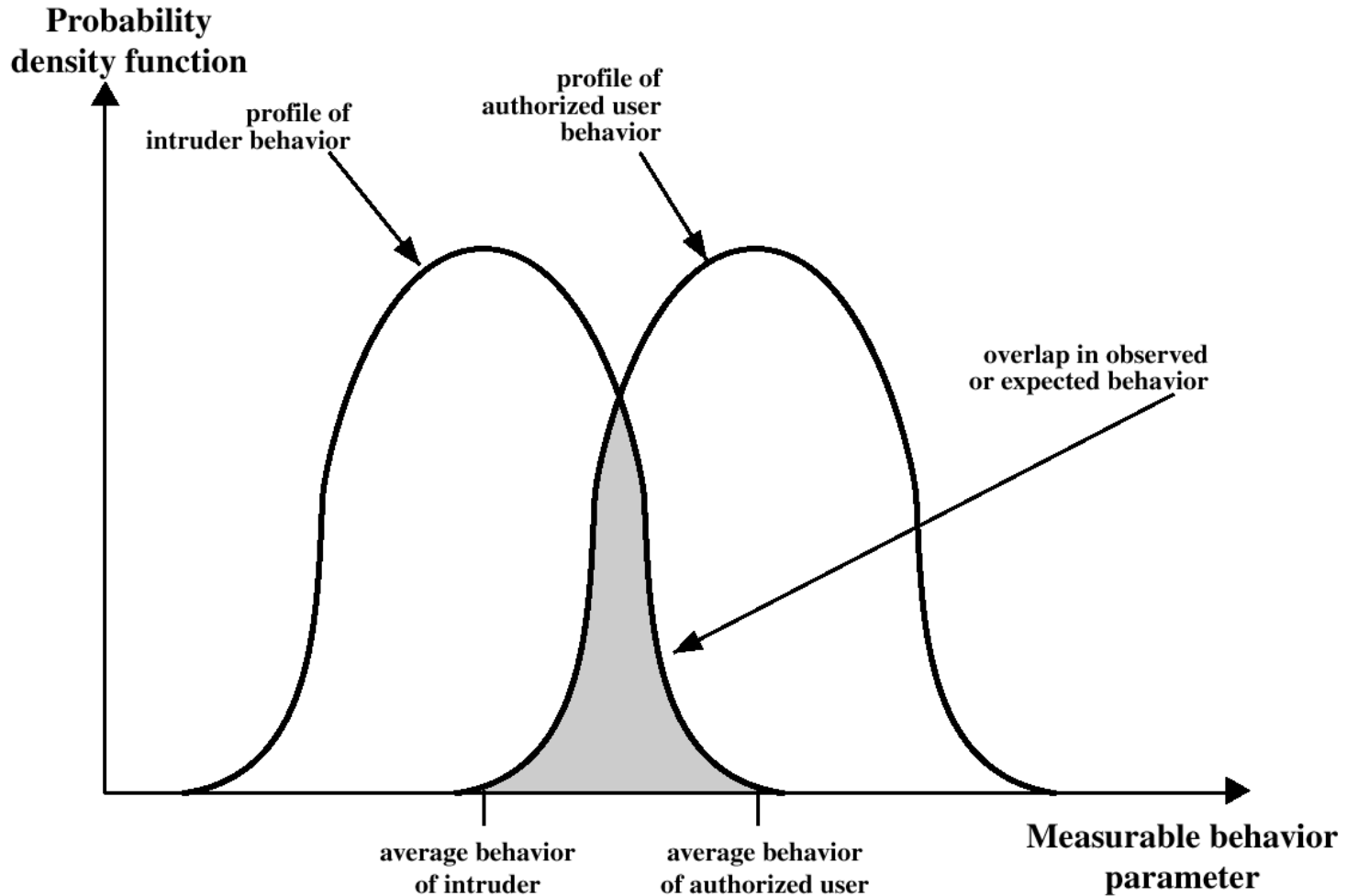
# Etapas de una intrusión

- Barre la red para: localizar IP en uso (*ping*), qué SO usan, qué puertos están abiertos
- Correr scripts “Exploit” contra puertos abiertos
- Obtiene acceso a programas Shell con “suid”
- Bajar de *Hacker Web site* versiones especial de archivos de sistema para tener acceso libre en el futuro sin que se registren uso de CPU o disco (evita auditorías)
- Usa IRC (Internet Relay Chat) para invitar a “amigos” *a la fiesta!!*

# Deteccción de intrusos

- La idea es identificar al intruso y sacarlo del sistema
- Un sistema efectivo de detección de intruso puede servir para disuadir
- La detección de intrusos facilita la recopilación de información sobre técnicas de intrusión, que se puede usar para prevenir futuras intrusiones

# Perfiles de comportamiento de Intrusos y Usuarios Autorizados



# Deteccción de intrusos

- Deteccción estadística de anomalías
  - Deteccción de umbrales (general)
  - Basado en perfiles (por usuario)
- Deteccción basada en reglas
  - Deteccción de anomalías
  - Identificación de la penetración (comportamientos sospechosos)

# Medidas usadas para detección de intrusos

- Frecuencia de conexión, por día y tiempo
- Frecuencia de conexión en varios puntos
- Tiempo desde última conexión
- Fallas de clave de acceso en la conexión
- Frecuencia de ejecución (comandos, prog.)
- Negación de ejecución (más privilegios)
- Frecuencia de lectura, escr., crear, borrar
- No. de fallos de lect., escr., crear y borrar

# Virus y Prog. Dañinos (*malware*)

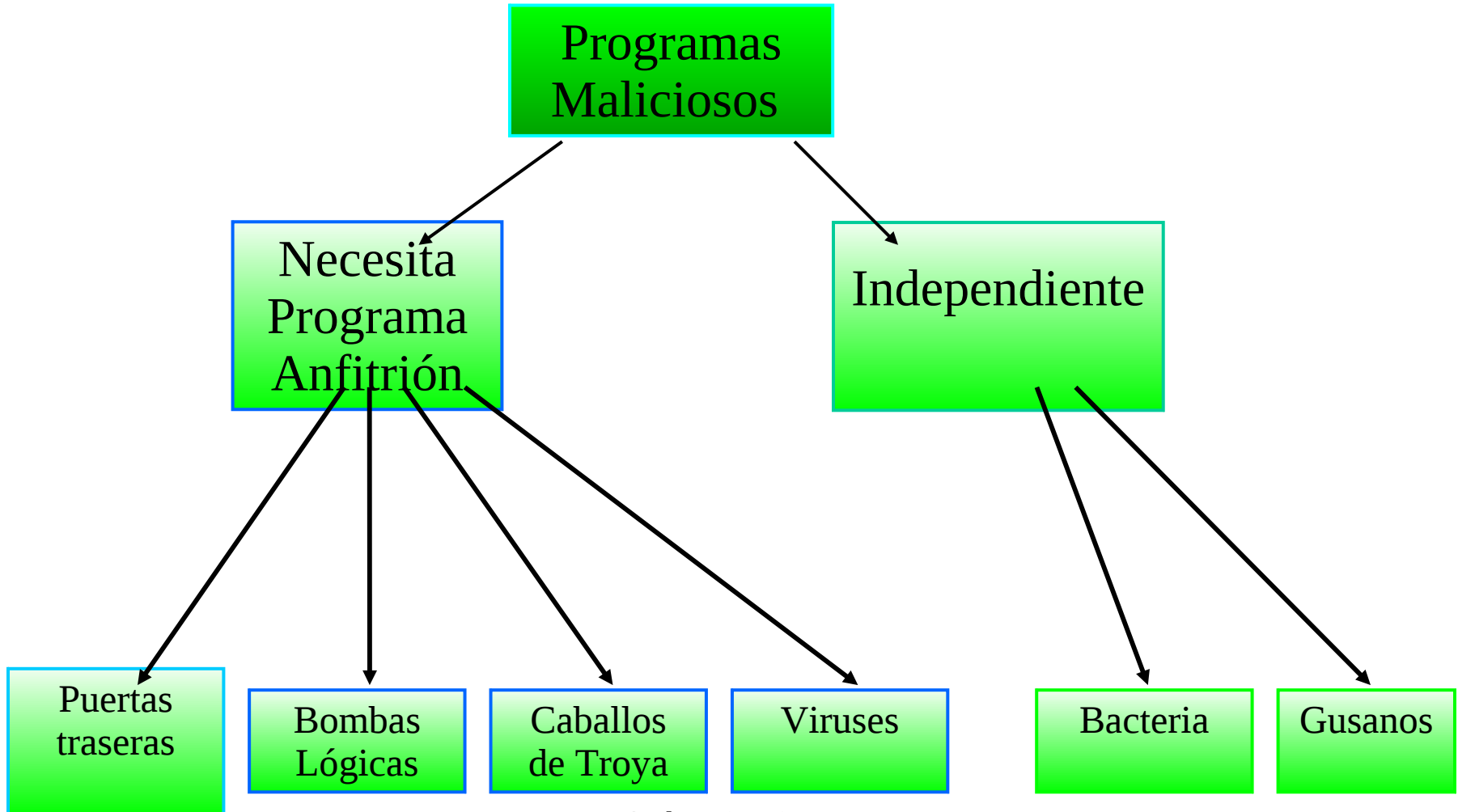
- Virus informáticos

- tienen la habilidad para replicarse a sí mismos en un número creciente de computadoras. Se distribuyen por medios portátiles o Internet (gusanos).

- Otros “Programas dañinos”

- pueden ser instalados a mano o pueden ser distribuido en paquetes comerciales.
- Muy difíciles de detectar antes que se activen (Troyanos, puertas traseras, bombas lógicas)

# Taxonomía de Programas Maliciosos



# Definiciones

- Virus - código que se copia a sí mismo en otros programas
- Una “Bacteria” se replica hasta agotar recursos (CPU, disco), causa Negación de Servicio (*DoS*)
- Carga útil - cosas dañinas que hace el *malware*, luego de extenderse
- Gusano - programa que se replica a sí mismo a través de la red (sobre correo-e o documentos adjuntos)

# Definiciones (cont.)

- Caballo de Troya - instrucciones en un programa aparentemente “inofensivo” que causa que ocurran cosas malas (p.e., cambiar permisos)
- Bomba Lógica - código dañino que se activa en un evento (p.e., fecha)
- Puerta Trasera - punto de entrada para depuración en el código, no documentado, que puede permitir usuarios no deseados
- Huevo de pascua - código ajeno al programa que hace algo “simpático”. Usado por programadores para mostrar que tienen el control del producto!

# Fases de los Virus

- **Fase inactiva** - el virus está inactivo
- **Fase de Propagación** - el virus coloca una copia suya idéntica en otros programas
- **Fase de activación** - el virus se activa para realizar la función para la cual se creó
- **Fase de ejecución** - se ejecuta la función

# Protección contra Virus

- Tener programa antivirus
- No ejecutar programas o macros de fuentes no conocidas
- Evitar los S.O. más vulnerables/atacados (*¿adivinen cuál?* )

# Estructura de Virus

```
program V :=  
  
  {goto main;  
   1234567;  
  
   subroutine infect-executable :=  
     {loop:  
      file := get-random-executable-file;  
      if (first-line-of-file = 1234567)  
        then goto loop  
        else prepend V to file; }  
  
   subroutine do-damage :=  
     {whatever damage is to be done}  
  
   subroutine trigger-pulled :=  
     {return true if some condition holds}  
  
main:  main-program :=  
      {infect-executable;  
       if trigger-pulled then do-damage;  
       goto next;}  
  
next:  
  
}
```

# Tipos de Virus

- **Parásito** - se añade a archivos ejecutables como parte de su código . Corre cuando lo hace el anfitrión.
- **Residente en memoria** - Se aloja en memoria como parte del kernel
- **Sector de arranque** - infecta sector de arranque de un disco, se extiende cuando el S.O. arranca
- **Furtivo** - diseñado para esconderse de antivirus.
- **Polimórfico** - muta en cada nuevo anfitrión, para evitar detección de “firma”

# Virus de macro

- Aplicaciones de MS Office permiten “macros” como parte del documento. Puede correr cuando el documento se abre, o cuando se escoge cierto comando (Save File).
- Independiente de la plataforma
- Infecta documentos, borra archivos, genera correos, y edita cartas

# Enfoques de Antivirus

- 1 Generación, Exploradores: examina archivos buscando firmas de virus conocidos. Verifica cambios de tamaño de archivos ejecutables
- 2 Generación, Exploradores Heurísticos: busca signos más generales, como segmentos de códigos común a varios virus. Verifica *checksum* o *hash*.
- 3 Generación, Captura actividad: permanece en memoria, y busca patrones de comportamiento (p.e., exploración de archivos).
- 4 Generación, Todas las anteriores!

# Técnicas avanzadas de antivirus

- Descifrado genérico (DG)
  - CPU Emulador
  - Examinador de firmas de Virus
  - Módulo de control de emulación
- Cuánto tiempo debe correr el DG en cada interpretación?

# Técnicas avanzadas de antivirus

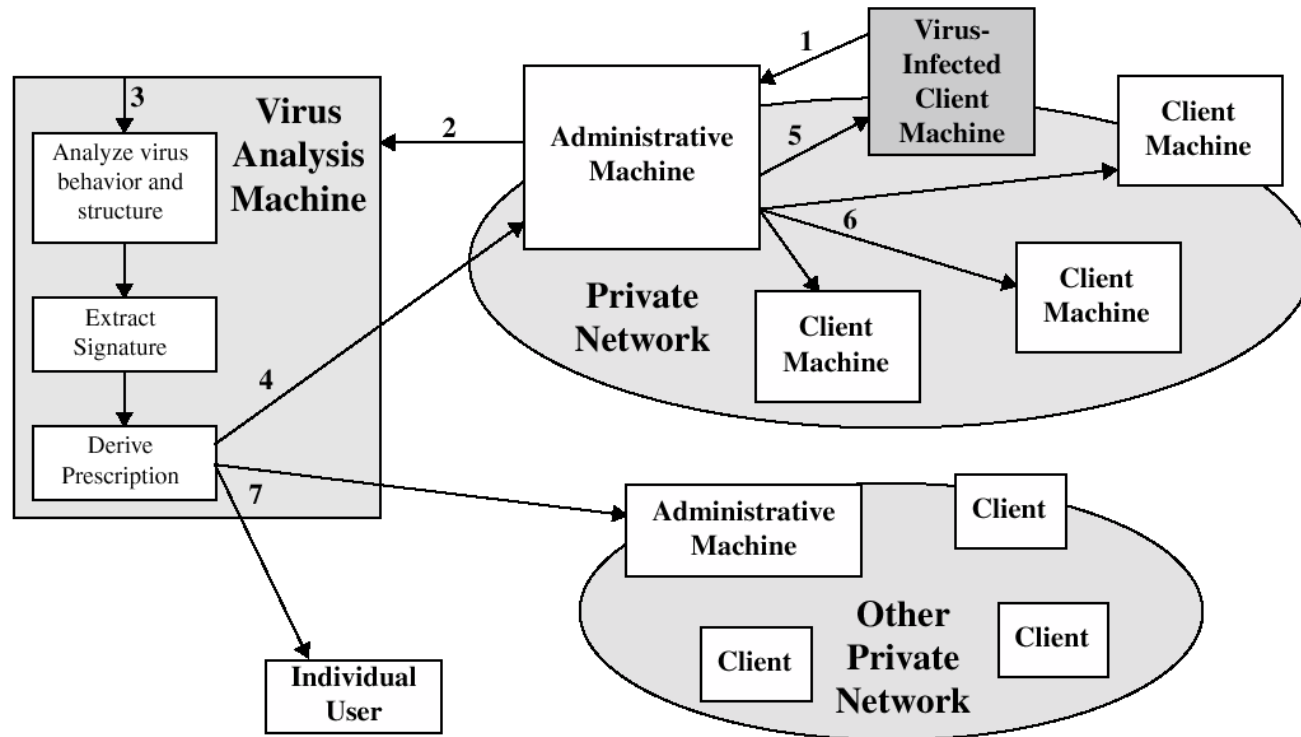


Figure 9.11 Digital Immune System