

## Ejemplo: T-POT

- <https://github.com/telekom-security/tpotce>
- Honeypot para ssh y telnet
- Basado en debian
- Combina una serie de honeypots “dockerizados”
- Incorpora herramientas: cockpit, cyberchef, ELK stack etc

(c) Prof.Miguel García Silvente

113

## sandboxes

- Herramientas que permiten confinar programas para limitar el acceso al sistema completo.
- El kernel de linux usa namespaces /proc/<pid>/ns
- Ejemplos:
  - **chroot** cambia el directorio raíz a partir de ese momento.  
Alternativa: **systemd-nspawn**
  - **firejail** permite ejecutar órdenes en un sandbox
  - **docker** permite controlar el entorno de las aplicaciones.
    - Iniciar servicio docker: systemctl start docker
    - Descargar una imagen: docker pull ubuntu
    - Ver las imágenes: docker images
    - Ejecutar: docker run -i -t ubuntu ls

(c) Prof.Miguel García Silvente

86

## Docker con httpd apache

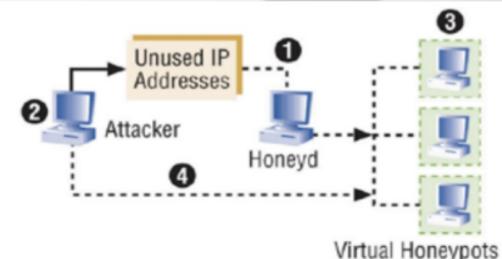
- En <https://hub.docker.com/> hay contenedores preconfigurados.
- Instalar imagen httpd: **docker pull httpd**
- Ejecutar imagen: **sudo docker run -dit --name httpd -p 8080:80 -v /home/user/website:/usr/local/apache2/htdocs/ httpd:2.4**
- Comprobar la ejecución: **sudo docker ps**
- Creamos fichero html: **vi /home/user/website/ejemplo.html**
- Detener ejecución: **sudo docker stop httpd**
- Borrar contenedor: **sudo docker rm httpd**
- Ejecutar órdenes: **sudo docker container exec httpd ls -l /etc**

(c) Prof.Miguel García Silvente

87

## Ejemplo: honeyd

1. Monitoriza las IPs no utilizadas.
2. Cuando un atacante prueba una IP no utilizada, toma esa IP (usando ARP spoofing).
3. Crea un honeypot virtual para que el atacante interactúe con el.
4. Puede crear varios honeypots virtuales y hacer creer al atacante que ha hackeado un sistema.



(c) Prof.Miguel García Silvente

112

## Diseño de red con honeypot



(c) Prof.Miguel García Silvente

111

## Malware en Linux

- Desde Windows XP, mejora mucho la seguridad en MS Windows.
- Aumenta el interés en crear malware para Linux porque empieza a usarse más, especialmente desde la iot (internet de las cosas).
- Ejemplo: la empresa ESET sacó a luz en la operación Windigo que había más de 25000 servidores infectados en tres años.
  - Linux/Ebury un backdoor OpenSSH utilizado para controlar los servidores y robar credenciales, y
  - Linux/Cdorked un backdoor HTTP utilizado para redirigir el tráfico Web.
- Se descubrieron graves vulnerabilidades como Heartbleed (openSSL) o Shellshock (bash, [thehackernews](#))

(c) Prof.Miguel García Silvente

88

## Malware: rootkit

- "Herramienta" que puede actuar independientemente, o bien acompañar a cualquier variante de código malicioso
  - objetivo principal: ocultar su actividad a usuarios y administradores del sistema.
- Misión principal: ocultar información tal como
  - Procesos,
  - Conexiones de red
  - Ficheros
  - Directorios
  - Elevación de privilegios,
  - etc

(c) Prof.Miguel García Silvente

89

## Honeypots para investigación

- No tienen como objetivo proteger redes.
- El objetivo es estudiar todo tipo de patrones de ataque y de amenazas.

(c) Prof.Miguel García Silvente

110

# Honeypots en producción

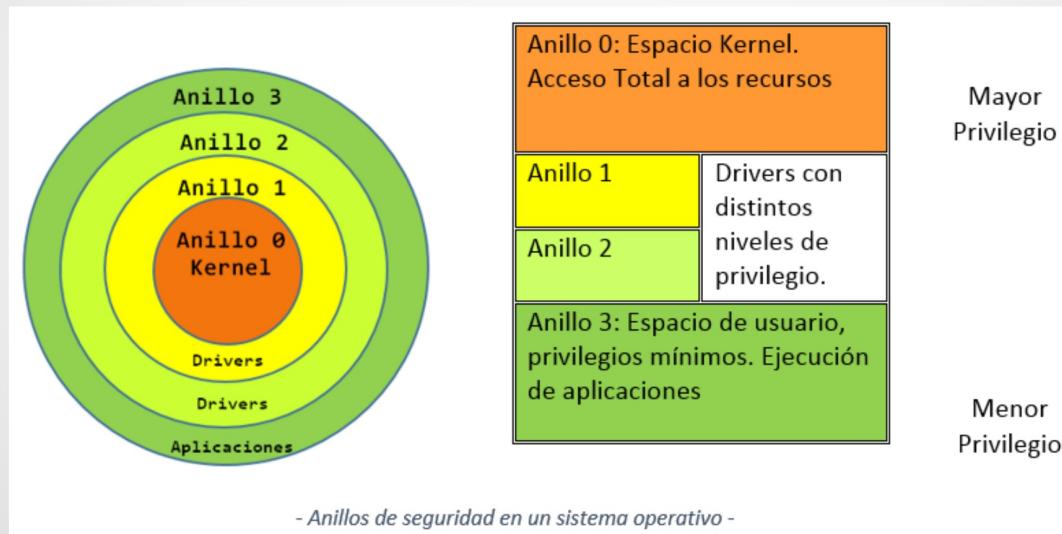
- Se usan para proteger organizaciones en producción.
- Protegen la red, ayudan a mejorar la seguridad.
- Se puede usar para prevención, detección y respuesta.
- A nivel de prevención, se pueden usar para ralentizar o para parar ataques automáticos.

(c) Prof.Miguel García Silvente

109

# Tipos de rootkits

## Espacio de usuario y espacio de kernel



(c) Prof.Miguel García Silvente

90

## Tipos de rootkits: bootkits

- Incorpora funcionalidades de arranque a los rootkits.
- Afecta al firmware de sistemas y sectores de arranque de discos.
- A la dificultad intrínseca de su detección se añade la persistencia y resistencia a ser eliminado.
- Un ejemplo es Thunderstrike, un bootkit para Macbooks que modifica el firmware de la EFI de arranque haciéndolo persistente a reinstalación de SO e incluso a sustitución del disco

## Honeypots por su propósito

- Producción
- Investigación

# Simulación de sistemas

Honeypots: permite “conocer a tu enemigo”

- No son una medida de protección contra ataques.
- De baja interacción: como medida de seguridad
  - Sticky honeypots: ralentizar los ataques automatizados y los rastreos.
  - Nepenthes, Honeyd, Honeytrap, Specter, KFsensor
- De alta interacción: para investigación.
- Otros: sitios webs o sala de chats para descubrir actividades criminales.
- Spam honeypots: Jackpot; smtpot.py y spamhole.
- Ejemplos: Deception Toolkit, Honeynet Project, PenTBox Security Suite, HoneyDrive

# Rootkit en espacio de usuario (I)

Sustituye ejecutables legítimos del sistema por otros modificados.

- Ficheros: ls, df, stat, du, find, lsof, lsattr, chatrr, sync...
- Conexiones: ip, route, netstat, lsof, nc, iptables, arp...
- Procesos: ps, top, pidof, kill, lsof...
- Tareas: crontab, at...
- Logs: syslogd, rsyslogd...
- Accesos: sshd, login, telnetd, inetd, passwd, last, lastlog, su, sudo, who, w, runlevel...

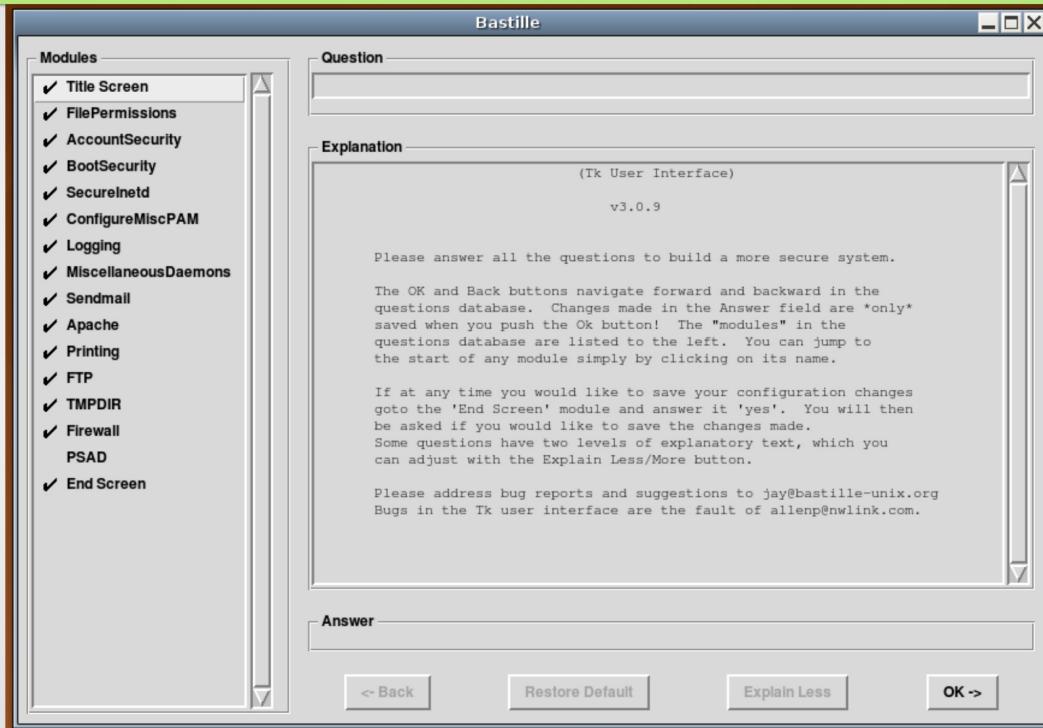
## Rootkit en espacio de usuario (II)

- Cambiar binarios es demasiado evidente.
  - Solución: interceptar llamadas a bibliotecas (si el binario está compilado dinámicamente).
- Se utilizan las variables de entorno LD\_PRELOAD, LD\_LIBRARY\_PATH, o ficheros de caché (/etc/ld.so.cache)

## Índice

- Autorización y control de acceso
- Cifrado
- Seguridad de un sistema: SELinux
- Malware
- **Simulación de sistemas**

# Bastille



(c) Prof. Miguel García Silvente

105

# Rootkit en espacio de usuario (III)

(c) Prof. Miguel García Silvente

94

# Interceptación con LD\_PRELOAD (I)

```
% ldd /bin/ls
linux-vdso.so.1 => (0x00007ffcabb3b000)
libselinux.so.1 => /lib64/libselinux.so.1 (0x000000309e400000)
libcap.so.2 => /lib64/libcap.so.2 (0x00000030a1000000)
libacl.so.1 => /lib64/libacl.so.1 (0x00000030a3000000)
libc.so.6 => /lib64/libc.so.6 (0x000000309cc00000)
libpcre.so.1 => /lib64/libpcre.so.1 (0x000000309e000000)
liblzma.so.5 => /lib64/liblzma.so.5 (0x000000309dc00000)
libdl.so.2 => /lib64/libdl.so.2 (0x000000309d800000)
/lib64/ld-linux-x86-64.so.2 (0x000000309c400000)
libattr.so.1 => /lib64/libattr.so.1 (0x00000030a0c00000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x000000309d400000)
```

(c) Prof. Miguel García Silvente

95

# System hardening

- Proceso de asegurar un sistema reduciendo sus vulnerabilidades o agujeros de seguridad, para los que se está más propenso cuanto más funciones desempeña.
- Una herramienta, Bastille:
  - Permite configurar características habituales de seguridad.
  - Realiza una serie de preguntas acompañada de descripciones:
    - Should we disallow root login on tty's 1-6? [N]:
      - Las tty, son terminales accesibles desde las combinaciones CTRL+ALT+F1 hasta F7, por lo que Bastille nos ofrece deshabilitar el acceso root a dichas terminales para que sea conectado antes con un usuario normal con los mínimos privilegios posibles.
    - Permite evaluar el estado actual y generar informes.

(c) Prof. Miguel García Silvente

104

## AIDE (II)

- crontab -e  
# comprobación  
0 2 \* \* \* /usr/sbin/aide --update | mail -s "web01 - Daily Change Report"  
user@domain.com
- # iniciar la bd AIDE una vez a la semana:  
0 3 \* \* 0 /usr/sbin/aide --init; mv -f /var/lib/aide/aide.db.new.gz  
/var/lib/aide/aide.db.gz
- Copia a dispositivo de sólo lectura  
mkisofs -V Aide\_DB`date +%F` -J -R -o aide.iso /Aide/directory  
cdrecord -v -eject aide.iso

## Interceptación con LD\_PRELOAD (II)

- Por ejemplo, cogemos una función que se use en un ejecutable, como strlen()  
**ltrace** /bin/ls  
...strlen...
- Hay que interceptar la llamada a la biblioteca (hooking) y modificar su comportamiento.
- Crear una biblioteca con una función strlen
- gcc -shared -fPIC -Wall -o mi\_ls.so mi\_ls.c
- export LD\_PRELOAD=\$PWD/mi\_ls.so

## Rootkit en espacio de usuario: resumen

- Son más fáciles de implementar.
- Son más fáciles de detectar:
  - Verificando la integridad con códigos hash.
  - Comprobando las variables de entorno.
  - Seguimiento de enlaces simbólicos, rutas
  - Configuración de bibliotecas dinámicas.
- Hay que compilar los binarios para cada versión
- La precarga de bibliotecas está limitada, por ejemplo, si realizan elevación de privilegios.

## AIDE (I)

- Está basado en tripwire
- AIDE crea una firma (usando hash) de los ficheros que se quieren supervisar y periódicamente supervisa los ficheros.
- Es muy útil en caso de intrusión porque identifica todos los objetos que han sido modificados.
- Se puede guardar la BD en otro sitio.
- Configuración: /etc/aide.conf
- Inicio: aide -i, Comprobación: aide -C, Actualizar: aide -u

## Verificar paquetes con dpkg

- dpkg –verify muestra los ficheros del sistema modificados.
- Usa /var/lib/dpkg/info/<paquete>.md5sums (que podrían ser modificados)
- También podría haber paquetes infectados en un repositorio:
  - Siempre hay que usar repositorios fiables.
  - Siempre hay que usar el sistema de verificación de firma de APT

## Rootkit en espacio de kernel (I)

- Es más interesante porque garantiza un control total y mejor ocultación:
  - Pueden modificar binarios.
  - Funciones y llamadas del sistema operativo.
- Más complejo, interacciona con funciones y llamadas del sistema y cualquier fallo en el kernel provocaría un error grave.

## Rootkit en espacio de kernel (II)

- Kernel de Linux es monolítico: la gestión de recursos, organización/planificación de procesos y acceso a dispositivos recae sobre él.
- Loadable Kernel Modules (LKM), un mecanismo que permite al kernel cargar/descargar en memoria código adicional:
  - Permite añadir funcionalidades sin recompilar.
  - Facilita el uso de rootkits.
- Sin usar módulos, manipulan directamente la imagen de memoria /dev/kmem pero se desactivó en ver. 2.6

## Detección de rootkits

- Con strace y buscando las posibles llamadas.
- Usando **Host Based Intrusion Systems (HIDS)**:
  - Libres: Samhain, LIDS, AIDE,
  - Comerciales: Tripwire, eEye Blink, Symantec HIDS
- chkrootkit: Comprueba signos de rootkits.
- rkhunter : Busca rootkits, backdoors y exploits locales
- unhide: Busca procesos escondidos y puertos TCP/UDP de rootkits.
- Otros: rkdet, checkps, zepoo, rkprofiler lx
- Logwatch, swatch: supervisan los logs del sistema.