



Fuente: Stable Diffusion AI

LABORATORIO 5. SEGURIDAD DE UN SO LINUX (No-AUTOMATIZABLE)

DEPARTAMENTO DE INFORMÁTICA. UNIVERSIDAD DE OVIEDO

Seguridad de Sistemas Informáticos | 2022 – 2023 (v3.1 “S-81 Isaac Peral”)





CONTENIDO

Infraestructura de este laboratorio	3
Bloque 1: Seguridad de los usuarios	5
Evaluación del nivel de seguridad actual	6
Cuentas y acceso: Evitar el modo de usuario único (<i>single user mode</i>)	6
Banners de advertencia	6
Seguridad adicional para <i>OpenSSH</i>	7
Configuración segura del módulo PAM	8
Antigüedad de la contraseña y cuentas inactivas	8
Restringir los inicios de sesión de cuentas de <i>root</i> y de servicios del sistema	8
Evitar el uso de contraseñas comunes	9
Bloque 2: Seguridad de los procesos	10
Compiladores	11
Deshabilitar volcados de memoria del núcleo (<i>core dumps</i>)	11
<i>AppArmor</i>	11
<i>Buscar y aplicar perfiles de AppArmor deshabilitados</i>	12
Instalar software de seguridad <i>apt</i>	12
Contabilidad de procesos	13
Bloque 3: Seguridad de archivos	14
Archivos de configuración de sesión	15
HIDS: <i>Tripwire</i>	15
Bloque 4. Seguridad de redes en un host	17
Configuración de la pila de red de un servidor individual	18
Deshabilitar protocolos de red poco comunes	18
Bloque 5: Log y monitorización	20
Configurar reglas de auditoría	21
Instalar <i>sysstat</i> para ver el consumo de recursos	22
Instalar <i>glances</i>	22
Análisis automatizado de logs: <i>LogCheck</i>	22
Evaluación del nivel de seguridad final con <i>Lynis</i>	23
Insignias y Autoevaluación	24

AVISO

Este documento forma parte de la asignatura “Seguridad de Sistemas Informáticos”, impartida en la *Escuela de Ingeniería Informática* de la *Universidad de Oviedo*. Es fruto del trabajo continuado de elaboración, soporte, mejora, actualización y revisión del siguiente equipo de profesores desde el año 2019

- Enrique Juan de Andrés Galiana
- Fernando Cano Espinosa
- Miguel Riesco Albizu
- José Manuel Redondo López
- Luís Vinuesa Martínez

Te pedimos por favor que **NO lo compartas públicamente en Internet**. No obstante, entendemos que puedas considerar este material interesante para otras personas. Por ese motivo, hemos creado una versión de este adaptada para que pueda cursarse de forma online, disponible gratuitamente para todo el mundo y que puedes encontrar en esta dirección: <https://ocw.uniovi.es/course/view.php?id=109>

A diferencia de esta versión, **la versión libre puedes promocionarla todo lo que quieras**, que para eso está 😊

GRACIAS POR TU COLABORACIÓN

INFRAESTRUCTURA DE ESTE LABORATORIO



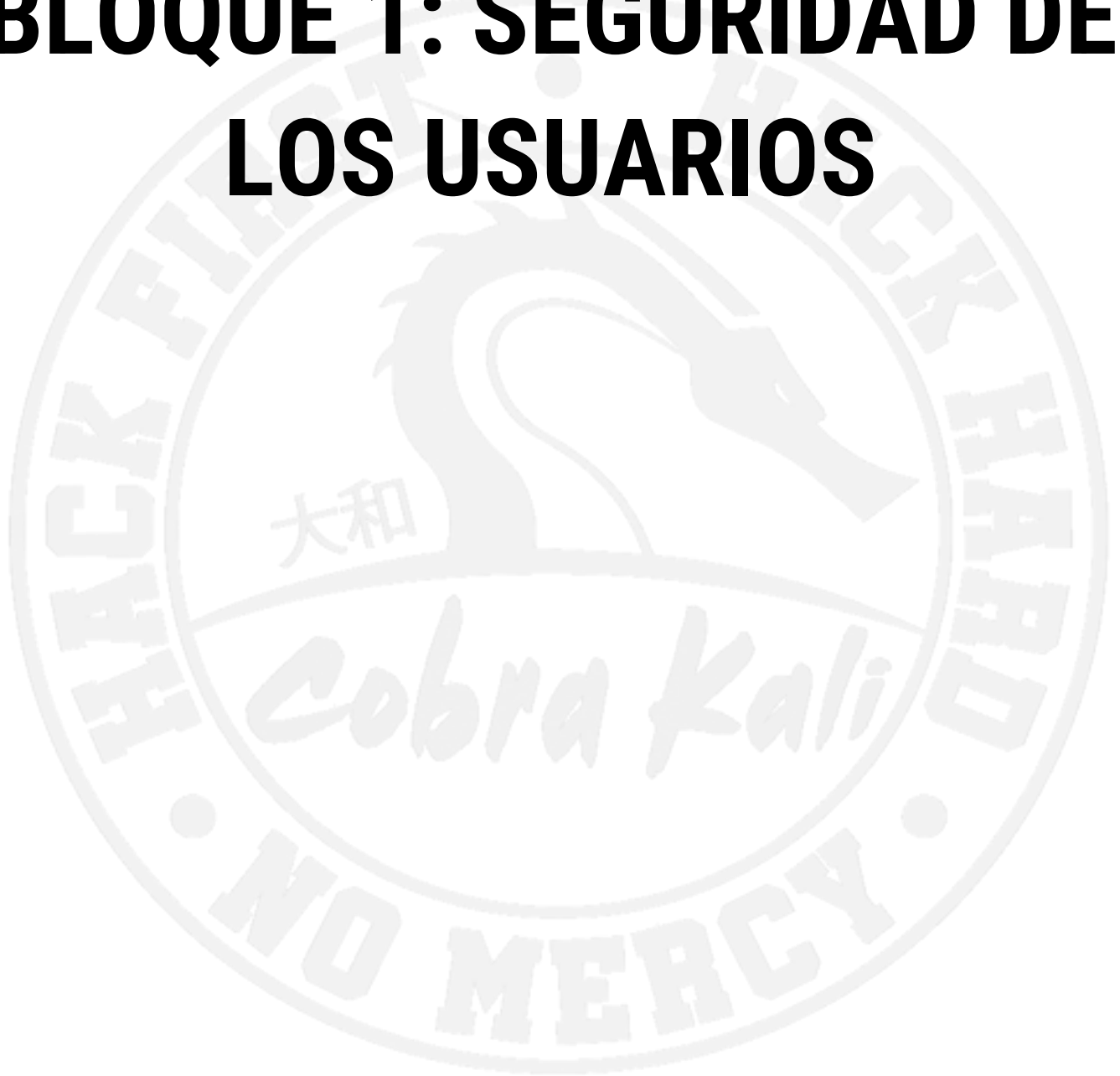
Este laboratorio solo trabajará con la máquina virtual *Ubuntu* de la asignatura curso. Sin embargo, hay una serie de cosas que debes tener en cuenta.

- **Recomendamos encarecidamente crear primero una instantánea de la máquina virtual.**
- Muchas actividades consisten en ir a la sección indicada del **fichero PDF del CIS Benchmark** y seguir sus instrucciones detalladas. **El archivo del CIS Benchmark a usar está en el campus virtual junto con este documento**, así que descárgalo primero. Los *CIS Benchmarks* se describen en el tema de teoría de "Políticas de seguridad". No te preocupes si aún no hemos llegado a este tema, en lo que concierne a este laboratorio, **son solo un conjunto detallado de instrucciones** a seguir para verificar o implementar ciertas operaciones de seguridad. En otras palabras, **es tu manual de instrucciones** 😊. No obstante, solo aplicaremos parte del documento.

ADVERTENCIA: Este laboratorio requiere copiar muchos comandos de un archivo PDF. Por favor, **COMPRUEBA CUIDADOSAMENTE lo que copias**, ya que caracteres Unicode como `❏`, `❏`, etc. pueden considerarse sintácticamente incorrectos en línea de comandos, al copiarse normalmente "tal cual" del fichero (es decir, en Unicode), por lo que debes reemplazarlos por otros equivalentes en ASCII. **Esto es especialmente problemático con las comillas**. Si la sintaxis de un comando no funciona, comprueba cuidadosamente la línea de comandos que has escrito para ver si hay caracteres no válidos copiados accidentalmente. Esto es aplicable a partir de ahora, incluso en futuros laboratorios.



BLOQUE 1: SEGURIDAD DE LOS USUARIOS



Evaluación del nivel de seguridad actual

Aplicación práctica: Necesitas evaluar cuál es el nivel de seguridad actual de tu máquina Linux

Vete al informe laboratorio 1 y sigue el procedimiento de auditoría de **lynis** para evaluar el nivel de seguridad actual de la máquina virtual. Guarda este resultado para compararlo más tarde con la máquina con el proceso de *hardening* terminado.

Resultados esperados: Esta actividad finaliza cuando puedes evaluar la seguridad de tu máquina virtual con **lynis**

Cuentas y acceso: Evitar el modo de usuario único (*single user mode*)

Aplicación práctica: Necesitas evitar que alguien arranque tu máquina y adquiera privilegios de **root** entrando en el modo *single user* (también llamado modo de mantenimiento)

Una de las cosas más importantes que debes hacer con respecto a la administración de cuentas es **evitar el acceso al modo de usuario único (*single user mode*)**. Este modo se usa para la recuperación ante desastres, cuando el sistema detecta un problema durante el arranque, o también deja entrar en el seleccionándolo manualmente en el gestor de arranque. Para evitar el arranque en modo *single user*, haz la tarea del CIS benchmark **1.5.3 Ensure authentication required for single user mode**.

Resultados esperados: Esta actividad finalizará cuando puedas evitar el arranque no autenticado en modo *single user* de acuerdo con las especificaciones del CIS.

Banners de advertencia

Aplicación práctica: Necesitas decirle a los posibles intrusos “fuera de mi jardín” (como Clint Eastwood en “Gran Torino” 😊)

Es una buena práctica de seguridad advertir a los usuarios que se conecten remotamente sobre las consecuencias de conectarse a un sistema, en caso de que no sean usuarios legítimos, o de las restricciones de uso (si son legítimos). Para poner banners de advertencia correctos en los diferentes puntos de acceso de un sistema, haz la tarea del CIS benchmark **1.8.1 Command Line Warning Banners**. Concretamente, implementa solo estas tareas:

- **1.8.1.1 Ensure message of the day is configured properly.**
- **1.8.1.2 Ensure local login warning banner is configured properly.**
- **1.8.1.3 Ensure remote login warning banner is configured properly.**

Puedes usar cualquier texto para los mensajes de advertencia, por ejemplo: <https://www.tecmint.com/protect-ssh-logins-with-ssh-motd-banner-messages/>

Resultados esperados: Esta actividad finalizará cuando tengas configurados *banners* de advertencia adecuados en tu sistema de acuerdo con las especificaciones del CIS.

Seguridad adicional para OpenSSH

Aplicación práctica: Necesitas fortalecer las conexiones remotas vía SSH todo lo posible

En laboratorios anteriores instalamos OpenSSH, te enseñamos a crear contraseñas seguras e incluso habilitamos un método sin contraseña para conectarse con él. Esto mejora su seguridad, pero podemos ir mucho más allá con la tarea del CIS Benchmark **5.2 SSH Server Configuration** y la mayoría de sus subtarefas. Se recomienda hacer **una copia de seguridad de la configuración actual** en caso de que algo salga mal y **ssh** quede no disponible debido a ello. Siempre puedes iniciar sesión a través de la consola del software de virtualización para arreglar cualquier desastre 😊.

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.factory-defaults  
sudo chmod a-w /etc/ssh/sshd_config.factory-defaults
```

Una vez hecho esto, modifica **/etc/ssh/sshd_config** con los valores y directivas indicados por las siguientes tareas del CIS Benchmark. **Ctrl+W** es el comando de búsqueda en el editor **nano**. Si un nombre de directiva no está ya dentro del archivo de configuración SSH de tu SO, agrega una línea con él y el valor recomendado.

- 5.2.1 Ensure permissions on **/etc/ssh/sshd_config** are configured.
- 5.2.2 Ensure permissions on SSH private host key files are configured.
- 5.2.3 Ensure permissions on SSH public host key files are configured.
- 5.2.4 Ensure SSH **Protocol** is not set to 1.
- 5.2.5 Ensure SSH **LogLevel** is appropriate.
- 5.2.6 Ensure SSH X11 forwarding is disabled.
- 5.2.7 Ensure SSH **MaxAuthTries** is set to 4 or less.
- 5.2.8 Ensure SSH **IgnoreRhosts** is enabled.
- 5.2.9 Ensure SSH **HostbasedAuthentication** is disabled.
- 5.2.10 Ensure SSH root login is disabled.
- 5.2.11 Ensure SSH **PermitEmptyPasswords** is disabled.
- 5.2.12 Ensure SSH **PermitUserEnvironment** is disabled.
- 5.2.13 Ensure only strong Ciphers are used.
- 5.2.14 Ensure only strong MAC algorithms are used.
- 5.2.15 Ensure only strong Key Exchange algorithms are used.
- 5.2.16 Ensure SSH Idle Timeout Interval is configured.
- 5.2.17 Ensure SSH **LoginGraceTime** is set to one minute or less.
- 5.2.19 Ensure SSH warning banner is configured.
- 5.2.20 Ensure SSH PAM is enabled.
- 5.2.21 Ensure SSH **AllowTcpForwarding** is disabled.
- 5.2.22 Ensure SSH **MaxStartups** is configured.
- 5.2.23 Ensure SSH **MaxSessions** is set to 4 or less.

Resultados esperados: Esta actividad finalizará cuando se mejore la seguridad de la configuración del servicio **ssh** de acuerdo con las especificaciones del CIS.

Configuración segura del módulo PAM

Aplicación práctica: *Necesitas fortalecer tu política de passwords lo más que puedas*

PAM (*Pluggable Authentication Modules*) es un servicio que implementa sistemas de autenticación modulares en sistemas UNIX. PAM se implementa como un conjunto de objetos compartidos que se cargan y ejecutan cuando un programa necesita autenticar a un usuario. Los archivos para PAM normalmente se encuentran en el directorio `/etc/pam.d`. **PAM debe configurarse cuidadosamente** para tener una seguridad adecuada en la autenticación del sistema, y podemos hacerlo implementando la tarea del CIS benchmark **5.3 Configure PAM** y sus subtareas.

- **5.3.1 Ensure password creation requirements are configured.**
- **5.3.2 Ensure lockout for failed password attempts is configured.**
- **5.3.3 Ensure password reuse is limited.**
- **5.3.4 Ensure password hashing algorithm is SHA-512.**

Resultados esperados: Esta actividad finalizará cuando crees una política de contraseñas del sistema más segura de acuerdo con las especificaciones del CIS.

Antigüedad de la contraseña y cuentas inactivas

Aplicación práctica: *Necesitas eliminar cuentas no usadas y passwords viejas*

La antigüedad de la contraseña (el número de días que una contraseña no se ha cambiado) y las cuentas inactivas pueden ser un vector de problemas de seguridad. No solo porque tengas más cuentas de usuario que supervisar, sino porque además no hay nadie que las use (y se pueda dar cuenta de que alguien más está usando su cuenta) y/o es más fácil que se filtre una clave que un atacante pueda usar si no se cambia. Podemos controlarlo aplicando la tarea del CIS benchmark **5.4.1 Set Shadow Password Suite Parameters** y sus subtareas:

- **5.4.1.1 Ensure password expiration is 365 days or less.**
- **5.4.1.2 Ensure minimum days between password changes is configured.**
- **5.4.1.3 Ensure password expiration warning days is 7 or more.**
- **5.4.1.4 Ensure inactive password lock is 30 days or less.**

Resultados esperados: Esta actividad finalizará cuando refuerces la política de caducidad e inactividad de contraseñas del sistema de acuerdo con las especificaciones del CIS.

Restringir los inicios de sesión de cuentas de root y de servicios del sistema

Aplicación práctica: *Necesitas asegurarte de que ni `root` ni una cuenta de servicio puedan hacer login interactivo*

Hay una serie de medidas necesarias para mejorar la seguridad de los inicios de sesión de cuentas de `root` y de servicios del sistema, o bien verificar que estén configurados correctamente. Podemos hacerlo



implementando la tarea del CIS benchmark **5.4.2 Ensure system accounts are secured** y la tarea **6.2.6 Ensure root is the only UID 0 account** .

Resultados esperados: Esta actividad finalizará cuando refuerces la política de inicio de sesión de las cuentas de **root** y de servicios del sistema de acuerdo con las especificaciones del CIS.

Evitar el uso de contraseñas comunes

Aplicación práctica: Necesitas evitar que tus usuarios usen password comunes fáciles de crackear

Estas operaciones ayudan a tener una directiva de contraseñas adecuada. No basta con aplicar una política de contraseñas seguras; necesitamos complementarlo con una prevención **que impida el uso de contraseñas comunes** para asegurarnos de que no sea sencillo que haya inicios de sesión no deseados en nuestro sistema. Esta actividad evita que los usuarios puedan usar contraseñas que pertenezcan a la lista del **millón de contraseñas más utilizadas en Internet**: <https://github.com/danielmiessler/SecLists>.

- Instala: `sudo apt-get install libpam-cracklib -y`
- Descargue la lista de contraseñas comunes: `sudo wget https://raw.githubusercontent.com/danielmiessler/SecLists/master/Passwords/xato-net-10-million-passwords-1000000.txt /usr/share/dict/ -O /usr/share/dict/million.txt`
- Hacer cumplir el contenido de la lista: `sudo create-cracklib-dict /usr/share/dict/million.txt`

Puedes comprobar esto intentando cambiar la contraseña de cualquier usuario (o creando un nuevo usuario) y usando cualquier palabra de esta lista como contraseña que quieres poner. Debería dar una advertencia.

Resultados esperados: Esta actividad finalizará cuando impidas el uso de contraseñas comunes al cambiar o crear contraseñas en el sistema.

BLOQUE 2: SEGURIDAD DE LOS PROCESOS



Compiladores

Aplicación práctica: Necesitas eliminar compiladores que tú no vas a usar pero sí que pueden usarse para desplegar malware

Como realmente no los estamos utilizando, puedes deshabilitar los compiladores comunes instalados en la mayoría de los sistemas *Ubuntu* con los siguientes comandos. Para simplificar, puedes ponerlos en un archivo `.sh`, darle permisos de ejecución y ejecutarlo directamente. **NOTA: si salen errores de "directory not found" no son un problema real: significan que un compilador no está instalado.** No añadas más compiladores a este *script*, ya que algunos son necesarios para que el sistema operativo funcione y deshabilitarlos puede impedir que el sistema siga en ejecución.

```
chmod 000 /usr/bin/byacc
chmod 000 /usr/bin/yacc
chmod 000 /usr/bin/bcc
chmod 000 /usr/bin/kgcc
chmod 000 /usr/bin/cc
chmod 000 /usr/bin/gcc
chmod 000 /usr/bin/*c++
chmod 000 /usr/bin/*g++
```

Resultados esperados: Esta actividad finalizará cuando deshabilites los compiladores de C/C++ en la MV, si existen.

Deshabilitar volcados de memoria del núcleo (core dumps)

Aplicación práctica: Necesitas eliminar los ficheros de core dump que se pueden usar para obtener información confidencial

Implementa la tarea del CIS benchmark **1.6.4 Ensure core dumps are restricted** para deshabilitar esta posible vía de obtener información confidencial

Resultados esperados: Esta actividad finalizará cuando se impidan los volcados de núcleo de acuerdo con las especificaciones del CIS.

AppArmor

Aplicación práctica: Necesitas asegurarte de que el sistema MAC AppArmor está activo y confinando tu navegador Firefox

AppArmor es un sistema de **control de acceso obligatorio** (MAC) que limita los programas a un conjunto de recursos a los que pueden acceder o usar. Es decir, restringe los programas a un conjunto de archivos, atributos y capacidades, por lo que no pueden causar daños graves si se modifica su código para hacer alguna operación indebida o son maliciosos (a menos que se les dé permiso). *AppArmor* funciona a nivel de *kernel* y se carga

durante el arranque. Controla los permisos a través de *perfiles*, que son un conjunto de reglas que determinan lo que el programa puede y no puede hacer. Hay dos formas de aplicar los perfiles:

- **Enforce:** Aplicación de la política definida en el perfil y notificación de intentos de violación de esta.
- **Complain:** Solo informa intentos de violación de la política, pero no obliga a que se cumpla.

La mayoría de los perfiles se cargan en el modo *Enforce*, aunque puede haber perfiles de terceros que también se carguen en el modo *Complain*.

Para comprobar que *AppArmor* está configurado correctamente en un sistema, implementa la tarea del *Benchmark CIS 1.7.1 Configure AppArmor*, aunque más concretamente estas dos.

- **1.7.1.1 Ensure AppArmor is installed**
- **1.7.1.3 Ensure all AppArmor Profiles are in enforce or complain mode**

Buscar y aplicar perfiles de *AppArmor* deshabilitados

Aparte de los perfiles que se ejecutan en el arranque, puede haber algunos que estén deshabilitados de forma predeterminada. Podemos comprobar si hay perfiles deshabilitados en la carpeta `/etc/apparmor.d/disable`. Puedes encontrar aquí dos perfiles deshabilitados: *Firefox* y *Rsyslogd*. Como *Firefox* es una aplicación “sensible”, debemos habilitarla siguiendo este procedimiento.

- Instala este paquete: `sudo apt install apparmor-utils`
- Desde un terminal, escribe: `sudo aa-enforce /etc/apparmor.d/usr.bin.firefox`
- Si quieres deshabilitarlo de nuevo, haz:

```
sudo ln -s /etc/apparmor.d/usr.bin.firefox /etc/apparmor.d/disable/  
sudo apparmor_parser -R /etc/apparmor.d/usr.bin.firefox
```

Resultados esperados: Esta actividad finalizará cuando compruebes que el estado de *AppArmor* sigue las especificaciones de los *benchmarks CIS* y consigas habilitar un perfil de *AppArmor* para *Firefox*.

Instalar software de seguridad *apt*

Aplicación práctica: Necesitas estar seguro de que no se instalan paquetes corruptos

Puedes obtener más seguridad en las instalaciones de software con `apt` instalando los paquetes `debsums` y `apt-show-versions`. Con ellos impedirás la instalación de paquetes alterados maliciosamente. Para probarlo, sigue estas instrucciones: <https://manpages.ubuntu.com/manpages/trusty/man1/debsums.1.html> que te permitirán comprobar la integridad de los paquetes que elijas

Resultados esperados: Esta actividad finalizará cuando se instale el software mencionado y lo uses para comprobar la integridad de algunos paquetes.

Contabilidad de procesos

Aplicación práctica: *Necesitas activar un log del comportamiento de los procesos para usos (forenses) posteriores*

La contabilidad de procesos permite hacer log de los comportamientos de los procesos, que pueden ser útiles para el análisis forense y para comprender fallos. Instalarlo es muy fácil:

- Instala el paquete `acct`: `sudo apt install acct`
- Crea un archivo de log: `sudo touch /var/log/pacct`
- Activa contabilidad de procesos: `sudo accton /var/log/pacct`

Resultados esperados: Esta actividad finalizará cuando el software esté instalado y en ejecución.

BLOQUE 3: SEGURIDAD DE ARCHIVOS



Archivos de configuración de sesión

Aplicación práctica: Necesitas poner los permisos por defecto de los ficheros que se creen nuevos en la configuración segura más adecuada

Los archivos nuevos se pueden crear con una configuración de permisos por defecto cuya seguridad se puede mejorar. Para mejorar estos permisos, implementa la tarea del CIS benchmark **5.4.4 Ensure default user umask is 027 or more restrictive**.

Resultados esperados: Esta actividad finalizará cuando el `umask` predeterminado tenga un valor seguro de acuerdo con las especificaciones del CIS.

HIDS: Tripwire

Aplicación práctica: Necesitas monitorizar cualquier cambio a ficheros importantes en tu SO para detectar comportamientos no deseados potenciales

Esta actividad cubre la tarea del CIS benchmark **1.4 Filesystem Integrity Checking** y sus subtareas. Aunque el benchmark usa un HIDS diferente, este cubre los mismos requisitos. Esta actividad se basa en la naturaleza de los archivos en un disco duro; los ficheros en un disco duro se pueden dividir en dos categorías:

- Los que normalmente **se modificarán con frecuencia**: archivos de usuario, sitios web, logs...
- Archivos que **rara vez se modifican**: controladores de dispositivo, archivos del sistema... una modificación de un archivo de este segundo grupo puede indicar una intrusión en el servidor o la presencia de ciertos tipos de *malware* (¡o también una actualización del sistema! 😊)

Hay una forma de detectar intrusiones que modifican archivos "sensibles" con herramientas como `tripwire`: un *Host-based Intrusion Detection System (HIDS)* que alerta al administrador cuando se modifica un archivo considerado importante (o que pertenece al sistema operativo), para que se puedan tomar medidas e investigar si es un problema. Lo primero es instalar el software en la máquina virtual (`sudo apt install tripwire`). Esta instalación es un proceso guiado en el que el instalador nos pedirá los parámetros necesarios. Estos son:

- Los incidentes de seguridad detectados normalmente se informarán vía email, utilizando un programa de correo electrónico como `postfix`. En nuestras pruebas no vamos a hacer esto para ahorrarnos instalar más software, por lo que cuando se nos pida que configuremos el correo electrónico debemos poner "Local only " (¡esto no es adecuado para entornos de producción!). También dejamos el nombre de correo electrónico predeterminado.
- *Tripwire* necesita dos claves para realizar una firma criptográfica de los archivos que monitoriza para que se detecten cambios (recuerda el tema de *Criptografía*). Es recomendable dejar que el instalador las genere. Hay dos claves: **clave de sitio (site key)** y **clave local (local key)**, y se nos pedirán contraseñas que protejan ambas. **Estas contraseñas NO se almacenan, ¡así que por favor no las olvides!**
 - Los archivos firmados con la clave de sitio son comunes a varios sistemas
 - Los archivos firmados con la clave local son típicos del sistema operativo local



- La base de datos de *Tripwire* se cifra con la clave de sitio. Si cambiamos la configuración de *tripwire*, debemos regenerar esta base de datos y se necesitará introducir esa clave de nuevo.
- Después de generar las claves protegidas, se creará un archivo con la directiva de supervisión de archivos del sistema (*twpol.txt*). Esta es la política seguida por *tripwire* en cada ejecución
- Finalmente, para usar *Tripwire* necesitamos inicializarlo (`sudo tripwire --init`): esto recorrerá el sistema de archivos, examinará los archivos de acuerdo con la política, firmándolos y almacenando estas firmas en una base de datos cifrada. Toda la configuración se almacena en */etc/tripwire*
- Desafortunadamente, la directiva que se instala de forma predeterminada suele devolver muchos errores cuando se inicializa. Ocurren porque se examinan archivos que cambian en cada arranque o que no existen. **Necesitamos modificar la política** para excluirllos y tener una ejecución más “limpia”. Para ello, sustituye el archivo */etc/tripwire/twpol.txt* por la versión que os damos en los archivos de laboratorio, que es el mismo archivo pero con las líneas que hacen referencia a estos archivos/rutas problemáticas comentadas.
- Una vez hecho esto, guarda los cambios y recrea la política de *Tripwire* con: `sudo twadmin -m P /etc/tripwire/twpol.txt`. Necesitaremos introducir la clave del sitio.
- Finalmente, podemos reinicializar *tripwire*: `sudo tripwire --init` para obtener un estado de los archivos del sistema libre de errores, útil para detectar cambios no deseados en adelante.
- Una vez terminado, podemos verificar el sistema de archivos con `sudo tripwire --check`

Resultados esperados: Esta actividad finalizará cuando puedas instalar y utilizar *Tripwire* correctamente

BLOQUE 4. SEGURIDAD DE REDES EN UN HOST



Configuración de la pila de red de un servidor individual

Aplicación práctica: *Necesitas configurar tu pila de red de la forma más segura posible*

Esta parte del laboratorio enseña a hacer **una configuración más segura a la pila de red de un host** a través de los *CIS Benchmarks*. Son una serie de ajustes que configuran la pila de red de un host para evitar ciertos tipos de ataques. Se pueden hacer sobre servidores individuales sin importar si hay un *firewall* activo o no para mejorar la seguridad de la configuración de un sistema. En otras palabras, en nuestra arquitectura de red segura (SNI) del tema 5 de teoría, esto se podría hacer sobre cada uno de sus servidores, sin importar subred o funcionalidad.

Esta actividad consiste en implementar los siguientes controles de seguridad del *CIS Benchmark* en un archivo. Ten en cuenta que casi todos ellos requieren agregar o modificar líneas en el mismo archivo, por lo que puedes hacerlos todos juntos simplemente siguiendo las instrucciones y cambiando el archivo con todos los parámetros en un solo paso.

Se requiere implementar las siguientes tareas de la sección **3 Network Configuration** del *CIS Benchmark* que proporcionamos.

- **3.1 Network Parameters (Host Only)**
 - 3.1.1 Ensure packet redirect sending is disabled
 - 3.1.2 Ensure IP forwarding is disabled
- **3.2 Network Parameters (Host and Router)**
 - 3.2.1 Ensure source routed packets are not accepted
 - 3.2.2 Ensure ICMP redirects are not accepted
 - 3.2.3 Ensure secure ICMP redirects are not accepted
 - 3.2.4 Ensure suspicious packets are logged
 - 3.2.5 Ensure broadcast ICMP requests are ignored
 - 3.2.6 Ensure bogus ICMP responses are ignored
 - 3.2.7 Ensure Reverse Path Filtering is enabled
 - 3.2.8 Ensure TCP SYN Cookies is enabled
 - 3.2.9 Ensure IPv6 router advertisements are not accepted

Resultados esperados: Esta actividad finalizará cuando configures los parámetros indicados de la pila de red de tu máquina virtual según lo especificado por el CIS.

Deshabilitar protocolos de red poco comunes

Aplicación práctica: *Necesitas eliminar los protocolos de red que no usas para minimizar tu superficie de exposición*

Este ejercicio requiere la implementación de las siguientes tareas de la sección **3.4 Uncommon Network Protocols** del *CIS Benchmark* que os proporcionamos.

- **3.4 Uncommon Network Protocols**
 - 3.4.1 Ensure DCCP is disabled



- 3.4.2 Ensure SCTP is disabled
- 3.4.3 Ensure RDS is disabled
- 3.4.4 Ensure TIPC is disabled

Resultados esperados: esta actividad finalizará cuando te asegures de que los protocolos de red indicados están deshabilitados en la máquina virtual según lo especificado por el CIS.



BLOQUE 5: LOG Y MONITORIZACIÓN



Configurar reglas de auditoría

Aplicación práctica: Necesitas poner en marcha un perfil de auditoría completo en tu SO Linux

Esta tarea requiere la implementación de las siguientes tareas de la sección **4 Logging and Auditing** del CIS Benchmark que os proporcionamos. Sin embargo, esta implementación se puede simplificar mucho gracias a los archivos incluidos como parte de los materiales de este laboratorio en la carpeta **audit**. Estos archivos son la implementación de cada control de seguridad del benchmark que implique editar o crear un archivo en la carpeta **/etc/audit/rules.d/**. De esta forma, **solo tienes que copiar todos esos archivos a esa carpeta para implementar la corrección de todos esos controles de seguridad**. De esta manera se implementarán automáticamente la mayoría de los controles de esta lista:

- **4 Logging and Auditing**
 - 4.1 Configure System Accounting (auditd)
 - 4.1.1 Ensure auditing is enabled
 - 4.1.1.1 Ensure auditd is installed
 - 4.1.1.2 Ensure auditd service is enabled
 - 4.1.1.3 Ensure auditing for processes that start prior to auditd is enabled
 - 4.1.1.4 Ensure audit_backlog_limit is sufficient
 - 4.1.2 Configure Data Retention
 - 4.1.2.1 Ensure audit log storage size is configured
 - 4.1.2.2 Ensure audit logs are not automatically deleted
 - 4.1.2.3 Ensure system is disabled when audit logs are full
 - 4.1.3 Ensure events that modify date and time information are collected
 - 4.1.4 Ensure events that modify user/group information are collected (Scored)
 - 4.1.5 Ensure events that modify the system's network environment are collected
 - 4.1.6 Ensure events that modify the system's Mandatory Access Controls are collected
 - 4.1.7 Ensure login and logout events are collected
 - 4.1.8 Ensure session initiation information is collected
 - 4.1.9 Ensure discretionary access control permission modification events are collected
 - 4.1.10 Ensure unsuccessful unauthorized file access attempts are collected
 - 4.1.11 Ensure use of privileged commands is collected
 - 4.1.12 Ensure successful file system mounts are collected
 - 4.1.13 Ensure file deletion events by users are collected
 - 4.1.14 Ensure changes to system administration scope (sudoers) is collected
 - 4.1.15 Ensure system administrator actions (sudolog) are collected
 - 4.1.16 Ensure kernel module loading and unloading is collected
 - 4.1.17 Ensure the audit configuration is immutable

Resultados esperados: esta actividad finalizará cuando te asegures de que los controles de seguridad de los benchmark CIS que configuran una política de auditoría robusta están implementados y el servicio de auditoría funciona una vez reiniciado tras hacer todos los cambios.

Instalar sysstat para ver el consumo de recursos

Aplicación práctica: Necesitas monitorizar el uso de recursos en tu SO Linux

La supervisión del consumo de recursos en un sistema puede identificar comportamientos sospechosos que podrían estar perjudicando el rendimiento del sistema. **sysstat** (<https://github.com/sysstat/sysstat>) es un paquete popular que contiene varias utilidades para monitorizar el rendimiento del sistema y su actividad de uso, que podemos instalar con `sudo apt install sysstat`:

- **iostat** informa de estadísticas de CPU y de entrada/salida para dispositivos de bloque y particiones.
- **mpstat** informa de estadísticas individuales o combinadas relacionadas con el procesador.
- **pidstat** informa de estadísticas para tareas (procesos) de Linux: E/S, CPU, memoria, etc.
- **tapestat** informa de estadísticas de unidades de cinta conectadas al sistema.
- **cifsio** informa de estadísticas de CIFS (Common Internet File System).

Resultados esperados: Esta actividad finalizará cuando puedas comprobar las estadísticas de uso del procesador utilizando este paquete.

Instalar glances

Aplicación práctica: Necesitas monitorizar el uso de CPU de tu SO Linux

Glances es otro paquete popular que permite monitorizar el uso de recursos en todo un sistema en una sola pantalla, para que el uso de recursos pueda ser controlado en tiempo real más fácilmente. Puedes instalarlo como cualquier paquete típico: `sudo apt install glances` y usarlo ejecutando `glances` en la línea de comandos.

Resultados esperados: Esta actividad finalizará cuando puedas comprobar las estadísticas de uso del procesador utilizando este paquete.

Análisis automatizado de logs: LogCheck

Aplicación práctica: Necesitas capacidades básicas de análisis de logs en tu SO Linux

LogCheck es una herramienta automatizada de comprobación de logs que puede ayudarnos a detectar comportamientos inusuales en el sistema, que deberían indicar actividad sospechosa. El curso opcional **Informática Forense** trata de este tipo de actividades (y otras relacionadas), por lo que no vamos a explicar esto en detalle, solo te mostraremos cómo realizar la detección básica de anomalías. Para instalarlo sigue este procedimiento:

- Instalar: `sudo apt install logcheck`
- Configurarlos como *Local only*, ya que no vamos a usar el correo electrónico para recibir informes.
- Dejar el nombre predeterminado.
- Comprobar logs sin enviar informes por correo electrónico: `sudo -u logcheck -o -t`



- Verificar los logs sin enviar informes de correo electrónico para eventos de inicio de sesión: `sudo -u logcheck -o -t | grep login`

Hay muchas más cosas que este software puede hacer que se pueden consultar aquí: <http://somebooks.es/recibir-informes-sobre-sucesos-de-ubuntu-server-18-04-lts-con-logcheck/>. No obstante, esto es suficiente para este laboratorio:

Resultados esperados: Esta actividad finalizará cuando puedas verificar los eventos de inicio de sesión utilizando *logs* con este paquete.

Evaluación del nivel de seguridad final con Lynis

Aplicación práctica: Sabes cómo comparar los índices de seguridad una vez has implementado todo el hardening en tu máquina virtual















Una vez realizada la aplicación manual de los controles de seguridad, usa **lynis** en la MV para evaluar su nivel de seguridad y ver si se ha conseguido mejorar si se compara con la evaluación inicial que realizamos.

Resultados esperados: Esta actividad finalizará cuando puedas obtener el nuevo nivel de seguridad de la máquina virtual securizada y compararlo con el original para ver si se ha mejorado.

INSIGNIAS Y AUTOEVALUACIÓN



NOTA: Tienes una versión de esta tabla de insignias en formato editable disponible en el *Campus Virtual*. Puedes usar este archivo para crear un documento en el formato que desees y tomar notas extendidas de tus actividades para crear un log de lo que has hecho. Recuerda que el material que elabores se puede llevar a los exámenes de laboratorio.

Nivel de Insignia	Desbloqueado cuando	¿Desbloqueado?
	Puedes responder a la siguiente pregunta: <i>¿Qué ventajas de seguridad crees que tienes al restringir el acceso al modo de usuario único (single user mode)?</i>	
	Conoces el significado del parámetro "password expiration"	
	Conoces el significado del parámetro "password minimum days"	
	Conoce el significado del parámetro "password expiration warning"	
	Conoce el significado del parámetro "inactive password lock"	
	Puedes responder a la siguiente pregunta: <i>¿Crees que los banners de advertencia son una medida de seguridad real? Si no, ¿por qué crees que se utilizan?</i>	
	Sabes cómo poner SHA-512 como el algoritmo hash de contraseña para ssh	
	Sabes cómo limitar la reutilización de contraseñas	
	Puedes responder a la siguiente pregunta: <i>¿Cuál es el propósito de umask y por qué 027 es un buen valor seguro predeterminado?</i>	
	Puedes responder a la siguiente pregunta: <i>¿Por qué es un problema tener varias cuentas de root? ¿Qué crees que pasó si encuentras múltiples root en tu sistema, pero no los has creado tú?</i>	
	Conoces dónde está el archivo de configuración SSH y comprendes la utilidad de sus diferentes parámetros de seguridad. Puedes responder a la siguiente pregunta: <i>¿Qué ventajas de seguridad crees que tiene al cambiar la configuración SSH predeterminada?</i>	
	Puedes responder a la siguiente pregunta: <i>Si cambiamos el puerto ssh predeterminado de 22 a otro, ¿qué ventaja de seguridad crees que podríamos tener?</i>	
	Comprendes por qué ser root es un problema de seguridad muy grave y por qué existen todas esas medidas que impiden los inicios de sesión root no autorizados.	
	Entiendes por qué las cuentas de servicios no deben usarse para inicios de sesión interactivos. Puedes responder a la siguiente pregunta: <i>si suponemos que por defecto no hay ninguna cuenta de servicio configurada para hacer inicios de sesión interactivos, ¿cuál podría ser la causa de encontrarnos de repente una configurada de esa manera?</i>	

	Puede responder a las siguientes preguntas: ¿Por qué es recomendable evitar contraseñas comunes como complemento a la aplicación de una política de contraseñas compleja? ¿Qué tipo de ataques previene esto? (¡recuerda los laboratorios anteriores! 😊)	
	Puedes responder a la siguiente pregunta: ¿Por qué es aconsejable deshabilitar los volcados de núcleo (core dumps) desde el punto de vista de la seguridad?	
	Puedes comprobar el estado de AppArmor y sabes cómo habilitar perfiles	
	Puedes responder a la siguiente pregunta: ¿Por qué detectar un consumo excesivo de recursos puede indicar un problema de seguridad?	
	Puedes responder a la siguiente pregunta: ¿Qué monitor de recursos te resulta más útil en tu MV? ¿Cuál usarías para monitorizar el uso de CPU?	
	Puedes responder a las siguientes preguntas: ¿Qué operación legítima específica crees que puede modificar un archivo del sistema y activar una alerta de seguridad de Tripwire? Como estas operaciones son legales, ¿qué debemos hacer si tenemos muchas de estas alertas?	
	Puedes mejorar la seguridad de la pila de red y optimizar los protocolos soportados por un host para mejorar su seguridad	
	Puede habilitar y configurar correctamente el demonio de auditoría de Linux (auditd)	
	Dominas los requisitos de calidad de password: entiendes los diferentes parámetros que controlan la calidad de una contraseña en el módulo PAM.	
	Puedes responder a la siguiente pregunta: ¿Por qué crees que no tener un compilador en un sistema local es positivo desde una perspectiva de seguridad? (PISTA: El malware también son programas, que tienen código fuente ... 😊)	
	Comprendes la importancia de verificar los logs y tener herramientas capaces de detectar comportamientos inusuales a través de su contenido.	
	Puedes responder a las siguientes preguntas: ¿El nivel de seguridad de la máquina virtual “reforzada” es superior al del sistema inicial? ¿Crees que lynis está alineado con el CIS (es decir, comprueba el cumplimiento de los mismos controles de seguridad)?	
	Puedes instalar y usar tripwire en una máquina virtual y comprendes la utilidad de esta clase de herramientas. Puedes responder a la siguiente pregunta: ¿Qué tipo específico de malware crees que puede detectar más fácilmente con ellos?	
	¡Esto es OSParta! Puedes aplicar y supervisar manualmente las configuraciones de seguridad más importantes en diferentes partes de un sistema operativo <i>Ubuntu Linux</i> siguiendo prácticas de seguridad verificadas y utilizando software típico para ello.	