Fuente: Bing Chat AI

Laboratorio 5. Seguridad de un SO Linux (No-Automatizable)

DEPARTMENTO DE INFORMÁTICA. UNIVERSIDAD DE OVIEDO

Seguridad de Sistemas Informáticos | 2023 – 2024 (V4.0 *"S-81 Isaac Peral"*)

Contenido

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 o clon enlazado de la MV.**
- 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 porque es enorme 😊.

🔍 ***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.**

Sobre *Wazuh*

Al final del laboratorio está la actividad "estrella" del mismo, la puesta en marcha de un **XDR/SIEM llamado *Wazuh***. Esta es una herramienta gratuita potentísima **de uso profesional** con un gran nº de funciones. Tanto es así que **volveremos a ella en los laboratorios 6, 7, 8-9 y 11**, y en este solo veremos una parte de sus funciones. Usarla tiene un precio no obstante, y es que necesitas aumentar los recursos de la MV que te damos a **6Gb de RAM y 4 cores para que funcione con agilidad**. Para ello te aconsejamos lo siguiente:

- Si no tienes en casa una máquina capaz de soportar la creación de una MV con estas características, te recomendamos que **le des prioridad a la actividad de *Wazuh* para hacerla en el laboratorio**, donde sí tienes *hardware* que soporte esa MV. **Puedes incluso empezar por ella sin problemas**.

🔍 ***Si haces también primero el ejercicio L5B4_AUDIT antes de clonar la máquina para crear la de Wazuh, este XDR podrá probablemente capturar más información de tu sistema, pero no es estrictamente necesario para este laboratorio.***

- Para esta actividad **se recomienda hacer un clon enlazado de la MV base** (mucho más ligero de crear y usar) y aumentarle los recursos solo a ese clon.
- De esta forma mantendrás dos MVs, pero solo tendrás que arrancar una de cada vez: **La MV "normal"** (con sus instantáneas o copias que quieras hacer y con la que trabajarás en los distintos labs) y su clon enlazado, **la MV "*Wazuh*"** con *Wazuh* instalado, que no necesita ninguna otra en funcionamiento. **Por favor, ten en cuenta que si borras la MV "normal" y has hecho un clon enlazado, perderás la**

MV "*Wazuh*" también. Si prevés que la vas a borrar, mejor haz una clonación completa de la MV, aunque ocupe más recursos.

- **No borres la MV *Wazuh* cuando la crees** porque la vamos a usar en otros, por favor.
- En este laboratorio la pondremos en marcha y solo veremos parte de sus características en su interfaz. Posteriormente volveremos a la misma interfaz para ver lo que significan otras cosas. Pero **es importante que hagas esta actividad con *Wazuh* primero** para enterarte bien del resto.

Bloque 1: Seguridad de los usuarios

Ejercicio L5B1_LYNIS1: Evaluación del nivel de seguridad actual

Descripción de la actividad / Aplicación práctica

Necesitas evaluar cuál es el **nivel de seguridad actual** de tu máquina *Linux*

Resultados Esperados

Esta actividad finaliza cuando puedes evaluar la seguridad de tu máquina virtual con `lynis` en su estado inicial

Otra información necesaria para su realización

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.

🔍 ***Lynis no es parte de los benchmark CIS, que tienen su propia herramienta de auditoría (CIS CAT). Ni siquiera usa exactamente los mismos controles de seguridad que los CIS, pero sí que es una herramienta de auditoría válida y muy rápida de usar, que podemos emplear para controlar cuánto hemos mejorado. También es una muestra de que no hay una sola fuente de controles de seguridad, y que las distintas fuentes priorizan distintas cosas***

Ejercicio L5B1_SINGLEUSER: Cuentas y acceso: Evitar el modo de usuario único (*single user mode*)

Descripción de la actividad / 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)

Resultados Esperados

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

Otra información necesaria para su realización

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...pero al entrar, estes virtualmente el usuario*** `root` 😟

Para evitar el arranque en modo *single user*, haz la tarea del *CIS benchmark* ***1.4.3 Ensure authentication required for single user mode***.

🔍 ***NOTA: El comando recomendado en el CIS Benchmark,*** `grep -Eq '^root:\$[0-9]' /etc/shadow || echo "root is locked"` ***no es correcto en la última revisión de Ubuntu 22.04. Han cambiado el algoritmo KDF por uno llamado "yescrypt". Este algoritmo ya no se referencia con un nº en el fichero*** `/etc/shadow` ***sino con el carácter "y". Cambia la expresión regular para reflejar esto.***

🛠 Ejercicio L5B1_BANNER: Banners de advertencia

🕵 Descripción de la actividad / Aplicación práctica

Necesitas decirles a los posibles intrusos "fuera de mi jardín" (como Clint Eastwood en "*Gran Torino*" 😊)

🎖 Resultados Esperados

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

📔 Otra información necesaria para su realización

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* ***1.7 Command Line Warning Banners.*** Concretamente, implementa solo estas tareas:

- ***1.7.1 Ensure message of the day is configured properly.***
- ***1.7.2 Ensure local login warning banner is configured properly.***
- ***1.7.3 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/

🛠 Ejercicio L5B1_SECSSH: Seguridad adicional para *OpenSSH*

🕵 Descripción de la actividad / Aplicación práctica

Necesitas **fortalecer las conexiones remotas vía SSH** todo lo posible

Resultados Esperados

Esta actividad finalizará cuando se **mejore la seguridad de la configuración del servicio SSH** de acuerdo con las especificaciones del CIS.

Otra información necesaria para su realización

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 subtareas.

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. El conjunto de tareas a implementar es el bloque ***5.2 Configure SSH Server*** del *benchmark*.

🔍 ***Otra forma de hacer el ejercicio es copiar el fichero de configuración, editarlo completo con todas las opciones de cada uno de los puntos con un editor potente como el Visual Studio Code y luego sobrescribir el existente. ¡Mucho mejor que ir probando las cosas 1 a 1!*** 😊

🛠 Ejercicio L5B1_PAM: Configuración segura del módulo PAM

Descripción de la actividad / Aplicación práctica

Necesitas **fortalecer tu política de *passwords*** todo lo que se pueda

Resultados Esperados

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

Otra información necesaria para su realización

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* ***5.4 Configure PAM*** y sus subtareas.

- ***5.4.1 Ensure password creation requirements are configured.***

- ***5.4.2 Ensure lockout for failed password attempts is configured.***
- ***5.4.3 Ensure password reuse is limited.***
- ***5.4.4 Ensure password hashing algorithm is SHA-512.***
- ***5.4.5 Ensure all current passwords uses the configured hashing algorithm.***

🛠 Ejercicio L5B1_PWDAGE: Antigüedad de la contraseña y cuentas inactivas

Descripción de la actividad / Aplicación práctica

Necesitas eliminar **cuentas no usadas y *passwords* viejas**

Resultados Esperados

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

Otra información necesaria para su realización

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* ***5.5.1 Set Shadow Password Suite Parameters*** y sus subtareas:

- ***5.5.1.1 Ensure password expiration is 365 days or less.***
- ***5.5.1.2 Ensure minimum days between password changes is configured.***
- ***5.5.1.3 Ensure password expiration warning days is 7 or more.***
- ***5.5.1.4 Ensure inactive password lock is 30 days or less.***
- ***5.5.1.5 Ensure all users last password change date is in the past***

🛠 Ejercicio L5B1_LOGONROOT: Restringir los inicios de sesión de cuentas de *root* y de servicios del sistema

Descripción de la actividad / Aplicación práctica

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

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.

Otra información necesaria para su realización

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* ***5.5.2 Ensure system accounts are secured*** y la tarea ***6.2.10 Ensure root is the only UID 0 account*** .

Ejercicio L5B1_COMMONPWD: Evitar el uso de contraseñas comunes

Descripción de la actividad / Aplicación práctica

Necesitas evitar que tus usuarios **usen password comunes** fáciles de crackear

Resultados Esperados

Esta actividad finalizará cuando **impidas el uso de contraseñas comunes** al cambiar o crear contraseñas en el sistema. Puedes contestar a estas preguntas:

- *¿Crees que puede haber una contraseña fuerte que sea común?*
- *¿Entiendes por tanto por qué esta medida de seguridad es un complemento a la de obligar a hacer contraseñas fuertes?*

Otra información necesaria para su realización

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.

Sí, esta actividad no forma parte de los benchmark CIS, pero se ha añadido porque, en nuestra experiencia personal, es un problema muy común

HACK FIRST
HACK HARD
NO MERCY
大和
Cobra Kai

Bloque 2: Seguridad de los procesos

Ejercicio L5B2_COMPILERS: Compiladores

Descripción de la actividad / Aplicación práctica

Necesitas **eliminar compiladores que tú no vas a usar**, pero sí que pueden usarse para desplegar *malware*

Resultados Esperados

Esta actividad finalizará cuando **deshabilites los compiladores de C/C++** en la MV, si existen, para dificultar que alguien compile programas no deseados en tu máquina

Otra información necesaria para su realización

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++
```

Ejercicio L5B2_COREDUMPS: Deshabilitar volcados de memoria del núcleo (*core dumps*)

Descripción de la actividad / Aplicación práctica

Necesitas **eliminar los ficheros de *core dump*** que se pueden usar para obtener información confidencial

Resultados Esperados

Esta actividad finalizará cuando se **impidan los volcados de núcleo** de acuerdo con las especificaciones del CIS. Puedes contestar a esta pregunta: *¿Tienes claro que tipo de cosas podrían contener esos archivos y por qué son peligrosos?*

Otra información necesaria para su realización

Implementa la tarea del *CIS benchmark* ***1.5.4 Ensure core dumps are restricted*** para deshabilitar esta posible vía de obtener información confidencial

Ejercicio L5B2_APPARMOR: Manejo de *AppArmor*

Descripción de la actividad / Aplicación práctica

Necesitas asegurarte de que el sistema **MAC *AppArmor* está activo** y confinando tu navegador *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*.

Otra información necesaria para su realización

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.
- ***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* ***1.6.1 Configure AppArmor,*** aunque más concretamente estas dos.

- ***1.6.1.1 Ensure AppArmor is installed***
- ***1.6.1.3 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
```

🛠 Ejercicio L5B2_SECAPT: Instalar software de seguridad *apt*

🥷 Descripción de la actividad / Aplicación práctica

Necesitas estar seguro de que **no se instalan paquetes corruptos**

🎖 Resultados Esperados

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

📱 Otra información necesaria para su realización

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 esto: https://manpages.ubuntu.com/manpages/trusty/man1/debsums.1.html que te permitirá comprobar la integridad de los paquetes que elijas.

🔍 ***Esta actividad no está en los benchmark CIS, pero si en los controles de seguridad de Lynis***

🛠 Ejercicio L5B2_PACCT: Contabilidad de procesos

🥷 Descripción de la actividad / Aplicación práctica

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

🎖 Resultados Esperados

Esta actividad finalizará cuando **este *software* de contabilidad esté instalado** y en ejecución.

📱 Otra información necesaria para su realización

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/pact`
- Activa contabilidad de procesos: `sudo accton /var/log/pacct`

🔍 ***Esta actividad no está en los benchmark CIS, pero si en los controles de seguridad de Lynis***

Bloque 3. Seguridad de redes en un host

Ejercicio L5B3_NETSTACK: Configuración de la pila de red de un servidor individual

Descripción de la actividad / Aplicación práctica

Necesitas configurar **la pila de red de tu máquina** de la forma más segura posible

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**.

Otra información necesaria para su realización

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 importan 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 Disable unused network protocols and devices***
  - *3.1.1 Ensure system is checked to determine if IPv6 is enabled*
  - *3.1.2 Ensure wireless interfaces are disabled*
- ***3.2 Network Parameters (Host Only)***
  - *3.2.1 Ensure packet redirect sending is disabled*
  - *3.2.2 Ensure IP forwarding is disabled*
- ***3.3 Network Parameters (Host and Router)***
  - *3.3.1 Ensure source routed packets are not accepted*
  - *3.3.2 Ensure ICMP redirects are not accepted*
  - *3.3.3 Ensure secure ICMP redirects are not accepted*
  - *3.3.4 Ensure suspicious packets are logged*
  - *3.3.5 Ensure broadcast ICMP requests are ignored*
  - *3.3.6 Ensure bogus ICMP responses are ignored*

- *3.3.7 Ensure Reverse Path Filtering is enabled*
- *3.3.8 Ensure TCP SYN Cookies is enabled*
- *3.3.9 Ensure IPv6 router advertisements are not accepted*

🛠 Ejercicio L5B3_RAREPROT: Deshabilitar protocolos de red poco comunes

Descripción de la actividad / Aplicación práctica

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

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.

Otra información necesaria para su realización

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*

Bloque 4: Log y monitorización

Ejercicio L5B4_AUDIT: Configurar reglas de auditoría

Descripción de la actividad / Aplicación práctica

Necesitas poner en marcha **un perfil de auditoría completo en tu SO *Linux***

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.

Otra información necesaria para su realización

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*

🛠 Ejercicio L5B4_SYSSTAT: Instalar *sysstat* para ver el consumo de recursos

Descripción de la actividad / Aplicación práctica

Necesitas **monitorizar el uso de recursos** en tu SO Linux

Resultados Esperados

Esta actividad finalizará cuando **puedas comprobar las estadísticas de uso** de la CPU usando este paquete.

Otra información necesaria para su realización

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.
- `cifsiostat` informa de estadísticas de CIFS (*Common Internet File System*).

🔍 ***Las herramientas de control de uso de recursos no son parte del benchmark CIS, pero se han añadido para poder tener una idea de lo que está pasando si en el transcurso de los laboratorios la MV se vuelve muy lenta sin razón aparente.***

🛠 Ejercicio L5B4_GLANCES: Instalar *glances*

Descripción de la actividad / Aplicación práctica

Necesitas **monitorizar el uso de CPU** de tu SO Linux

Resultados Esperados

Esta actividad finalizará cuando puedas **comprobar las estadísticas de uso de la CPU** usando este paquete.

Otra información necesaria para su realización

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: `sudo apt install glances` y usarlo ejecutando `glances` en la línea de comandos.

🔍 ***Las herramientas de control de uso de recursos no son parte del benchmark CIS, pero se han añadido para poder tener una idea de lo que está pasando si en el transcurso de los laboratorios la MV se vuelve muy lenta sin razón aparente.***

🛠 Ejercicio L5B4_LYNIS2: Evaluación del nivel de seguridad final con *Lynis*

Descripción de la actividad / Aplicación práctica

Sabes cómo **comparar los índices de seguridad** una vez has implementado todo el hardening en tu MV

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.

Otra información necesaria para su realización

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.

🛠 Ejercicio L5B4_WAZUH: HIDS gracias a una XDR profesional: *Wazuh* como monitorización de archivos, procesos y usuarios

Descripción de la actividad / Aplicación práctica

Consiste en que **aprendas a instalar un *software* XDR profesional** que pueda vigilar tu MV de forma que tengas una visión completa de lo que ocurre en ella, y que así entiendas parte de las funciones que puede realizar un XDR sobre una máquina, y su importancia en un despliegue real.

Resultados Esperados

Entiendes cómo el XDR *Wazuh*, entre sus muchas funciones, **puede funcionar como HIDS** y ayudarte a vigilar las anomalías que ocurren en tu MV. Puedes contestar a estas preguntas:

- *Ahora que sabes cómo se despliega un XDR/SIEM y sus agentes, ¿Cómo crees que funcionan las empresas que alquilan uno a una empresa de seguridad que se encarga de vigilar sus máquinas en busca de anomalías?*
- *¿Qué crees que quiere decir que Wazuh te reporte una ejecución de sudo exitosa?*
- *¿Y una que no ha tenido éxito?*

- *¿Qué puede significar que Wazuh detecte una instalación de software que no has iniciado tú? ¿Se te ocurre alguna posibilidad legítima de que eso ocurra? ¿E ilegítima? (vale, esta última es muy fácil 😊)*
- *Si pones a Wazuh a vigilar ficheros importantes del SO en ciertos directorios y detecta un cambio en la hash de alguna de ellas, ¿significa siempre que ha pasado algo malo?*
- *Por ejemplo, ¿qué razones legítimas y no legítimas puede haber para que haya un cambio en el fichero* `/etc/shadow` *o en el* `/etc/passwd`*?*
- *Aunque en este ejercicio vemos como Wazuh detecta comportamientos sospechosos, no estamos respondiendo a ellos de ninguna forma. Si te digo que en un SOC tienen un procedimiento de respuesta preparado para cualquiera de estas anomalías, de manera que cuando se detecta una y se confirma se pone en marcha dicho procedimiento para responder a ella (manual, automático o mixto), ¿Comprendes ahora más o menos cómo trabaja a grandes rasgos un SOC?*

Otra información necesaria para su realización

Un *Host-Based Intrusion Detection System* (HIDS) es un **sistema de detección de intrusiones** capaz de monitorizar y analizar **los componentes internos** de un sistema informático. Este tipo de software puede monitorizar todo o parte del comportamiento dinámico y el estado de un sistema informático, dependiendo de cómo esté configurado.

🔍 ***Por ejemplo, un HIDS podría detectar qué programa accede a qué recursos y descubrir que, por ejemplo, un procesador de textos ha comenzado de repente e inexplicablemente a modificar la base de datos de contraseñas del sistema, lo que es indicio de una infección en marcha.***

Además, un HIDS podría examinar el estado de un sistema, su información almacenada, ya sea en RAM, en el sistema de archivos, archivos de log, etc.; y verificar que el contenido de estos es el que se espera, es decir, que no haya sido cambiado por intrusos o de forma inesperada. Para ello, recurre a la creación de **firmas criptográficas** (como las del **Tema 2** de teoría) de los ficheros que le mandemos monitorizar, de manera que se detecte cualquier cambio en los mismos.

En comparación con los sistemas de detección de intrusiones basados en red, los HIDS tienen la ventaja de **poder identificar ataques internos**. Los NIDS examinan el tráfico de red, mientras que los HIDS examinan los datos que se originan en los sistemas operativos.

Es importante mencionar que los HIDS suelen hacer todo lo posible para evitar que sus bases de datos de objetos, *checksums* y sus informes sufran cualquier forma de manipulación. Si los intrusos lograran modificar cualquiera de los objetos que los HIDS monitorizan, también podrían modificar los datos que usa un HIDS.

En los *CIS Benchmarks* hay un apartado para instalar un HIDS, debido a que su presencia eleva considerablemente el nivel de seguridad de un sistema. En este caso, el CIS ha optado por un HIDS **que no depende de ningún componente externo** para funcionar y que funciona dentro del mismo host, llamado ***AIDE***. Concretamente, su instalación se cubre en estos apartados:

- ***1.3 Filesystem Integrity Checking***
  - ***1.3.1 Ensure AIDE is installed (Automated)***
  - ***1.3.2 Ensure filesystem integrity is regularly checked (Automated)***

Un HIDS muy similar a este, que a veces se usa en sustitución de *AIDE*, es ***Tripwire*** (https://linuxconfig.org/intrusion-detection-systems-using-tripwire-on-linux). No obstante, estos HIDS,

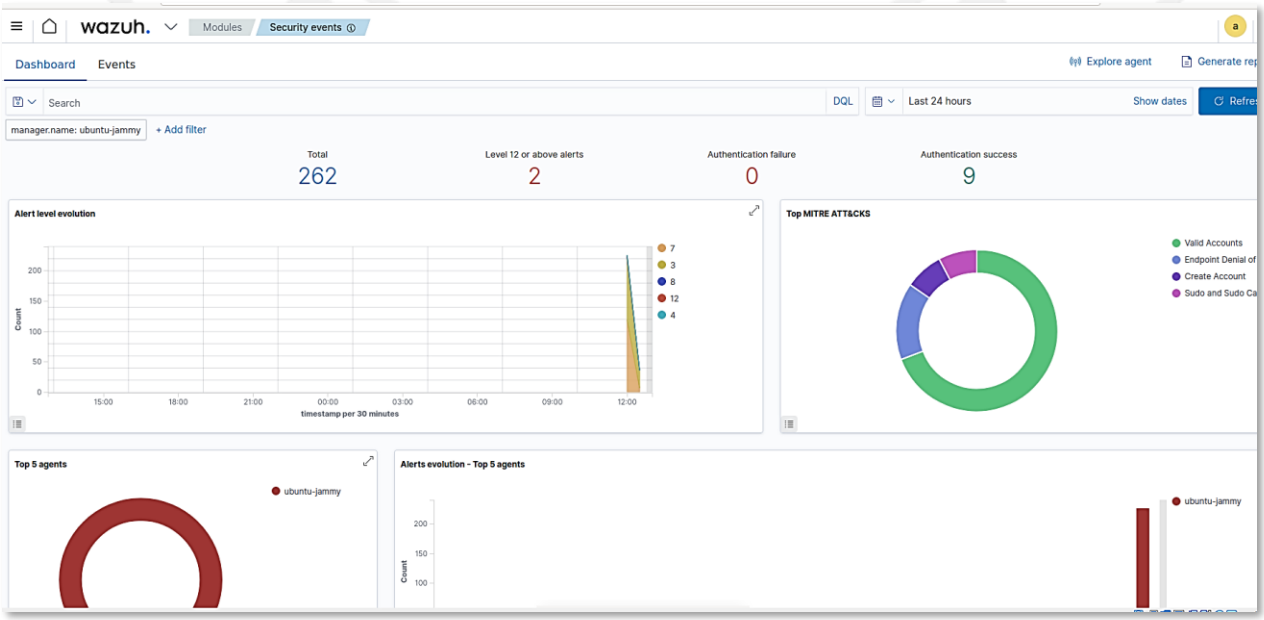
precisamente por no depender de nada externo, tienen una serie de funcionalidades muy limitadas (básicamente detectar cambios en ficheros importantes). *Wazuh* les supera ampliamente en funciones, como veremos, al ser un XDR que incluye funciones de HIDS. En las siguientes secciones veremos como ponerlo en marcha en nuestra infraestructura, lo que cubriría de sobras el punto 1.3 completo de los *benchmark* CIS.

🔍 ***Wazuh no es parte de los benchmarks CIS, pero como es una herramienta que cubre las mismas necesidades y muchas más que usaremos en laboratorios posteriores, por lo que la vamos a usar en su lugar.***

Instalación de *Wazuh single-node* en una MV usando *Docker*

Para poner en marcha *Wazuh* sin problemas, te recomendamos seguir estos pasos:

- **Haz el clon enlazado de la MV** de la asignatura como te indicamos al principio del laboratorio, **con 6Gb de RAM y 4 *cores*** (¡la potencia tiene un precio! 😊). Apaga tu MV normal y arranca esta.
- Hay varias formas de instalar *Wazuh*, pero la mejor en nuestro caso es ***Wazuh-Docker***, que lo instala usando el mismo sistema que nosotros usamos en la asignatura para desplegar los laboratorios. Las instrucciones concretas las puedes encontrar aquí: https://documentation.wazuh.com/current/deployment-options/docker/wazuh-container.html, y de ellas tienes que asegurarte de lo siguiente:
  - El tipo de instalación que vamos a hacer es ***Single-node deployment***. Es igual de funcional que la otra, pero con todo en una máquina (no tenemos recursos para hacer una instalación distribuida).
  - En el paso de generar los certificados, nos da dos opciones: **generarlos o aportarlos nosotros**. En este caso es mejor **generarlos** con el *software* que nos proporcionan (`generate-indexer-certs.yml`). Asegúrate de ejecutarlo una vez antes de lanzar *Wazuh* como indica la web. No es necesario cambiar ningún fichero de los que proporcionan.
- Lanzar ahora el XDR como indica el tutorial, yendo al directorio donde has descomprimido los contenidos del *git* y tecleando `docker-compose up -d`. Por favor, espera unos 30 segundos hasta que todo el sistema se ponga en marcha. Cuando todo esté listo, deberías poder acceder al XDR abriendo el navegador de la MV y yendo a la dirección https://127.0.0.1. La cuenta de usuario por defecto es "`admin`" y la password "`SecretPassword`". Si todo va bien, deberíamos esta pantalla:



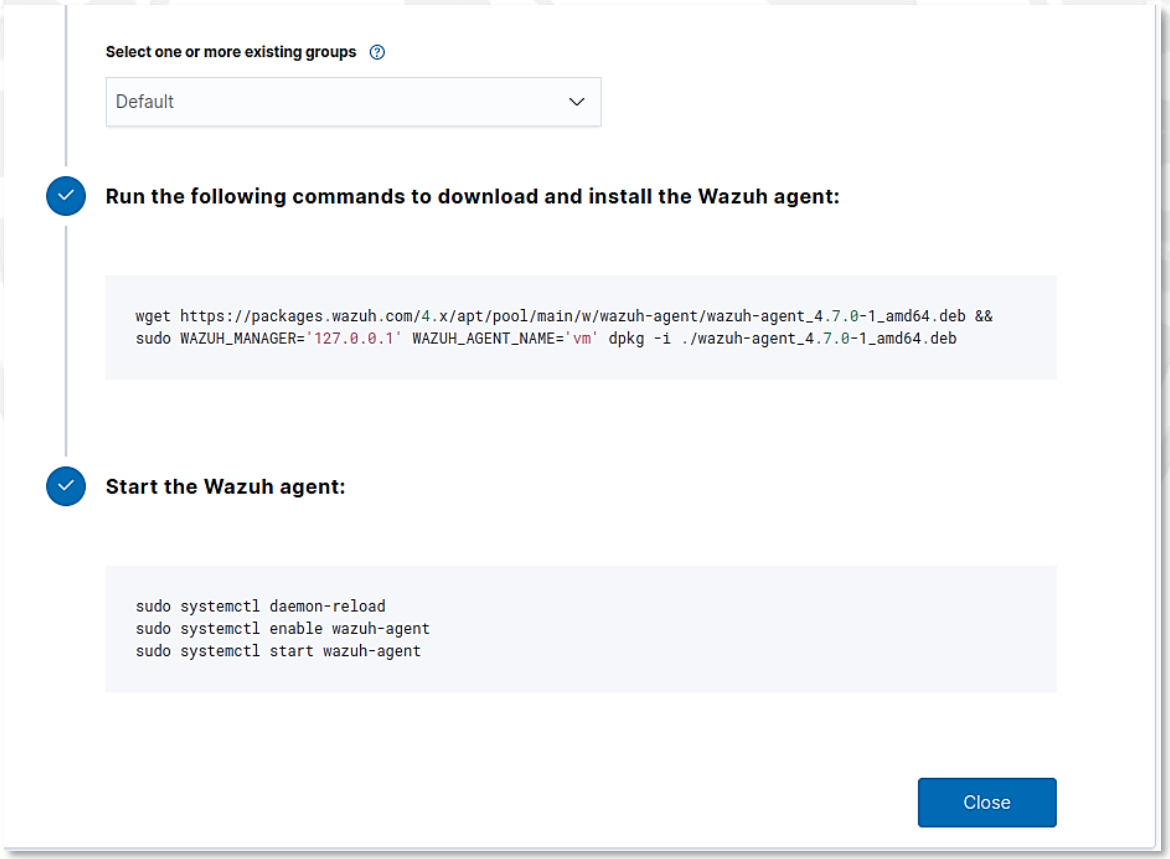
🔍 ***Si te decides a desplegar esto en un entorno real, ¡por favor no te olvides de cambiar la password de admin!***

Wazuh funcionando como HIDS

Wazuh es ahora mismo un sistema de vigilancia, pero ahora mismo **no tiene máquinas que vigilar**. De hecho, nos avisa diciendo que **no tiene agentes**. Un agente es un *software* que desplegamos en cada máquina vigilada y que se encarga de:

- **La recopilación de información** del sistema en el que está instalado
- Estar configurado para recopilar **diferentes tipos de información** (cosas como los diferentes *logs de* las acciones que se producen, firmas de archivos... todo lo que necesitas)
- **Enviárselo al XDR** mediante comunicaciones cifradas fuertes. De hecho, si juegas un poco con la GUI puedes ver el algoritmo que usa para enviar la información. *¿Sabes qué algoritmo utiliza? ¿Es un cifrado simétrico o asimétrico?* (**Laboratorios 3-4**)

Entonces el XDR interpreta la información para detectar anomalías. *¿Cómo se despliega un agente?* No hay problema, porque ***Wazuh* nos los genera por nosotros**. Hacemos clic en el mensaje que se queja de que no hay agentes y configuramos uno nuevo para monitorizar nuestra propia MV, seleccionando la creación de un agente `.deb` de 64 bits para *Ubuntu* con estas características:



Ahora simplemente, desde una terminal de nuestra MV **copiamos los comandos que nos da el propio *Wazuh*** para instalarlo y arrancarlo. Con esto, *Wazuh* debería mostrar que tiene **1 agente activo** y empezará a recopilar información de la propia máquina que lo ejecuta. Ni que decir tiene que monitorizar más máquinas es lo mismo, y solo tendríamos que indicar la IP donde está la MV en la que has arrancado *Wazuh* al generar el agente, obviamente asumiendo que dicha IP es alcanzable desde la máquina a vigilar.

🔍 ***¿Entiendes por tanto lo tremendamente fácil que es monitorizar toda una infraestructura de máquinas con un solo Wazuh? El principal problema es que a más máquinas monitorizadas más recursos necesitas, pero para una instalación pequeña es una solución muy buena. Por cierto, por si te lo preguntas, si, Wazuh tiene agentes para Windows y MAC también*** 😉 **. También puedes considerar que los recursos consumidos por *Wazuh* son demasiados, pero si con una única instalación central puedes monitorizar varias máquinas, ya no es un problema tan grave**

Wazuh, para ahorrar recursos, **no viene con todas sus funcionalidades activas**. Una de las que viene sin activar nos interesa, y es la monitorización de cambios en ficheros importantes (***File Integrity Monitoring*** o FIM) y su posterior notificación si se detecta alguno. Para esto *Wazuh* usa un *software* que va integrado en sus agentes llamado **OSSEC**. OSSEC por defecto viene con esta funcionalidad desactivada y debes activarla cambiando su fichero de configuración (`/var/ossec/etc/ossec.conf`) en la MV de esta forma:

```
  GNU nano 6.2                  /var/ossec/etc/ossec.conf *                 I
 93      <interval>12h</interval>
 94      <skip_nfs>yes</skip_nfs>
 95    </sca>
 96
 97    <!-- File integrity monitoring -->
 98    <syscheck>
 99      <disabled>no</disabled>
100
101      <!-- Frequency that syscheck is executed default every 12 hours -->
102      <frequency>43200</frequency>
103
104      <scan_on_start>yes</scan_on_start>
105
106      <!-- Directories to check  (perform all possible verifications) -->
107      <directories realtime="yes" check_all="yes">/etc,/usr/bin,/usr/sbin</di>
108      <directories>/bin,/sbin,/boot</directories>
109
110      <!-- Files/directories to ignore -->
111      <ignore>/etc/mtab</ignore>
112      <ignore>/etc/hosts.deny</ignore>
          [ line 107/220 (48%), col 23/86 ( 26%), char 2926/6043 (48%) ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

Con esta configuración le estamos diciendo a OSSEC que monitorice todos los cambios en tiempo real de los directorios `/etc`, `/usr/bin` y `/usr/sbin`, que contienen programas ejecutables críticos del SO. Más información:

- https://www.ossec.net/docs/manual/syscheck/index.html
- https://documentation.wazuh.com/current/user-manual/capabilities/file-integrity/how-to-configure-fim.html

🔍 ***Podemos monitorizar más directorios, e incluso ficheros sueltos, añadiendo más entradas al fichero. Todo depende de lo que nos interese y lo que haya en el sistema, por supuesto. ¡Es cuestión de probar y experimentar!*** 😊

Hecho esto, solo tenemos que **reiniciar el agente en la VM** (`sudo systemctl wazuh-agent restart`) y ya se podrán detectar cambios en los ficheros de esos directorios. Ahora ya tenemos *Wazuh* listo para funcionar como HIDS, por lo que para terminar el ejercicio haremos las siguientes pruebas en la MV, comprobando luego en el GUI de *Wazuh* (pulsa en *Refresh*) si se detectan:

- Haz un `sudo` a un comando, pero **mete mal la clave** para que falle
- Haz un `sudo`, pero esta vez **correcto**
- Instala algo con `apt` o actualiza el SO
- Edita el fichero `/etc/passwd` y cambia el *shell* del usuario `root` de `nologin` a `/bin/bash`. *¿Qué crees que hace esto? ¿Para qué podría usarse, si en lugar de* `root`*, fuera otro usuario más "inocente"? ¿Lo detecta Wazuh?*

🔍 ***Recuerda: ¡Conserva esta MV para futuros laboratorios!***

✍ 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: *¿Qué ventajas de seguridad crees que tienes al restringir el acceso al modo de usuario único (single user mode)?*	
	Conoces el significado del parámetro "*password expiration*"	
	Conoces el significado del parámetro "*password minimum days*"	
	Conoces el significado del parámetro "*password expiration warning*"	
	Conoces el significado del parámetro "*inactive password lock*"	
	Puedes responder a la siguiente pregunta: *¿Crees que los banners de advertencia son una medida de seguridad real? Si no, ¿por qué crees que se utilizan?*	
	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: *¿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?	
	Conoces dónde está el archivo de configuración SSH y comprendes la utilidad de sus diferentes parámetros de seguridad. Puedes responder a la siguiente pregunta: *¿Qué ventajas de seguridad crees que tiene al cambiar la configuración SSH predeterminada?*	
	Puedes responder a la siguiente pregunta: *Si cambiamos el* puerto `ssh` predeterminado *de* `22` *a otro, ¿qué ventaja de seguridad crees que podríamos tener?*	
	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: *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?*	

	Puedes 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 `wazuh` en una máquina virtual y comprendes la utilidad de los XDR a la hora de monitorizar las actividades del usuario procesos y uso de ficheros de una máquina. 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 *Ubuntu Linux* siguiendo prácticas de seguridad verificadas y utilizando software típico para ello, así como poner en marcha un sistema de vigilancia profesional que incluye un HIDS en tiempo real.	