TEMA 3
SEGURIDAD DE SISTEMAS OPERATIVOS

Distintas áreas en las que fortalecer la seguridad de un sistema operativo

José Manuel Redondo López Proyecto *"S-81 Isaac Peral"* v4.0

Fuente: Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

Logo: @creative_vanesa (*Instagram*)

Objetivos generales del tema

- **Entender cómo mejorar manualmente la seguridad de un SO**
  - Cuáles son las áreas básicas del sistema operativo para hacerlo
- **Aprender qué áreas requieren un control y tratamiento especial**
- **Conocer software de protección especializado para ciertas áreas de los sistemas operativos**
- **Evaluar la seguridad de un sistema operativo determinado**
- **Y relacionarse con otros cursos**
  - Complementar los contenidos de ASR desde el punto de vista de la seguridad
  - Enlazar con el final de SO para ver "qué más hay" con respecto a la seguridad del sistema operativo

ÍNDICE

< Ir al índice

La "Cadena de seguridad"

La seguridad como un esfuerzo combinado

*Iconos por Bing Chat AI

Fuente: Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

¿QUÉ TE VAS A ENCONTRAR EN ESTE BLOQUE?

En este bloque aprenderás

- Los elementos implicados en una "**cadena de seguridad**" típica
- Por donde se puede iniciar un ataque **usando el software de un sistema como vehículo**
- El problema de los **diseños inseguros** de dispositivos o funcionalidades
- El problema del **exceso de privilegios** de un usuario o programa
- Que los usuarios son muchas veces el **eslabón más débil** de la cadena

La seguridad como una "cadena"

- El nivel de seguridad de un sistema no se obtiene sumando todas las medidas de seguridad implementadas en una organización...
- **... sino comprobando el nivel de seguridad del elemento "más débil"** de esta organización
- Estos elementos son
  - Aplicaciones en uso
  - Diseño de software
  - Sistemas operativos
  - Usuarios (¡y muchas veces son el eslabón más débil!)
  - El contexto del **lugar físico** donde están los equipos (**Seminario 2**)

DEFECTOS DE SEGURIDAD EN EL SOFTWARE

- El **malware** aprovecha **defectos** (bugs o vulnerabilidades) del **software**
  - Apuntando a bugs o loopholes **para explotarlos**
  - Aplicaciones **obsoletas** (Ej.: Internet Explorer, programas sólo compatibles con Windows XP...)
  - Aplicaciones con **vulnerabilidades conocidas**
    - Ej.: versiones vulnerables de plugins de navegador como Acrobat Reader, Java SE...
- A veces, la instalar versiones actualizadas no desinstala las antiguas
  - ¡Así que el problema de seguridad sigue ahí!
- Hay que monitorizar los avisos de seguridad de los proveedores de aplicaciones
  - Indican actualizaciones que solucionan los problemas documentados en las BBDD de CVE (**Tema 1**)
- **Mantener un control adecuado de las aplicaciones instaladas en cualquier sistema operativo y sus parches es vital ("tener un inventario")**
  - **SOLO instala las aplicaciones que sean estrictamente necesarias** en cualquier SO
  - **Ídem con dependencias/frameworks y librerías** de las aplicaciones (herramientas SCA, **Tema 6**)

SISTEMAS OPERATIVOS

- **La homogeneidad en el uso de SO también puede ser una vulnerabilidad**
  - Cuando todos los equipos en una red ejecutan el mismo SO, si se explota uno, todos los demás también podrían caer
  - Ej.: ataque EternalBlue + WannaCry en 2017
- **Pero introducir diversidad de SO por robustez también tiene efectos no deseados**
  - Produce complejidad de mantenimiento, administración y autenticación, con aumento de costes...
- **Lo que podríamos hacer es asegurarnos de que todos los dispositivos no dependan del mismo servicio de autenticación**
  - Ej.: dominio de Active Directory
  - Su diversificación podría evitar un paro total si el servicio se ve comprometido o eliminado
  - Permite que algunos dispositivos puedan ayudar a recuperar los nodos deshabilitados
  - Da opciones de recuperación en ataques de malware o ransomware
- **La instalación del SO debe hacerse con la configuración más segura (Tema 4)**
  - ASR cubre la instalación básica de SO Linux y Windows

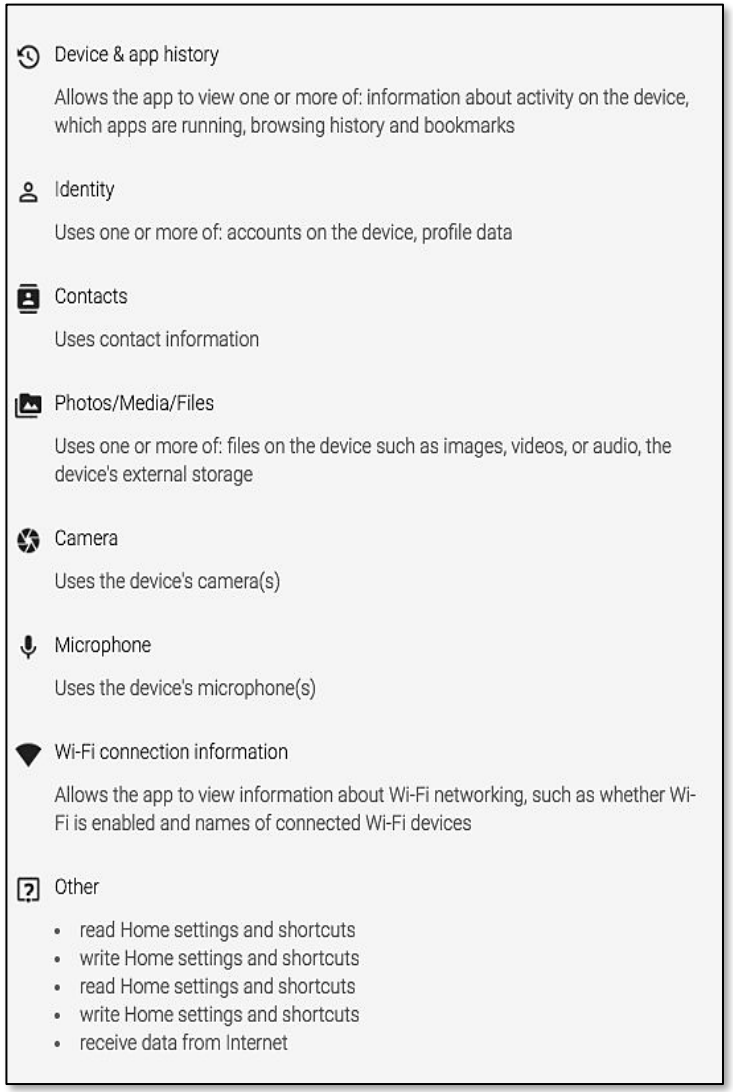
Diseño inseguro de funciones o errores de usuario

- **Hay operaciones legítimas comunes diseñadas sin tener en cuenta la seguridad**
  - Los atacantes suelen engañar a usuarios para que las hagan y explotar sus sistemas. Ej.:
    - Estafa de vishing de Microsoft y RAT (Remote Administration Tools)
    - El soporte de Asistencia remota incrustado de Windows (estafa de la "llamada de Microsoft")
- **Un ejemplo es el problema de arranque / ejecución desde dispositivos externos**
  - Las PCs arrancan desde HDD / SSD, pero pueden configurarse para hacerlo desde un dispositivo USB
    - También sucede cuando **se ejecuta software automáticamente** al insertar hardware externo (USB)
  - Estos mecanismos fueron diseñados como características útiles, pero sin pensar en la seguridad
    - Distribuidores de malware engañan al usuario **para arrancar o ejecutar algo desde dispositivos infectados**
    - **Cualquier dispositivo que se conecte a un puerto USB** se puede utilizar para propagar malware
      - Incluso luces, ventiladores, altavoces, juguetes... ¡ten MUCHO cuidado!
    - Los dispositivos pueden infectarse durante su fabricación o envío (**ataque a la cadena de suministro**)
  - Esto se puede evitar configurando el SO para que arranque solo desde el disco duro interno
    - Y **bloqueando las funciones de ejecución automática de los dispositivos**
- **Clientes de e-mail vulnerables podrían ejecutar adjuntos maliciosos**

Usuarios y Código con privilegios excesivos

- El **privilegio** mide cuánto puede modificar un sistema un usuario o programa
- En sistemas mal diseñados, a los usuarios y programas **se les pueden asignar más privilegios** de los necesarios
  - **Usuarios con privilegios excesivos**: Algunos sistemas permiten a todos los usuarios modificar las estructuras internas del SO
    - No hay ninguna distinción entre un `Administrador` / `root` y un usuario normal, o es pobre
  - **Código con privilegios excesivos**: los sistemas permiten a los programas acceder y usar todos los privilegios y permisos del usuario que los ejecuta
    - El malware que se ejecute con privilegios excesivos puede usarlos para atacar o dañar el sistema
    - Casi todos los SO populares y muchas aplicaciones dan demasiados privilegios a partes de su código
      - Se debe a errores o prácticas de desarrollo seguro inadecuadas (Tema 6)

Fuente: Permisos de una App Android



LOS USUARIOS COMO FUENTES DE RIESGO POTENCIAL

- Los usuarios **pueden ser la mayor amenaza** para la seguridad de una organización, ya sea consciente o inconscientemente
  - Problemas con la operación, manejo y administración de información o software
  - Abrir archivos adjuntos de correo electrónico maliciosos
  - Copiar documentos importantes en lugares con seguridad inadecuada / accesibles al público
  - No bloquear sesiones al dejar el equipo desatendido
  - Pérdidas de información importante
  - ...
- Algunos de estos ataques son por negligencia o falta de formación, pero otros pueden ser "**trabajos internos**" (ataques deliberados)
  - **¡Hay estudios que determinan que la mitad de los ataques a una organización provienen de sus empleados!**

Los usuarios como problemas: Contramedidas

- **Formación**
  - Cada usuario debe recibir **formación** sobre los recursos de la empresa y cómo pueden o no pueden ser utilizados
  - **Se debe desarrollar un conjunto de reglas de gestión para crear una directiva de seguridad** (**Tema 4**)
  - **Que incluya consecuencias** para los usuarios que no cumplen con esas reglas
- **Mejora de la seguridad del SO (este tema)**
  - Configurar correctamente un SO para detener las actividades peligrosas del usuario (ya sean intencionadas o no)
- **Crear un Sistema de Gestión de Seguridad de la Información** (SGSI) (**Tema 4**)
  - Que describa una política de seguridad y cómo se reforzarán los activos contra ataques
  - Que detalle qué hacer en caso de ataques, y qué pasos se deben realizar para recuperarse de ellos
  - Que fuerce a la realización de auditorías periódicas para comprobar el cumplimiento de las normas establecidas
  - Y con pleno consenso de todos los empleados de la organización

¿Crees que lo has entendido todo? ¡Autoevalúate!

?

- **Eres capaz de hacer lo siguiente**
  - *Entender que la seguridad no depende de una sola cosa, sino de las aplicaciones en funcionamiento (incluyendo el SO), cómo estén diseñadas y de las acciones de los usuarios*
  - *Comprender que mantener un inventario actualizado de todas las aplicaciones usadas es imprescindible para una seguridad adecuada*
    - *Incluyendo minimizar las aplicaciones que se instalan en cualquier escenario*
  - *Tener claras las ventajas y desventajas de tener un mismo SO y/o servicio de autenticación en toda una infraestructura*
  - *Comprender que algunas funciones se han diseñado en origen sin la seguridad en mente, y que ahora estamos sufriendo las consecuencias de ello*
  - *Tener claro que, como desarrollador, es muy tentador dar "permisos para todo" para evitar problemas de ejecución, pero que al hacerlo estamos abriendo la puerta a problemas de seguridad graves*
  - *Pensar en los usuarios como amenazas potenciales siempre, ya sea por descuido (mitigable con formación) o de forma deliberada (mitigable con una adecuada política de seguridad)*

< Ir al índice

Fuente: Bing Chat AI

Arranque >

Logs >

HARDENING DE SO GENERAL

Primeros pasos para mejorar la seguridad de nuestro SO

*Iconos por Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

¿QUÉ TE VAS A ENCONTRAR EN ESTE BLOQUE?

En este bloque aprenderás

- Que el **proceso de arranque** requiere especial protección porque puede darle a alguien acceso total al sistema
- Que en una instalación "seria", habitualmente hay **particiones asignadas** a distintas partes del sistema de ficheros
- La **importancia de la ubicación** de los ficheros y de las restricciones de seguridad que pueda haber a nivel de kernel
- La enorme importancia a nivel de seguridad que tienen los **logs de un sistema** y su correcta configuración, interpretación y estudio para una adecuada vigilancia
- Que para implementar vigilancia adecuadamente necesitas **HIDS** (Host-Based Intrusion Detection Systems)

Seguridad en el proceso de arranque

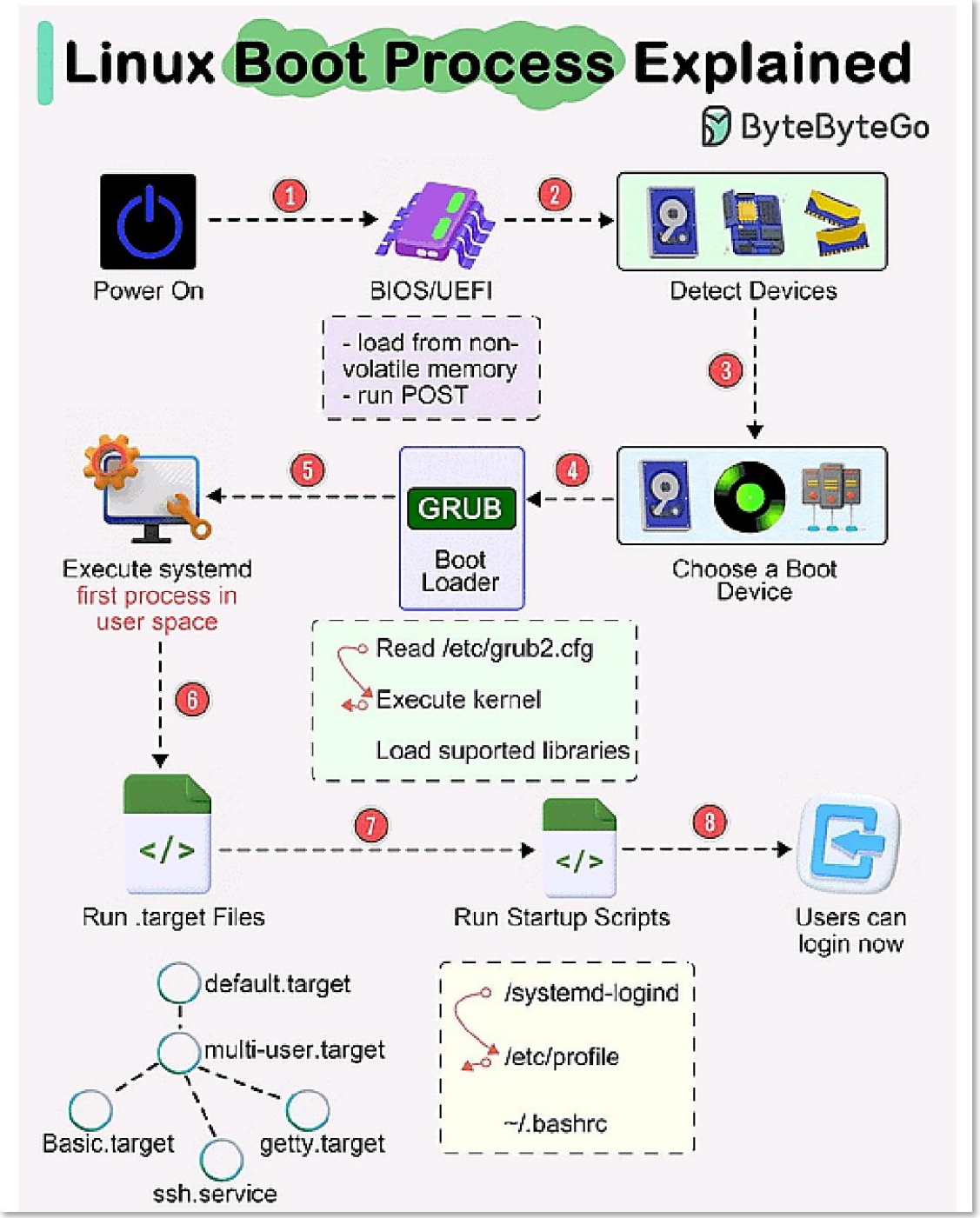
¡No queremos empezar ver el mundo arder! ☺

***Iconos por Bing Chat AI**

Fuente: Bing Chat AI

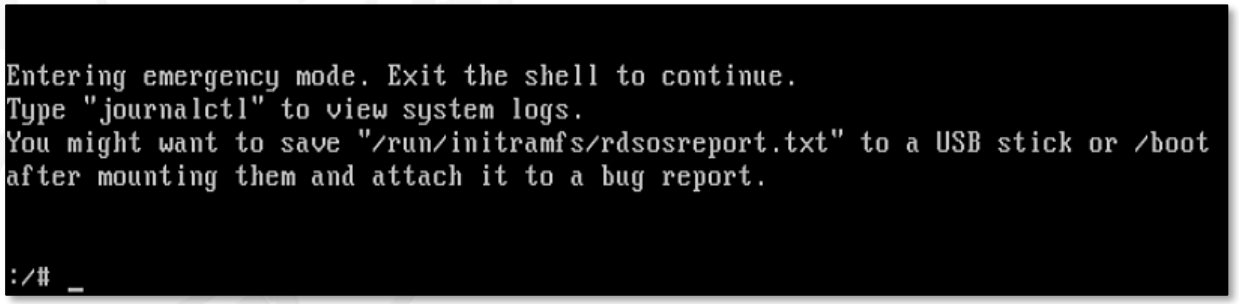
El proceso de arranque de un SO

- Es un proceso de varias etapas donde se carga todo lo necesario para trabajar
- Si se interrumpe o interviene, hay formas de conseguir acceso a la máquina con privilegios elevados
  - Modos de mantenimiento o single-user
- Por tanto, su protección es de importancia capital
  - O podemos vernos comprometidos antes de ver la pantalla de login...



Establecer una password de arranque / al bootloader

- El proceso de arranque de un SO (BIOS / UEFI) se describe en detalle en ASR
  - También cubre los detalles del gestor de arranque
- Las contraseñas de arranque evitan que usuarios no autorizados con acceso físico al sistema obtengan privilegios de `root`
  - A través del **modo de usuario único/mantenimiento (single user / maintenance mode)** mencionado
- Estos modos generalmente permiten operaciones peligrosas como
  - Cambio de la configuración del arranque
  - Acceso a la consola del gestor de arranque
  - Restablecer la contraseña de `root`
  - Acceso a SO no seguros (Live DVD)
  - ...



- Tampoco queremos que los usuarios vuelvan a habilitar estos modos
  - Establecer permisos adecuados garantiza que solo los usuarios con privilegios puedan modificar parámetros de arranque importantes

Particionamiento de discos

- **Los sistemas de archivos críticos deben separase en particiones distintas**
  - Las particiones y los sistemas de archivos se explican en ASR
- **Varios directorios del sistema de nivel superior deben colocarse en su propia partición física o volumen lógico**
  - Garantiza la separación y protección de datos
  - Evita el desbordamiento de particiones
  - Aísla una posible corrupción de datos
  - Proporciona separación lógica de datos
  - Disminuye la duración de `fsck`
  - Permite utilizar diferentes sistemas de archivos, adaptados a diferentes necesidades
  - Permite utilizar contextos SELinux específicos
- **Ej.: estos filesystems Linux deben montarse en particiones separadas**
  - `/home` (datos personales de cada usuario)
  - `/var` (archivos escritos mientras se ejecuta el SO)
  - `/var/tmp`
  - `/tmp`
    - Archivos temporales "desechables"
  - `/var/log`
  - `/var/log/audit`
    - Archivos de log, lo que pasa en ejecución del SO
  - Si el tamaño de una partición crítica no se calcula correctamente, ¡el SO puede no funcionar!
- **Windows tiene una partición de recuperación aislada**
  - Los datos del usuario deben estar en otro

RESTRINGIR OPCIONES DE MONTAJE

- **Algunos SO permiten marcar particiones con propiedades de seguridad. Ej.: en Linux:**
  - `nodev`: especifica que el sistema de archivos no puede contener dispositivos especiales
    - Un sistema de archivos accesible por todo el mundo no debe permitir crear dispositivos de caracteres o acceder al hardware de un dispositivo
  - `nosuid`: especifica que el sistema de archivos no puede contener archivos con el permiso `setuid` habilitado
    - Evitar **ejecutables SETUID** en un sistema de archivos con permisos de escritura para todos
    - Se pueden usar para escalar privilegios (explicado en **tema 7**)

```
-rwsr-xr-x  1 redondo redondo 138208 sep 14 10:50 ls
```

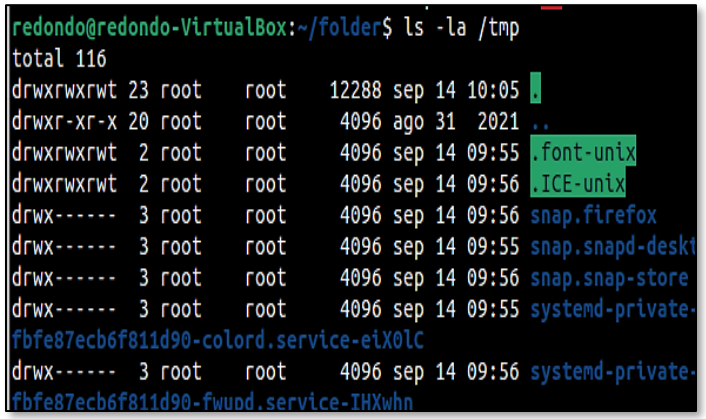
  - `noexec`: útil para particiones sin binarios (como `/var`)
    - O que contienen binarios que no deben ejecutarse...

- **Ejemplo**
  - El directorio de arranque de un sistema Linux contiene archivos importantes relacionados con el kernel
  - Debería **restringirse a permisos de solo lectura** y marcarse con las tres propiedades anteriores

```
 GNU nano 2.2.4            Archivo: /etc/fstab

# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    nodev,noexec,nosuid 0       0
# / was on /dev/sda1 during installation
/dev/sda1       /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda2 during installation
/dev/sda2       none            swap    sw              0       0
```

LOCALIZACIÓN DE FICHEROS TEMPORALES

- **Las ubicaciones de archivos temporales suelen tener permisos de escritura para todo el mundo de forma predeterminada**
  - Como `/tmp` y `/var/tmp` en **Linux**
  - Varios **daemons** / aplicaciones los usan para almacenar temporalmente datos, información de log...
    - O para compartir información entre sus subcomponentes
  - Pero debido a su naturaleza compartida, posibilitan varios ataques
    - Fugas de datos confidenciales si estos archivos contienen secretos
    - Ataques a la integridad de procesos y datos con **race conditions** (TPP)
    - Ataques DoS debido a **race conditions** (TPP), etc.
    - ...



- **Las aplicaciones maliciosas generalmente escriben en directorios temporales y luego intentan ejecutar lo que se escribieron**
  - Por lo tanto, es aconsejable evitar la ejecución de software desde directorios temporales
  - **Linux**: montar `/tmp` en una partición separada con las opciones `nodev`, `nosuid` y `noexec` habilitadas
    - Esto denegará la ejecución de binarios desde `/tmp`, impedirá activar el bit `suid root` de cualquier binario y deshabilitará la creación de cualquier dispositivo de bloques

Otras consideraciones

- **Espacio de swap**

```
-rw-------   1 root root 2147483648 ago 31  2021 swapfile
```

  - Cuando un SO se queda sin RAM, intercambia partes de ella a un archivo o partición de swap (SO)
  - Esta tiene partes de la RAM del proceso, que **puede contener información crítica** como contraseñas, tokens privados o datos con acceso restringido
  - Por lo tanto, es muy recomendable **cifrar el archivo de intercambio** o partición para evitar fugas de información crítica

- **Módulos del kernel**
  - Las vulnerabilidades de seguridad en el código del kernel no son frecuentes
    - ¡Pero sus consecuencias pueden ser dramáticas!
  - Por ello, no deben cargarse módulos del kernel si no se utilizan (Ej.: USB, Bluetooth ...)
    - Para proteger el sistema de la explotación de cualquier defecto de implementación que tengan
  - Y esto nuevamente se hace instalando **solo** el software / controladores que **realmente** necesitemos

- **La pila de red (lo que gestiona las conexiones) debe configurarse correctamente**
  - Incluyendo **desactivar el soporte para cualquier protocolo no usado**

- **La administración general del disco, las cuotas, etc. se explican en ASR**

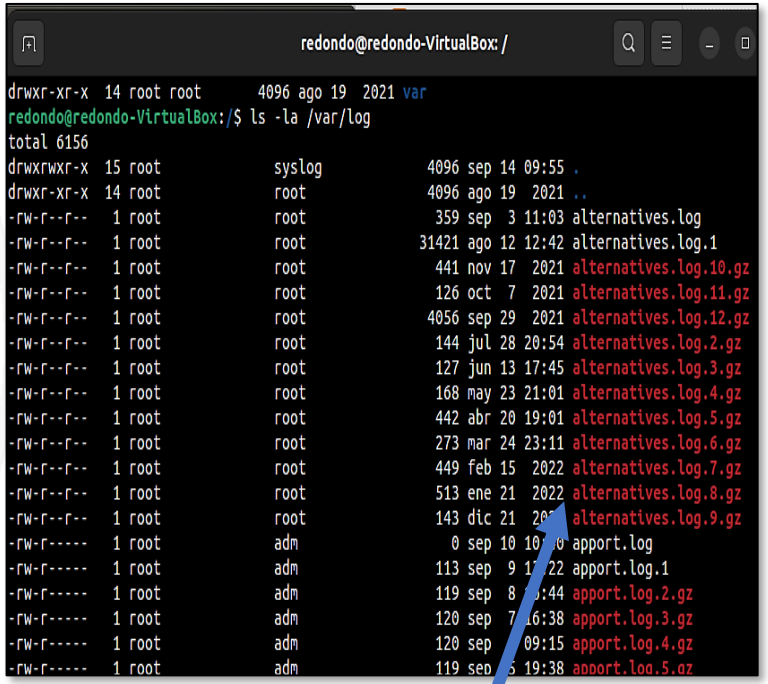
Log y Auditoría

Registrando lo que ha pasado

*Iconos por Bing Chat AI

IMPLEMENTAR UNA CONFIGURACIÓN DE AUDITORIA ADECUADA

- El registro de las actividades del sistema y los eventos relevantes para la seguridad es VITAL
- Tener una configuración de auditoría adecuada implica
  - **Administración del log del sistema**: el daemon/servicio de auditoría nunca debe deshabilitarse y su configuración debe ser inmutable
    - El malware tiende a deshabilitarlo / configurarlo incorrectamente
    - Debe recopilar información de carga / descarga de módulos del kernel
    - Y sobre procesos que arranquen antes del demonio de auditoría
  - **Administración del tamaño del log**: Tienen un tamaño máximo
    - Y deben conservarse durante un período requerido
    - Si sus archivos agotan espacio, se necesitan acciones/notificaciones
      - Enviar correos electrónicos de alerta
      - Rotaciones de logs (reiniciar el log después de alcanzar un tamaño, almacenar los datos anteriores en otro archivo)
      - O enviar logs a sistemas de almacenamiento externos periódicamente



Logs rotados (renombrados y comprimidos por si son necesarios en el futuro)

➢ Ejercicio **L5B4_AUDIT**

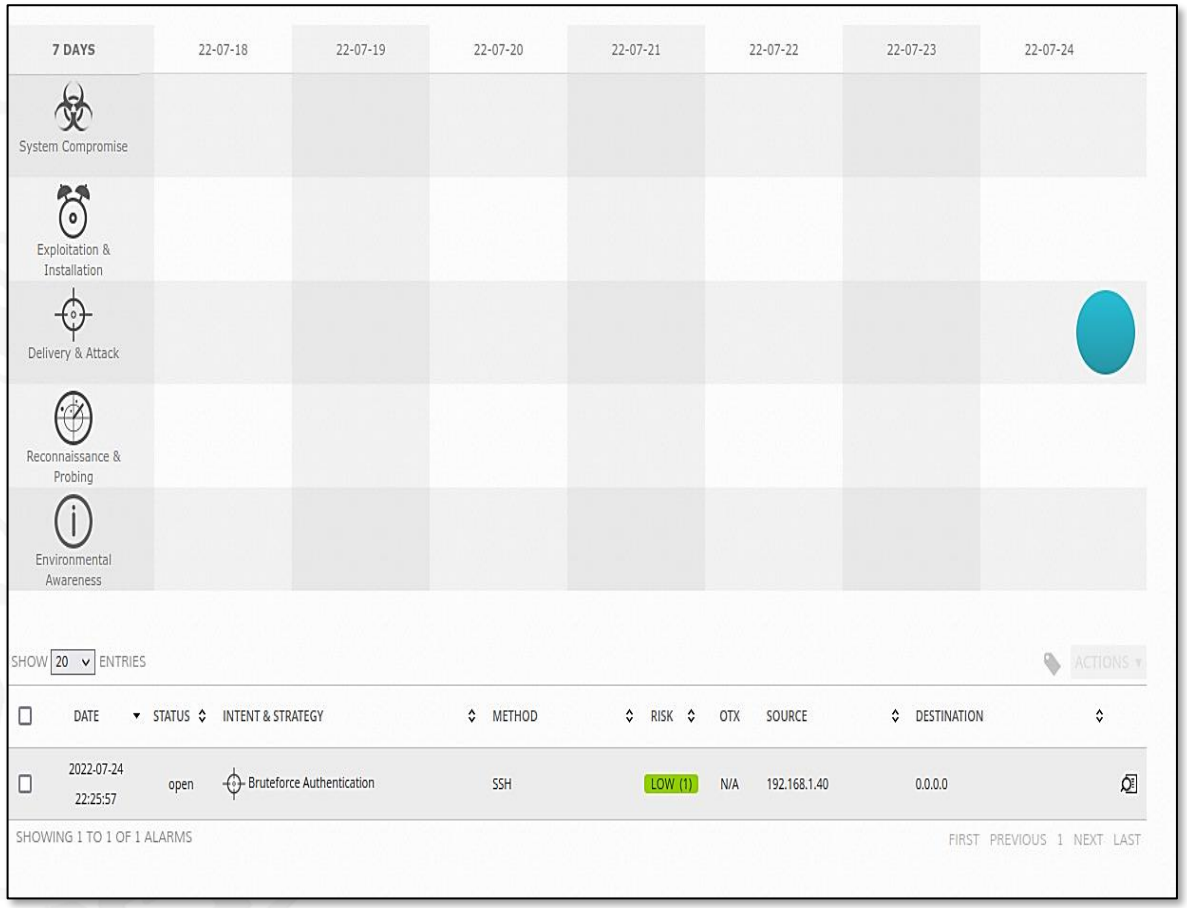
Implementar una configuración de auditoria adecuada

- El comportamiento del usuario debe ser auditado para crear un rastro de cada acción problemática
  - **Usuarios estándar**: **inicio y cierre de sesión**, intentos de **alterar la hora**, cambiar **permisos**, **eliminar** archivos, **intentos de acceso no autorizado** a archivos y **exportar archivos** a medios externos
    - Todos los comandos capaces de realizar estas operaciones deben ser auditados
    - Intentar cambiar los archivos de log indican **eliminación de evidencias de intrusión** (**ataque en curso**)
  - **Administradores**: cada acción de administración o uso de comandos que requieran **privilegios altos**, modificaciones del **entorno de red**, el **inicio de sesión** y la **información de usuario / grupo**, y los **controles de acceso obligatorios (MAC) del sistema**
    - La modificación de partes sensibles del SO es parte del trabajo de un administrador
    - ¡Pero también pueden indicar un compromiso del sistema si se detectan cambios no deseados!
  - En Linux, incluye el uso de **contabilidad de procesos** (`pacct`)
- Las ubicaciones de los logs en cada SO se explicarán en ASR
  - En IFA te enseñan cómo analizar e interpretar el contenido del log
  - También se pueden usar técnicas de ML / IA (SI) para analizar su contenido e identificar amenazas

➢ Ejercicio **L5B2_PACCT**

IMPLEMENTAR UNA CONFIGURACIÓN DE AUDITORIA ADECUADA

- **Los archivos de log generalmente contienen**
  - Fecha, hora y usuario que realizó una acción
  - Detalles sobre cada acción importante
- **Hay muchos archivos de log, pertenecientes a los principales servicios instalados**
  - https://thehackerway.com/2021/06/21/15-ficheros-de-log-interesantes-en-sistemas-linux/
- **Los Host-based Intrusion Detection Systems (HIDS) los usan para detectar amenazas**
  - Como OSSEC (ver imagen)
    - https://computingforgeeks.com/how-to-install-ossec-hids-on-ubuntu-18-04-16-04-debian-9/
  - Pueden tener otras funcionalidades
  - Como la supervisión de políticas, comprobación de la integridad de los archivos, alertas en tiempo real, detección de rootkits, respuesta activa...



OSSEC mandando información a un SIEM OSSIM (https://cybersecurity.att.com/products/ossim). OSSEC también es parte de otros productos, como Wazuh (visto en prácticas)

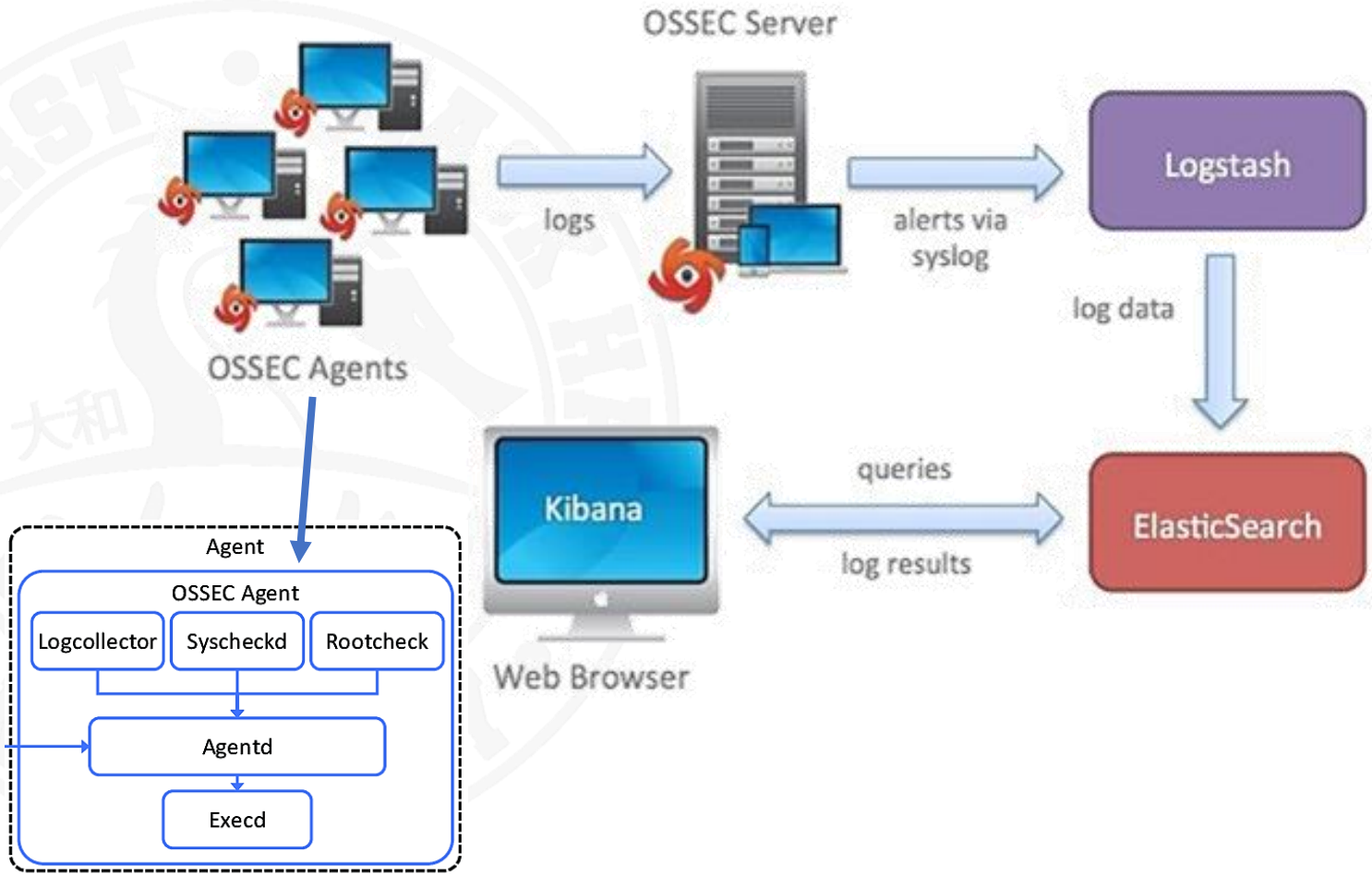
ALMACENAMIENTO REMOTO DE LOGS DEL SISTEMA

- **La mayoría de los equipos en infraestructuras seguras no mantienen logs locales**
  - Los envían a **servidores de log centrales** para almacenarlos y analizarlos
  - **Esto protege la integridad del log de ataques locales**: si un atacante obtiene acceso `root` en el sistema local, podría alterar o eliminar sus datos de log
  - Permite tener fácilmente un **backup** de los logs de todas las máquinas de la infraestructura
  - Además, tener una máquina con hardware especializado en mantener y analizar textos grandes
- **Y, por supuesto, debemos proteger los servidores centrales de log a toda costa**
  - ¡Su información es vital!
  - LAN de administración, aislada del resto de la infraestructura del servidor y de las redes (Tema 5)
  - Correctamente identificado y configurado para recibir solo log de sistemas autorizados (y no enviar logs a falsos sistemas de log remotos centralizados)
- **El servicio `rsyslog` normalmente implementa esto en Linux**
- **Para técnicas forenses de análisis de log, debes cursar IFA**

AUDITAR LOS EQUIPOS DE UNA INFRAESTRUCTURA

- **La instalación de agentes de un HIDS en cada máquina permite una auditoría completa de una infraestructura**
  - En base al contenido de los logs de cada sistema, detección de cambios no autorizados en ficheros, comportamientos sospechosos...
- **Toda esa información se agrega**
  - En un servidor que centraliza los logs enviados con `rsyslogd` (Logstash)
- **Y se puede visualizar y extraer información globalmente**
  - Gracias a herramientas como Kibana y ElasticSearch (**ELK stack**) (**AS**)

Fuente: https://www.researchgate.net/figure/Integration-architecture-of-OSSEC-12_fig1_320649753



Fuente: https://www.researchgate.net/figure/Open-Source-Host-based-Intrusion-Detection-System-OSSEC-components_fig1_335806029

¿Crees que lo has entendido todo? ¡Autoevalúate!

?

- **Eres capaz de hacer lo siguiente**
  - *Entender que durante el proceso de arranque hay opciones de pasar a un modo de mantenimiento o emergencia que te dan acceso pleno al sistema*
    - *Y que solo podemos evitarlos poniendo una contraseña en el lugar adecuado*
  - *Comprender las ventajas de dividir el sistema de ficheros en varias particiones o discos*
    - *Y cómo esta división permite sacar mayor partido de las opciones de montaje de particiones en **Linux**, por ejemplo*
  - *Tener claro que los ficheros temporales podrían tener datos privados*
    - *O usarse de forma maliciosa para causar que un proceso no pueda seguir dando servicio*
    - *Y que en ese sentido los ficheros de swap pueden ser otra fuente de fugas de datos privados*
  - *Entender que la mejor forma de no soportar una funcionalidad potencialmente insegura es directamente no instalar soporte para dichas funcionalidades a nivel de **kernel***
  - ***Sobre los logs***
    - *Entender que todo lo relevante que pasa en el sistema y con los usuarios, erróneo o no, debe quedar reflejado en los logs*
    - *Tener claro que el volumen de información generado es tal que sin una herramienta de análisis (como un HIDS) no podríamos ver las amenazas a un sistema*
    - *Y que, debido a la tremenda importancia de los logs, es muy frecuente almacenarlos externamente*

< Ir al índice

HARDENING DE SERVICIOS / SOFTWARE

Seguridad aplicada a programas

*Iconos por Bing Chat AI

Fuente: Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

¿QUÉ TE VAS A ENCONTRAR EN ESTE BLOQUE?

En este bloque aprenderás

- Que el software se debe **actualizar y minimizar** en todo momento
- Las operaciones más importantes a la hora de **gestionar el software**
- Diferentes modos de control de acceso, concretamente **MAC** y **DAC**
- La importancia del **control, vigilancia y escaneo** del software ejecutado

Mantenimiento y actualización de software

- Mantener el software es muy importante para tener un sistema seguro
  - Parchear el software tan pronto como haya nuevas versiones, para evitar que los atacantes utilicen vulnerabilidades conocidas para infiltrarse en un sistema
  - Los cambios en cualquier componente pueden tener efectos significativos en la seguridad del SO
  - **Las empresas suelen tener una política de actualizaciones adaptada a su contexto**
- Activar el firmado criptográfico de paquetes (Tema 2) garantiza que el software no haya sido manipulado y sea de un proveedor de confianza (Ej.: `debsums`)
- Actualizar todos los paquetes y dependencias regularmente, especialmente cuando se reportan vulnerabilidades graves en el software que utilizamos
  - **Sistemas basados en Debian**: `sudo apt update`, `sudo apt full-upgrade`
    - Y `sudo snap refresh` en las versiones más recientes de Ubuntu
  - **Sistemas basados en Red Hat**: `sudo yum upgrade` (visto en ASR)
- La mejor protección contra el software vulnerable es ejecutar menos software
  - Cómo iniciar y detener los servicios del SO está cubierto en ASR

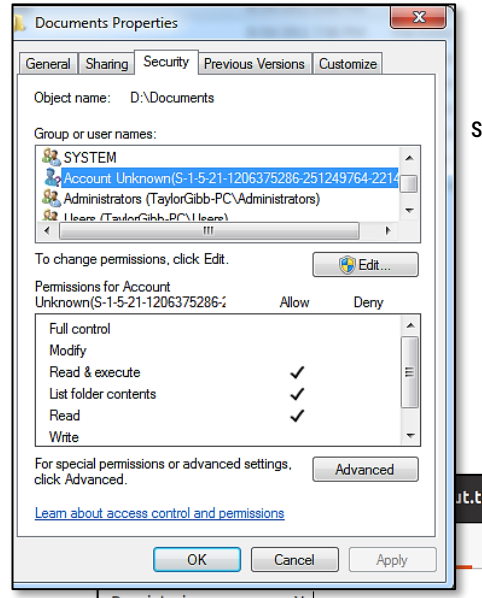
Gestión de software importante

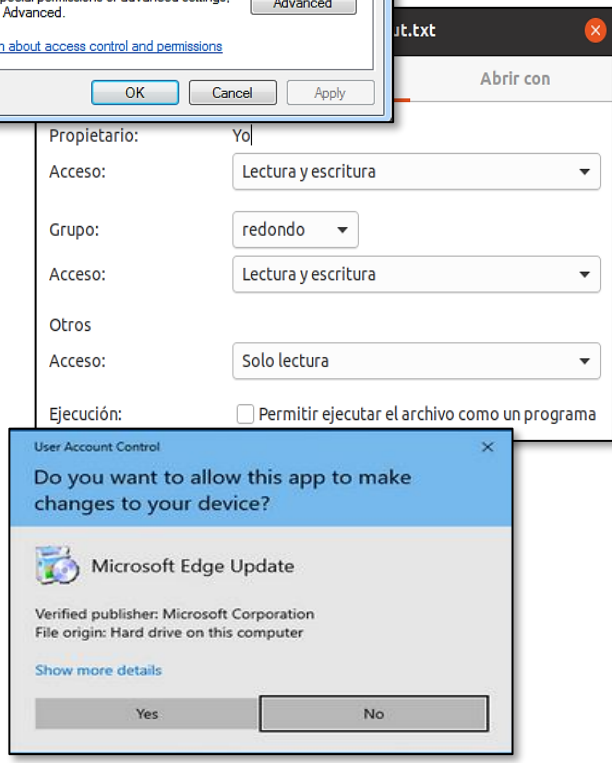
- **Habilitar la programación de tareas**
  - Es necesario para tareas de mantenimiento y de soporte de seguridad
  - Normalmente está habilitado por defecto, tanto en sistemas Windows como Linux (`crond` daemon)
- **Habilitar servicios de sincronización de hora (NTP) (consulta ASR)**
  - La sincronización del tiempo es esencial para servicios de autenticación como Kerberos
  - Es importante para **mantener logs precisos** y auditar adecuadamente las violaciones de seguridad
  - La fecha/hora por defecto se controla automáticamente: **intenta no cambiarlas manualmente**
- **Controlar el consumo de recursos de los procesos con herramientas adecuadas**
  - Ej.: `sysstat`, `glances` en Linux
- **Deshabilitar compiladores**
  - Los compiladores pueden ser usados por atacantes para compilar malware en un servidor
  - Desactivarlos también ayuda a protegerse de ataques que explotan sus vulnerabilidades
    - También sigue la regla de **no instalar cosas que no son realmente necesarias**
    - Un atacante puede encontrar la forma de ejecutar igualmente su código malicioso

➢ Ejercicios **L5B4_SYSSTAT**, **L5B4_GLANCES**, **L5B2_COMPILERS**

TIPOS DE CONTROL DE ACCESO

- **DAC (Control de acceso discrecional, "típicos" permisos)**
  - Restringir el acceso con identidades de usuario o sus grupos
  - El permiso de acceso se puede transferir a otras entidades
  - Ej.: los archivos de Linux tienen permisos de lectura, escritura y ejecución
    - Con tres bits para usuario, grupo y "otros"
- **MAC (Control de acceso obligatorio)**
  - Una entidad (por ejemplo, el SO) **restringe** la capacidad de un sujeto para **acceder o realizar algún tipo de operación** en un objeto u objetivo
  - Ej.: Windows Mandatory Integrity Control (MIC) controla el acceso a objetos y funciona antes de evaluar los permisos de archivo (DAC)
    - Combina 4 niveles de integridad (**IL**) (**bajo**, **medio**, **alto** y **sistema**) con una política obligatoria para evaluar el acceso
    - Los usuarios estándar reciben un **medio**, y los privilegiados **alto**
    - Procesos iniciados por el usuario reciben IL "**bajo**" si su ejecutable es IL "**bajo**"
    - Los servicios del sistema reciben el nivel de integridad del "**sistema**"
  - Un proceso **no puede interactuar con otro que tiene un IL más alto**

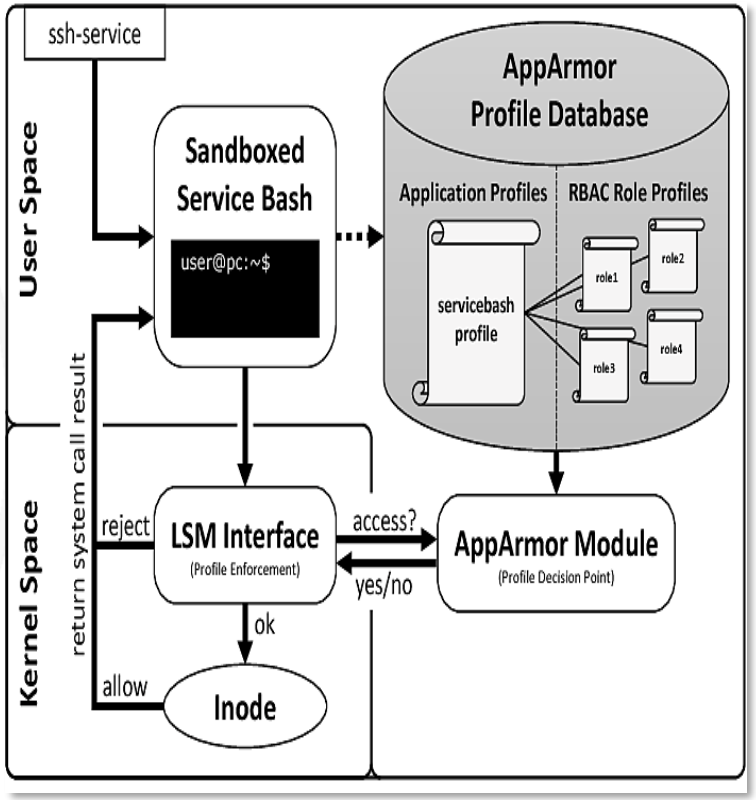




Fuente: https://helpdesk.athens.edu/hc/en-us/articles/4404584871575-Microsoft-Edge

Mandatory Access Controls (MACs)

- **El equivalente Linux al MIC de Windows es una característica del kernel**
  - Protege contra programas mal configurados o comprometidos
  - Sistemas RedHat usan **SELinux** y los basados en Debian usan **AppArmor**
    - Ambos limitan a qué archivos pueden acceder los programas...
    - ¡Y qué acciones pueden hacer!
- **Para que funcione bien hay que tomar una serie de medidas**
  - Asegurarse de que esté habilitado y "**enforcing**"
    - Tiene políticas de seguridad diseñadas para evitar que los procesos causen daños al sistema o eleven sus privilegios
  - Asegurarse de que todos los procesos del sistema estén **cubiertos** por él
- **Los bits de permiso del sistema operativo y los atributos de archivos y procesos (Windows / Linux) se describen en ASR**
  - Además de su asociación con los usuarios



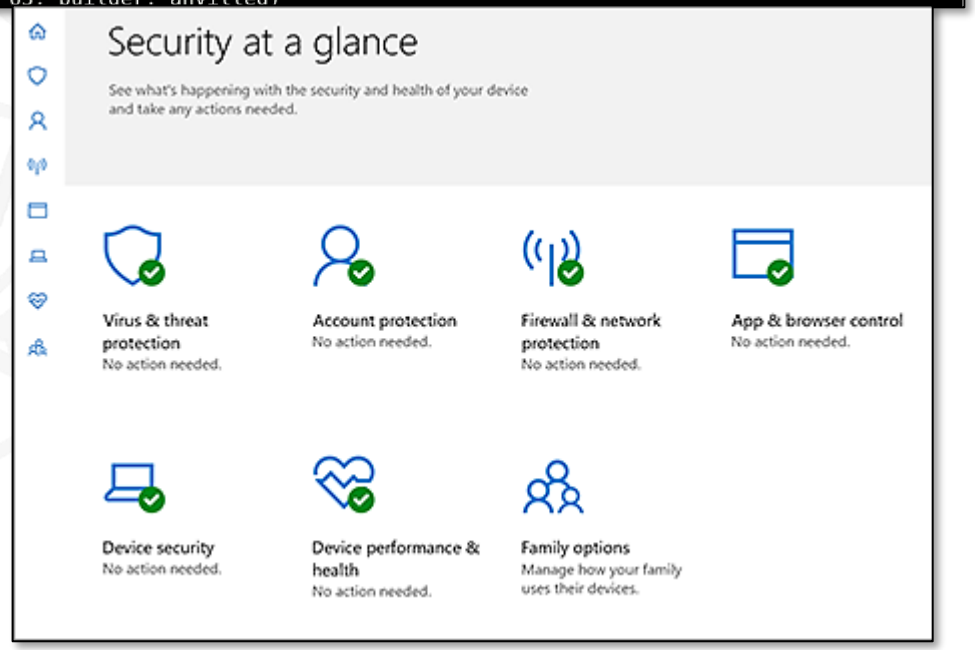
Fuente:
https://www.semanticscholar.org/paper/Using-RBAC-to-Enforce-the-Principle-of-Least-in-Kern-Anderl/c3438119a1d466f69568b48d6ae3f1664d1e3830

Programas: Malware

Seguridad de Sistemas Informáticos

- Deben comprobarse periódicamente los programas instalados para buscar **malware**
- En **Linux**, puedes hacerlo con
  - Linux Malware Detect: https://www.globo.tech/learning-center/install-use-lmd-clamav-ubuntu-16/
  - ClamAV: https://kifarunix.com/install-and-use-clamav-on-ubuntu-20-04/
    - ClamAV generalmente funciona en segundo plano
    - Necesitamos verificar su registro para verificar las infecciones detectadas
- Los sistemas **Windows** tienen **Windows Defender** cumpliendo esta función
- En caso de duda, podemos realizar un análisis exhaustivo con **VirusTotal**
  - https://www.virustotal.com/es/

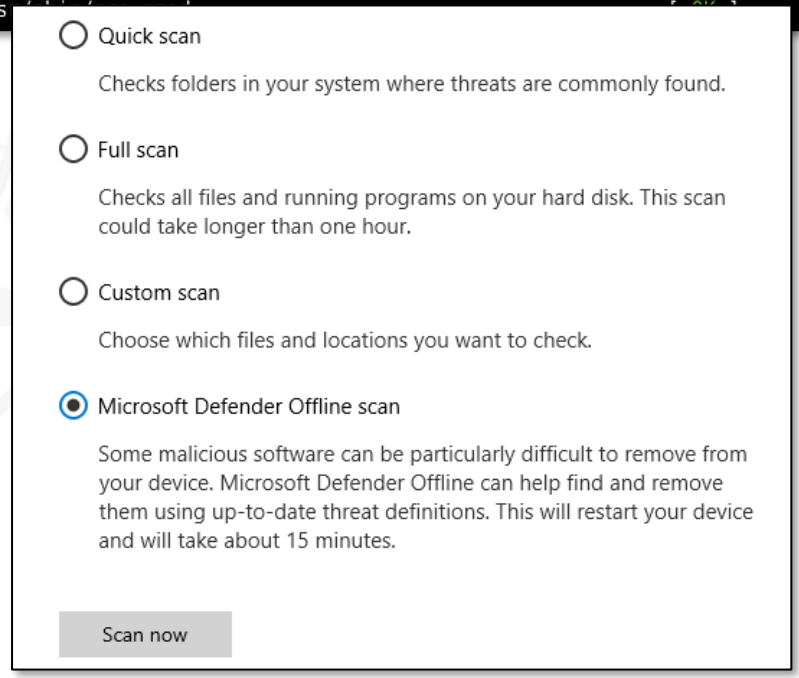
```
redondo@miw:~$ cat /var/log/clamav/freshclam.log
Thu Jul 16 16:33:51 2020 -> ------------------------------------
Thu Jul 16 16:33:51 2020 -> freshclam daemon 0.102.3 (OS: linux-gnu, ARCH: x86_6
4, CPU: x86_64)
Thu Jul 16 16:33:51 2020 -> ClamAV update process started at Thu Jul 16 16:33:51
 2020
Thu Jul 16 16:33:51 2020 -> daily database available for download (remote versio
n: 25875)
Thu Jul 16 16:33:57 2020 -> Testing database: '/var/lib/clamav/tmp.f35da/clamav-
d61d951eebdd284e3c30c2caa8440014.tmp-daily.cvd' ...
Thu Jul 16 16:34:07 2020 -> Database test passed.
Thu Jul 16 16:34:07 2020 -> daily.cvd updated (version: 25874, sigs: 3410927, f-
level: 63, builder: raynman)
Thu Jul 16 16:34:07 2020 -> main database available for download (remote version
: 59)
Thu Jul 16 16:34:12 2020 -> Testing database: '/var/lib/clamav/tmp.f35da/clamav-
38f2c15e309879a91acf2ccca95f72bf.tmp-main.cvd' ...
Thu Jul 16 16:34:17 2020 -> Database test passed.
Thu Jul 16 16:34:17 2020 -> main.cvd updated (version: 59, sigs: 4564902, f-leve
l: 60, builder: sigmgr)
Thu Jul 16 16:34:17 2020 -> bytecode database available for download (remote ver
sion: 331)
Thu Jul 16 16:34:17 2020 -> Testing database: '/var/lib/clamav/tmp.f35da/clamav-
0f7c356ca71f5dee810bc8384b4a4c36.tmp-bytecode.cvd' ...
Thu Jul 16 16:34:18 2020 -> Database test passed.
Thu Jul 16 16:34:18 2020 -> bytecode.cvd updated (version: 331, sigs: 94, f-leve
l: 63, builder: anvilleg)
```



PROGRAMAS: ROOTKITS

- Los **rootkits** necesitan herramientas y atención especial
  - En Windows, Defender y System Guard protegen de ellos
  - En Linux, podemos comprobar el sistema contra rootkits con software como `chkrootkit` o `rkhunter`
  - También pueden detectar exploits locales y backdoors
    - https://sourceforge.net/projects/rkhunter
    - https://computernewage.com/2015/01/10/detecta-rootkits-y-rastros-de-malware-en-linux-con-chkrootkit-y-rkhunter/
    - https://www.tecmint.com/install-rootkit-hunter-scan-for-rootkits-backdoors-in-linux/
- Podemos comprobar la máquina local o montar discos en otra para escanearla externamente
  - Esto es aconsejable, ya que los rootkits generalmente se esconden cuando su sistema se está ejecutando.
  - Microsoft Defender Offline se creó por este motivo
    - https://support.microsoft.com/en-us/help/17466/windows-microsoft-defender-offline-help-protect-my-pc

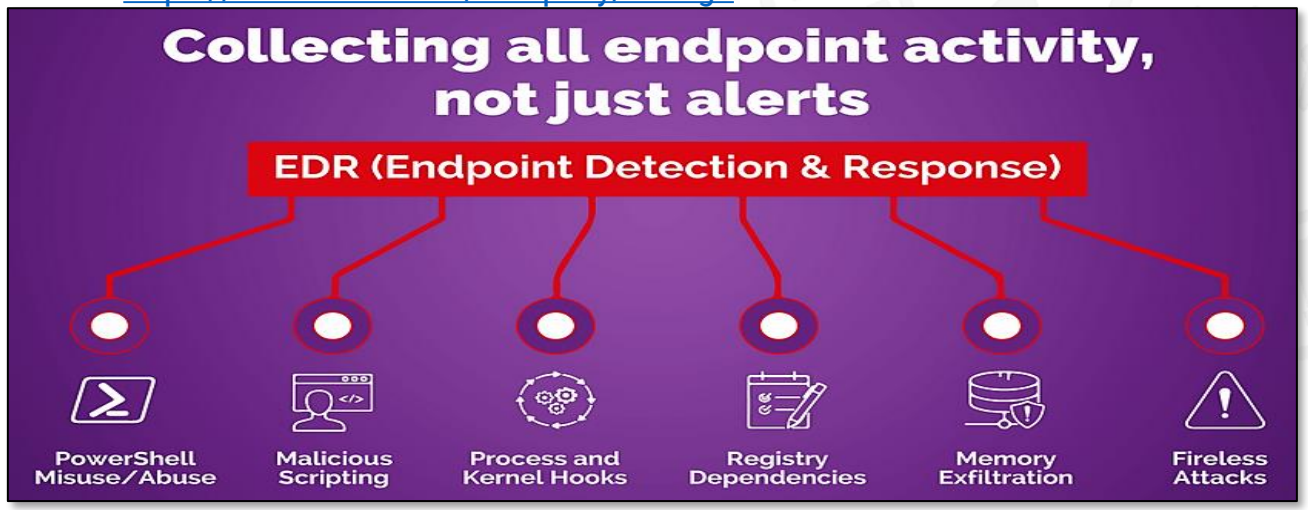
```
redondo@miw:~$ sudo rkhunter --check
[ Rootkit Hunter version 1.4.6 ]

Checking system commands...

  Performing 'strings' command checks
    Checking 'strings' command                              [ OK ]

  Performing 'shared libraries' checks
    Checking for preloading variables                       [ None found ]
    Checking for preloaded libraries                        [ None found ]
    Checking LD_LIBRARY_PATH variable                       [ Not found ]

  Performing file properties checks
    Checking for prerequisites                              [ OK ]
    /usr/sbin/adduser                                       [ OK ]
    /usr/sbin/chroot                                        [ OK ]
    /usr/sbin/cron                                          [ OK ]
    /usr/sbin/groupadd                                      [ OK ]
    /usr/sbin/groupdel                                      [ OK ]
    /us
```



Tecnologías Endpoint Detection and Response (EDR o XDR)

- **Monitoriza dispositivos y responde de forma continuada para mitigar amenazas**
- **Identifica comportamientos sospechosos y Advanced Persistent Threats (APT) en puntos finales (dispositivos informáticos) en un entorno**
  - Recopila y agrega datos de equipos y otras fuentes, y puede dirigirlos a un SIEM (**tema 5**)
  - Su funcionalidad principal es **alertar a administradores de anomalías**, más que proteger dispositivos
    - Sin embargo, muchos tienen respuesta en tiempo real a las amenazas, detectan inyecciones de malware, crean listas de elementos permitidos y bloqueados, y otras capacidades de respuesta avanzadas
  - Algunos utilizan la clasificación del framework MITRE ATT&CK para las amenazas (**temas 5 y 7**)

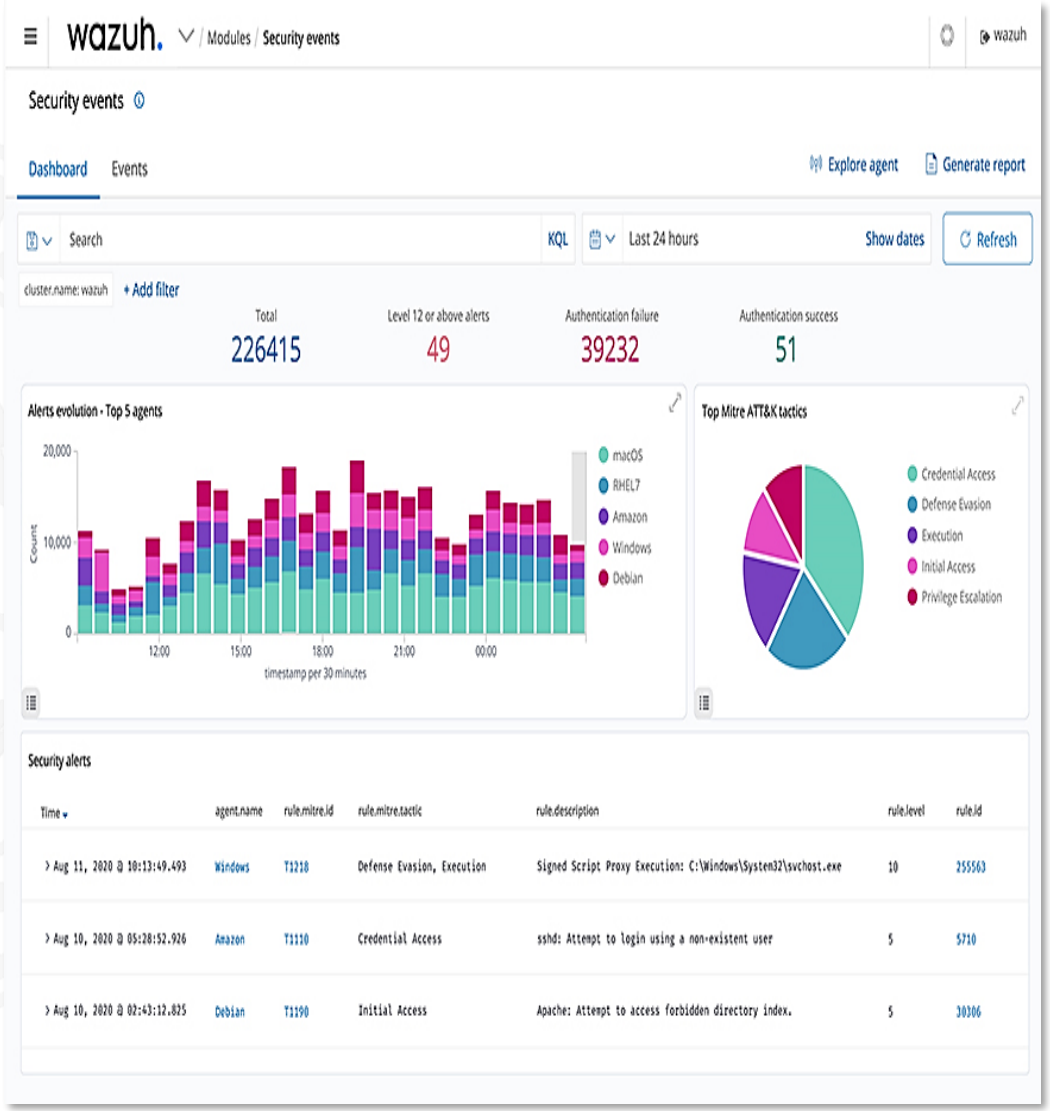
Fuente: https://ar.linkedin.com/company/absega



Fuente: https://www.dnsstuff.com/best-edr-solutions

EDR/XDR vs Antimalware

- **Un antimalware es más simple y de ámbito más limitado que un EDR**
  - De hecho, muchos EDR incorporan un antimalware
- **Un EDR/XDR puede tener más funcionalidades**
  - Escaneo, detección y eliminación de malware
  - Firewall
  - Herramientas de monitorización variadas para muchas partes del sistema (Ej.: **HIDS**, mira la imagen anterior)
  - Protección contra intentos de explotar vulnerabilidades del sistema y robo de información
- **Proporcionan una protección más completa**
  - Siguen una arquitectura cliente-servidor
  - Wazuh (https://wazuh.com/) es una de las soluciones XDR/SIEM más usadas, open source y de uso gratuito



¿Crees que lo has entendido todo? ¡Autoevalúate!

?

- **Eres capaz de hacer lo siguiente**
  - *Entender que sin un control de lo que instalas y sin un mantenimiento (actualizaciones) adecuado, un sistema va a ser mucho más inseguro*
  - *Comprender la importancia de programar tareas de seguridad, en lugar de confiar en que nos acordaremos de ellas*
  - *Comprender la importancia de sincronizar adecuadamente la hora, ya que es imprescindible para una gran cantidad de servicios*
  - *Entender por qué deshabilitar compiladores que no usemos es una medida más de seguridad*
  - *Tener clara la diferencia fundamental entre DAC y MAC, y quien es el que decide a lo que se accede o no en cada caso*
  - *Comprender la importa de tener herramientas de escaneo de malware adecuadas a cada tipo de malware existente*
  - *Entender la enorme importancia a la hora de vigilar comportamientos inadecuados en un sistema de un EDR/XDR*

< Ir al índice

Fuente: Bing Chat AI

Passwords >

USUARIOS

Asegurando la gestión de usuarios

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

*Iconos por Bing Chat AI

¿QUÉ TE VAS A ENCONTRAR EN ESTE BLOQUE?

- **En este bloque aprenderás**
  - Los puntos principales a la hora de gestionar de forma segura a los **usuarios** de un sistema
  - La enorme diferencia entre entrar a un sistema como **usuario estándar o de privilegios elevados**
  - Las medidas adecuadas para crear **contraseñas seguras y proteger cuentas**

Reglas generales

- **En general, a menos usuarios que puedan iniciar sesión, menos opciones de intrusión**
  - La creación de usuarios está cubierta en ASR
  - Por tanto, hay que minimizar el nº de usuarios que puedan acceder de forma remota o local
  - Incluso con **máquinas virtuales o contenedores "desechables"**, sin usuarios con inicio de sesión
    - Ejecutan un proceso al inicio y se regeneran por completo si algo falla
- **Si los usuarios de inicio de sesión son imprescindibles...**
  - Debemos disminuir el nº de usuarios con privilegios de administrador / sudoers al mínimo (o 0)
  - No queremos muchos usuarios con capacidad de hacerse pasar por `root` a través de `sudo`
- **En Linux, si debe haber sudoers, podríamos**
  - Dejar un solo usuario sudoer para fines de administración
  - Limitar el número de comandos a los que los sudoers pueden hacer `sudo`, incluso solo uno

CUENTAS Y ACCESO

- **Si un atacante obtiene acceso interactivo a una cuenta, puede realizar cualquier acción o acceder a cualquier archivo al que tenga acceso esa cuenta**
  - Por tanto, un sistema seguro debe evitar el acceso a usuarios no autorizados tanto como sea posible
  - Especialmente para cuentas privilegiadas
- **Incluye habilitar Multi-Factor Authentication (MFA) en servicios de acceso**
  - Como SSH: https://medium.com/mi-rincon/doble-factor-de-autenticacion-2fa-c2c8d728b978
  - Hablaremos más sobre MFA sobre el **Tema 6**
- **También mejorar la configuración del acceso para que sea lo más segura posible**
  - Por ejemplo, dejar las **opciones de configuración más seguras posibles** en el servicio SSH
- **Recuerda: La forma más fácil de obtener acceso no autorizado a un sistema Linux es arrancar el servidor en single user mode**
  - Los atacantes pueden seleccionar un kernel para arrancar desde un elemento de menú GRUB

Ficheros de configuración de la sesión

- **Cuando un usuario inicia sesión en una cuenta, el sistema configura su sesión leyendo varios archivos**
  - Muchos de ellos están en el directorio principal del usuario y pueden tener permisos débiles debido a errores de usuario o una configuración incorrecta
- **En sistemas Linux, un valor de máscara mal configurado podría dar lugar a archivos con permisos excesivos accesibles por usuarios no autorizados**
  - `umask 027` es mejor desde el punto de vista de la seguridad
  - `umask 077` es aún mejor para `root` (solo el propietario puede leer o ejecutar nuevos archivos)
  - Evita algunos errores comunes de administrador de sistemas
  - Es más difícil para un atacante escalar privilegios
  - La desventaja es cuando se crean archivos / directorios en un directorio compartido al que acceden otros usuarios
  - En Linux, esto se consigue incluyendo en `/etc/profile` y en `/etc/bash.bashrc`: `umask 027`

MENSAJES DE LOGIN

- **Los banners de inicio de sesión advierten a cualquier posible intruso que quiera acceder al sistema que ciertos tipos de actividad son ilegales**
  - También informa a los usuarios autorizados y legítimos de sus **obligaciones** (uso aceptable del entorno o entornos informatizados o en red)
  - Estos mensajes de advertencia antes del inicio de sesión pueden **disuadir** el uso no autorizado, aumentar la concienciación de seguridad y ser una base legal para denunciar accesos no autorizados
- **Ej.: El DoD (EE.UU) usa esto (los sistemas Linux ponen esto en `/etc/issue`)**

You are accessing a U.S. Government (USG) Information System (IS) that is provided for USG-authorized use only. By using this IS (which includes any device attached to this IS), you consent to the following conditions:

- The USG routinely intercepts and monitors communications on this IS for purposes including, but not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.

- At any time, the USG may inspect and seize data stored on this IS.

- Communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may be disclosed or used for any USG-authorized purpose.

- This IS includes security measures (e.g., authentication and access controls) to protect USG interests -- not for your personal benefit or privacy.

- Notwithstanding the above, using this IS does not constitute consent

Restricción de logins de root / Administrador

- **Normalmente, el administrador debe acceder al sistema a través de una cuenta única sin privilegios**
  - Solo inicia sesión como administrador cuando sea estrictamente necesario (**emergencias**)
  - En sistemas Linux es recomendable utilizar `su` o `sudo` para ejecutar comandos privilegiados desde cuentas sudoer
- **Medidas para reforzar los inicios de sesión de `root` / `Administrador`**
  - **Solo una cuenta debe tener este rol**
  - **Impedir los inicios de sesión directos como `root`**: en Linux es mejor deshabilitarlos directamente (ver imagen)
    - Usar la autenticación multi-factor (MFA) en cuentas privilegiadas
  - **Proteger los inicios de sesión de cuentas de sistema (cuentas de servicio)**: no otorgar privilegios de inicio de sesión interactivos a las cuentas de servicio
    - Dificulta que los atacantes las usen

¡No es bueno que root pueda acceder al sistema interactivamente, también debería ser nologin!

cat /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run
systemd-network:x:101:103:systemd Network Management,,,:/run/sy
n
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/reso
syslog:x:103:107::/home/syslog:/usr/sbin/nologin
messagebus:x:104:108::/nonexistent:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:106:112::/run/uuidd:/usr/sbin/nologin
```

Gestión de passwords de usuario

No dejes que passwords pobres te arruinen el día

*Iconos por Bing Chat AI

Fuente: Bing Chat AI

POLÍTICAS DE PASSWORDS

- **A pesar de sus problemas, usar nombre de usuario / contraseña es muy común**
  - En sistemas Linux, esta característica la soportan los archivos `/etc/passwd` y `/etc/shadow` (ASR)
  - Las contraseñas nunca se almacenan en texto plano (recuerda los KDF del **Tema 2**)
- **La política de contraseñas de una determinada empresa se crea respondiendo a**
  - *¿Qué caracteres están permitidos y prohibidos dentro de una contraseña?*
  - *¿Se requiere que el usuario use **caracteres de diferentes juegos** (letras minúsculas y mayúsculas, dígitos y símbolos especiales)?*
  - *¿Con qué frecuencia puede un usuario cambiar su contraseña?*
  - *¿Con qué frecuencia puede un usuario reutilizar una contraseña? ¿Mantenemos un **historial de sus contraseñas anteriores** para que no pueda usar las mismas otra vez?*
  - *¿Cómo de rápido puede un usuario cambiar su contraseña después de un cambio anterior?*
    - Los usuarios pueden saltarse su historial de contraseñas anteriores cambiándola varias veces hasta que puedan volver a poner la contraseña inicial
  - *¿Cuándo debe un usuario cambiarla? ¿Tras X días? ¿Tras un bloqueo por intentos excesivos de login?*
  - *¿Cómo de diferente debe ser una contraseña nueva de la última?*
  - *¿Se impide al usuario usar su nombre de usuario u otra información de su cuenta en la contraseña?*

Medidas comunes de protección de passwords

- **Establecer una política de caducidad de password**
  - Requiere establecer una advertencia de contraseña, edad mínima y máxima
  - **La advertencia de antigüedad de passwords** permite a los usuarios cambiarla antes de que caduque
  - **La duración mínima de passwords** ayuda a evitar cambios repetidos que inhabiliten las medidas que previenen la reutilización de contraseñas
  - **La antigüedad máxima de la passwords** garantiza que los usuarios deban cambiar periódicamente sus contraseñas (**actualmente no se recomienda usar una**, solo cambiarla cuando se ha filtrado)
- **Deshabilitar cuentas inactivas**
  - Garantiza que los atacantes no puedan usar cuentas sin actividad si han comprometido su password
- **Implementar una política adecuada de seguridad y administración de contraseñas**
  - En los sistemas Windows, esto se logra mediante políticas del SO
    - Si estamos en un controlador de dominio, podemos establecer una para todas las máquinas de un dominio
  - En los sistemas Linux, esto se logra normalmente configurando un módulo llamado PAM

POLÍTICA DE FORTALEZA DE PASSWORDS

- **Algoritmo de hash de contraseñas**
  - Las distribuciones modernas de Linux y Windows llevan por defecto algoritmos seguros (ver **Tema 2**)
- **Intentos fallidos de contraseña**
  - Capacidad para bloquear cuentas de usuario si han introducido mal la contraseña varias veces
  - El bloqueo de cuentas de usuario puede facilitar los ataques DoS, pero evita la averiguación de una contraseña por fuerza bruta
- **Limitar la reutilización de contraseñas**
  - Establecer la frecuencia con la que se puede reutilizar una contraseña antigua
  - No permitir que los usuarios reutilicen contraseñas recientes que podrían haberse visto comprometidas
  - Por ejemplo, el Departamento de Defensa de los Estados Unidos lo fija en 5 contraseñas
    - Una contraseña no puede volverse a ponerse hasta que hayamos usado 5 contraseñas distintas posteriormente a haber usado esta

Ejemplo de política de fortaleza de passwords: Requisitos de calidad de las password

- El uso de una contraseña compleja / segura aumenta el tiempo y los recursos necesarios para comprometerla
- Los SO generalmente nos permiten obligar a los usuarios a usarlas controlando varias variables
  - **Caracteres mínimos requeridos en una contraseña**
    - Cuanto más corta sea la contraseña, menor será el número de combinaciones posibles que deben probarse antes de que la contraseña se vea comprometida (más débil)
  - **Uso de dígitos, minúsculas y mayúsculas** en una contraseña
  - **Número mínimo de caracteres que deben ser diferentes de la contraseña anterior**
  - **Número máximo de caracteres no alfanuméricos** en una contraseña
  - **Rechazar contraseñas que contengan más de N caracteres iguales consecutivos**
- También deben ser contraseñas que no sean comunes o ya filtradas
  - Aunque cumplan con todas las condiciones para ser fuertes

Ejemplo de implementación de política de passwords: Módulo PAM

- **Biblioteca que permite a los administradores elegir cómo las aplicaciones autentican a los usuarios en sistemas Linux**
  - Y así fortalecer su política de contraseñas
- **Ofrece múltiples esquemas de autenticación de bajo nivel en una API de alto nivel**
  - Sin embargo, las modificaciones del PAM pueden tener consecuencias inesperadas
    - A veces hay que restaurar los valores predeterminados del sistema cuando cambios manuales impiden que las contraseñas funcionen normalmente
  - Herramientas como `authconfig` o `system-config-authentication` permiten hacer eso
  - **Manual de usuario**
    - Using Pluggable Authentication Modules (PAM)
    - The Linux-PAM System Administrators' Guide
  - **Más información**
    - Set a password policy in Red Hat Enterprise Linux 7
    - How to configure password complexity [...] using pam_passwdqc.so
    - Hardening Your System With Tools and Services
    - Configuring Services: PAM

Formación de los usuarios

- **Formar a los usuarios para la prevención de incidentes de seguridad es algo imprescindible, pero también complejo**
  - Necesitamos encontrar materiales adecuados a nuestras necesidades
  - Lo usuarios deben entender la importancia de este proceso de formación y que son parte clave en la seguridad de la empresa
- **Hay muchos recursos gratuitos disponibles para formar a usuarios de distintos niveles de experiencia técnica**
  - **El INCIBE** tiene muchos cursos de formación para distintos perfiles de usuarios
    - Ej.: "**Experiencia Senior**" (usuarios sin conocimientos): https://www.incibe.es/ciudadania/experiencia-senior
  - Como parte de nuestra iniciativa "**Cobra Kali**" tenemos el proyecto "**P-45 Audaz**"
    - **Formación contra ciberfraudes** (gente sin conocimientos técnicos)
      - https://github.com/jose-r-lopez/Fraudes-y-Timos/wiki
      - Podcast y blog donde semanalmente se "disecciona" un tipo de fraude
      - También en YouTube: https://www.youtube.com/channel/UCrB3D_97pYWlfBZBmCJiYzw
    - Curso "**Técnicas de seguridad y prevención del fraude en mensajería electrónica**"
      - https://uniovix.uniovi.es/course/view.php?id=62

¿Crees que lo has entendido todo? ¡Autoevalúate!

?

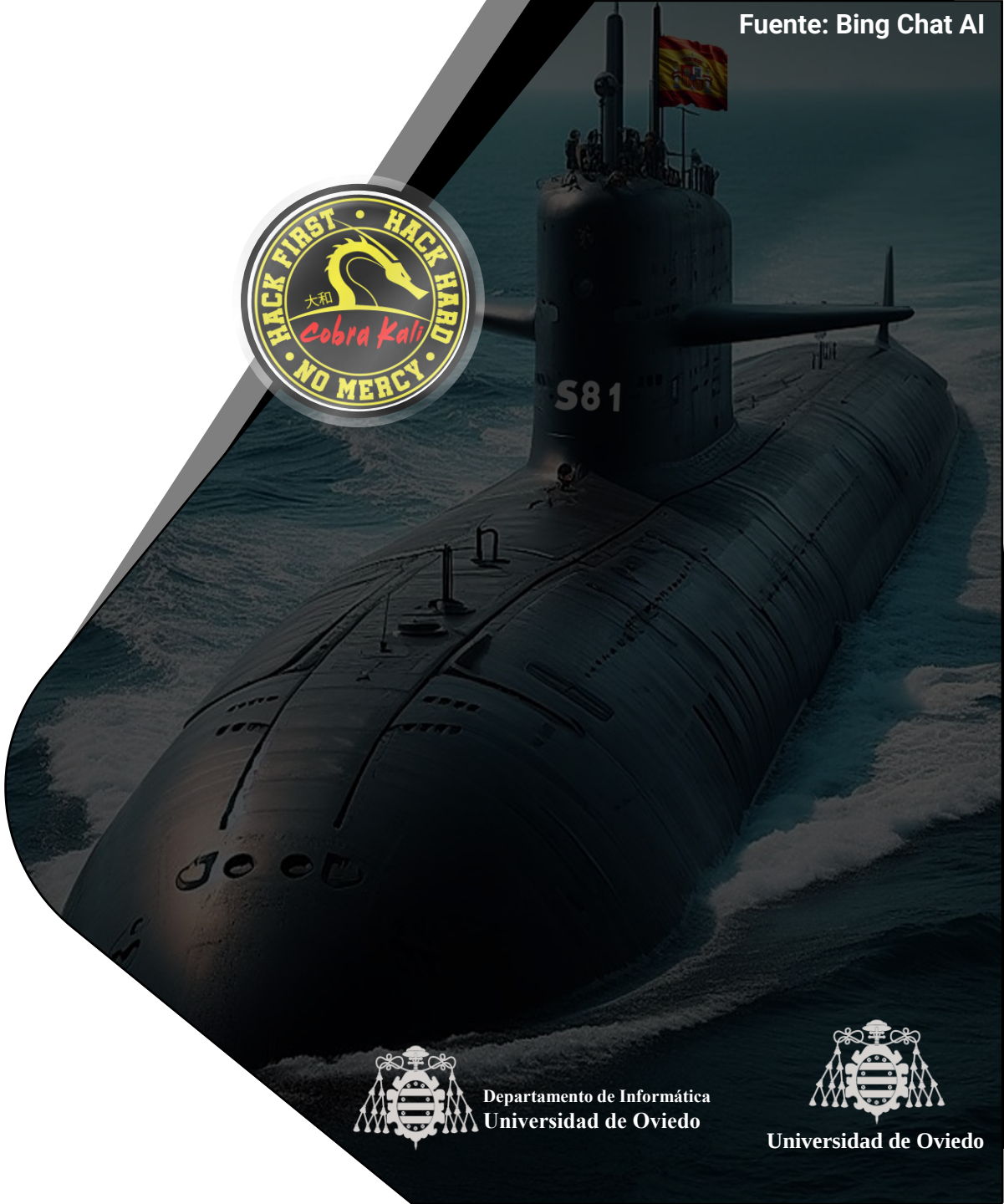
- **Eres capaz de hacer lo siguiente**
  - *Entender que cuantos menos usuarios puedan iniciar sesión en el sistema, mayor seguridad*
    - *Incluyendo minimizar el nº de sudoers en Linux*
  - *Comprender que todo servicio de acceso remoto a un sistema debería tener un MFA*
  - *Comprender la importancia de configurar correctamente los ficheros de configuración de las sesiones de usuario*
  - *Tener claro que los avisos que se colocan en las pantallas de login no tienen un valor real a la hora de mejorar la seguridad*
  - *Entender por qué es tan importante operar como una cuenta de privilegios elevados el menor tiempo posible*
    - *Incluso prohibiendo el acceso interactivo con la misma*
  - *Tener claro las reglas de creación de una buena contraseña*
  - *Tener claras las reglas de protección de cuentas relativas a una buena gestión de contraseñas:*
    - *Caducidad de contraseñas solo en caso de filtración, deshabilitar cuentas inactivas, forzar la creación segura, limitar la reutilización, bloquear ante un nº excesivo de intentos fallidos...*
  - *Entender que, sin una adecuada formación de los usuarios, estas medidas son insuficientes*

< Ir al índice

ARCHIVOS

Asegurando el acceso a tus datos

*Iconos por Bing Chat AI



Fuente: Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

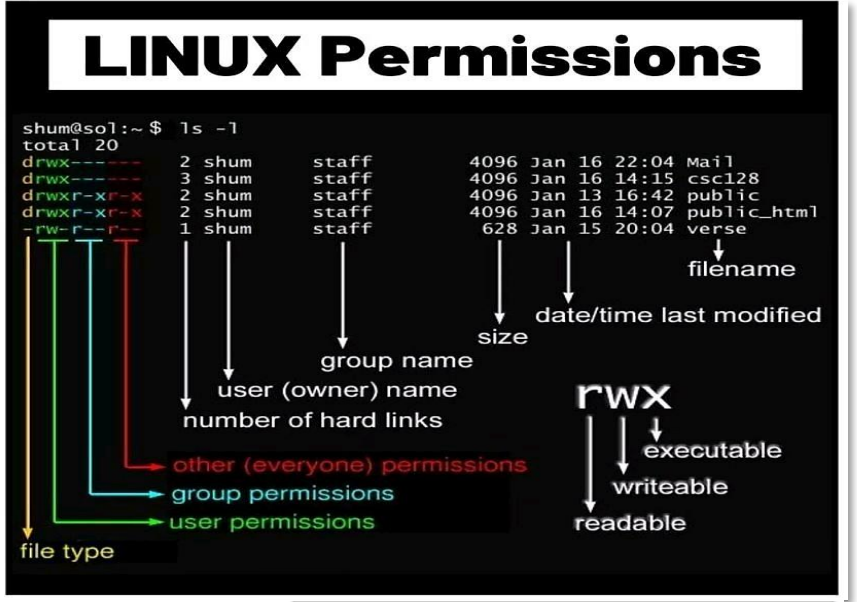
¿QUÉ TE VAS A ENCONTRAR EN ESTE BLOQUE?

En este bloque aprenderás

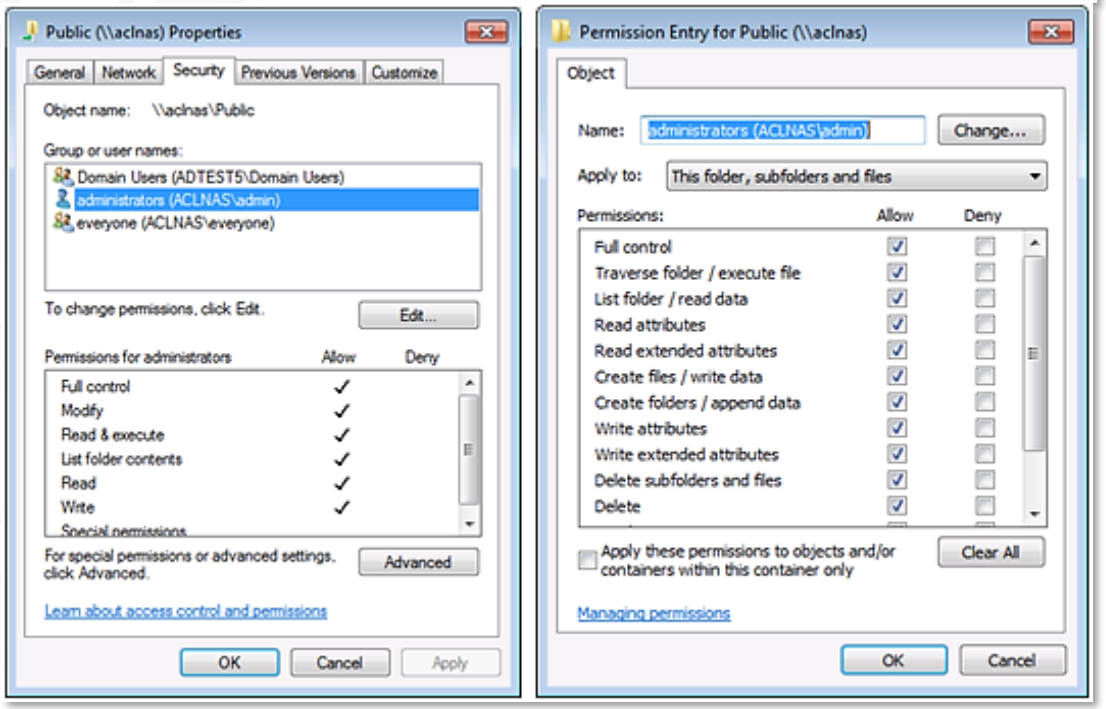
- La importancia de los **permisos** de ficheros y directorios en los distintos SO de cara a mantener una seguridad adecuada
- Las operaciones que debes hacer para **gestionar correctamente un sistema de ficheros**
- Lo importante que es **controlar los cambios** a los ficheros que no deberían modificarse habitualmente en un SO

Permisos y límites de ficheros críticos

- **Para entender esto debes recordar el esquema de permisos usado por tu SO**
  - La primera imagen muestra el típico de **Linux**
- **Windows usa un esquema diferente, basado en usuarios que pertenecen a grupos**
  - Se pueden asignar permisos a ficheros o directorios a varios usuarios o grupos, formando una **Access Control List** (ACL) (ver segunda imagen)
  - Hay muchos permisos definidos permitiendo un acceso más granular que en **Linux**
- **Los permisos se ven en SO y ASR**



Fuente: https://www.devopsschool.com/blog/linux-tutorials-chmod-commands/



Fuente: http://qnapsupport.net/qnap-uygulamalari/kullanici-ve-klasor-olusturma/klasor-olusturma-ve-yetkilendirme/how-to-use-windows-acl-to-manage-user-permissions-on-the-qnap-nas

PERMISOS Y LÍMITES DE FICHEROS CRÍTICOS

- **La seguridad de un SO se basa en permisos de archivos y directorios adecuados**
  - Evitan que usuarios no autorizados lean o modifiquen archivos a los que no deberían tener acceso
  - Los permisos para muchos archivos deben establecerse de la forma más restrictiva posible para garantizar que la información confidencial esté protegida adecuadamente
  - Se deben mantener los permisos restrictivos por defecto para los archivos que actúan como bases de datos de seguridad importantes, y comprobar que no se han cambiado (**indicio de compromiso, IoC**)
    - `passwd`, `shadow`, `group` y `gshadow` (Linux) (carpeta `/etc`)
    - El archivo `SAM` (Windows) en la carpeta `C:\Windows\System32\config`
- **Hay que asegurarse de que todos los archivos son propiedad de un usuario y grupo**: los archivos sin propietario pueden deberse a
  - Un intruso que usa herramientas o malware diverso para filtrar datos
  - Instalación incorrecta del software o fallos en su eliminación
  - No eliminar todos los archivos que pertenecen a una cuenta de usuario que se ha borrado

Permisos y límites de ficheros críticos

- **Sobre ficheros y directorios con permiso de escritura para todos (world-writable)**
  - No deben existir archivos con esos permisos
    - Sus datos los puede modificar cualquier usuario del sistema, y supone un alto riesgo de seguridad
    - Deben combinarse los permisos de usuario / grupo para permitir accesos legítimos
  - En Linux, los directorios world-writable deben tener sus **sticky bits** activos
    - Si no se hace en directorios públicos, usuarios no autorizados podrían eliminar archivos en su estructura
- **Deshabilitar sistemas de archivos no utilizados en Linux**
  - Linux permite agregar y eliminar automáticamente sistemas de archivos en ejecución
  - Habilitarlos conlleva cierto riesgo (¡minimiza siempre el software!)
- **Deshabilitar los core dumps (imágenes de la RAM grandes volcadas en archivos)**
  - Pueden contener datos confidenciales presentes en la RAM de programas cuando se hizo el volcado
    - Contraseñas, claves API, tokens de acceso, ID de sesiones HTTP activas...
    - Existen herramientas para extraer información de ellos: IFA usa Volatility Framework
      - https://cqureacademy.com/blog/forensics/memory-dump-analysis
      - http://manpages.ubuntu.com/manpages/bionic/man8/crash.8.html

➢ Ejercicio **L5B2_COREDUMPS**

Permisos y límites de ficheros críticos: Cambios no deseados

- Después de la instalación del SO base, los archivos del disco se pueden dividir en dos categorías
  - **Archivos que se modifican con frecuencia**: archivos de usuario, archivos de desarrollo, logs...
  - **Archivos que no deben modificarse** (o que se modifican solo en circunstancias específicas)
    - Archivos del SO, controladores de dispositivos... normalmente se modifican con **actualizaciones** del SO
    - Una modificación no planificada de estos archivos puede indicar malware (Ej.: ransomware)
- Un **HIDS (Host-Based Intrusion Detection System)** puede detectar modificaciones en archivos confidenciales
  - Detectar una puede revelar la presencia de malware antes de que sea demasiado tarde
  - Tripwire, Aide y OSSEC (mencionados anteriormente) son tres de los más famosos
  - **Más ejemplos**: https://www.dnsstuff.com/host-based-intrusion-detection-systems
- Los detalles sobre los sistemas de archivos y su contenido se explican en **ASR**

¿Crees que lo has entendido todo? ¡Autoevalúate!

?

- **Eres capaz de hacer lo siguiente**
  - *Comprender las diferencias entre los permisos de **Linux** y los de **Windows***
  - *Entender que hay ficheros que deben tener unos permisos muy restrictivos por la naturaleza de los datos que guardan*
    - *Y que, de no estar así, es un indicio de compromiso del sistema*
  - *Comprender que un permiso de escritura para todos es un riesgo de seguridad*
  - *Entender que los **core dumps** pueden tener información privada y por ello deben deshabilitarse*
  - *Comprender que un HIDS es clave para identificar cambios en ficheros del SO que no deberían cambiar salvo circunstancias muy concretas*

< Ir al índice

HARDENING Y EVALUACIÓN DE SEGURIDAD

Procedimientos de hardening eficientes

*Iconos por Bing Chat AI

Fuente: Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

IMPLEMENTANDO Y EVALUANDO LA SEGURIDAD

- **Fortalecer un SO significa aumentar su nivel de seguridad actual**
  - Este proceso también se conoce como **hardening**
- **Para ello, aplicamos una serie de operaciones de seguridad a partes del SO**
  - En esta presentación revisamos las áreas más importantes
  - Cada una de estas operaciones de aplicación de seguridad se denomina **control de seguridad**
  - Por tanto, reforzar cualquier SO **significa implementar una lista (extensa) de controles de seguridad**
    - ¡Comprobando previamente si no están aplicados ya!
  - Si no tenemos una lista de controles de seguridad disponibles, siempre podemos confiar en las **guías de mejores prácticas** para productos de software concretos como plan de respaldo.
    - Las oficiales españolas se pueden consultar gratuitamente desde la web del CCN-CERT: https://www.ccn-cert.cni.es/informes/informes-de-buenas-practicas-bp/
- **¡Este proceso de hardening se puede aplicar a cualquier producto de software!**
  - El hardening de software significa que primero obtengamos **su mejor lista de controles de seguridad**
  - Y luego implementar cada uno de ellos, si no lo están ya
    - ¡Excepto si eso rompe con su funcionalidad legítima!

IMPLEMENTANDO Y EVALUANDO LA SEGURIDAD

- **Las listas de controles de seguridad que usemos no pueden ser cualquiera**
  - **Validadas** por terceros de confianza, **completas**: (deben cubrir todas las partes "sensibles" del software) y **correctamente estructuradas**
  - Para facilitar su implementación, todo control de seguridad de la lista debe tener al menos
    - Una **justificación** (**rationale**) que explique por qué es necesario cada control
      - Así sabremos si vale la pena implementarlo en caso de que choque con la funcionalidad legítima del sistema
    - Un procedimiento exacto para **verificar o auditar** si el control de seguridad ya está implementado
    - Un procedimiento exacto para **implementar** el control de seguridad
- **Asegúrate de sacar tus listas de controles de lugares fiables**
  - Veremos algunos en el tema siguiente
  - **¡Por favor, no uses foros, StackOverflow, artículos de blog, etc. para estas cosas!**
- **No dejes tu SO o software sin aplicarle alguna lista de controles**

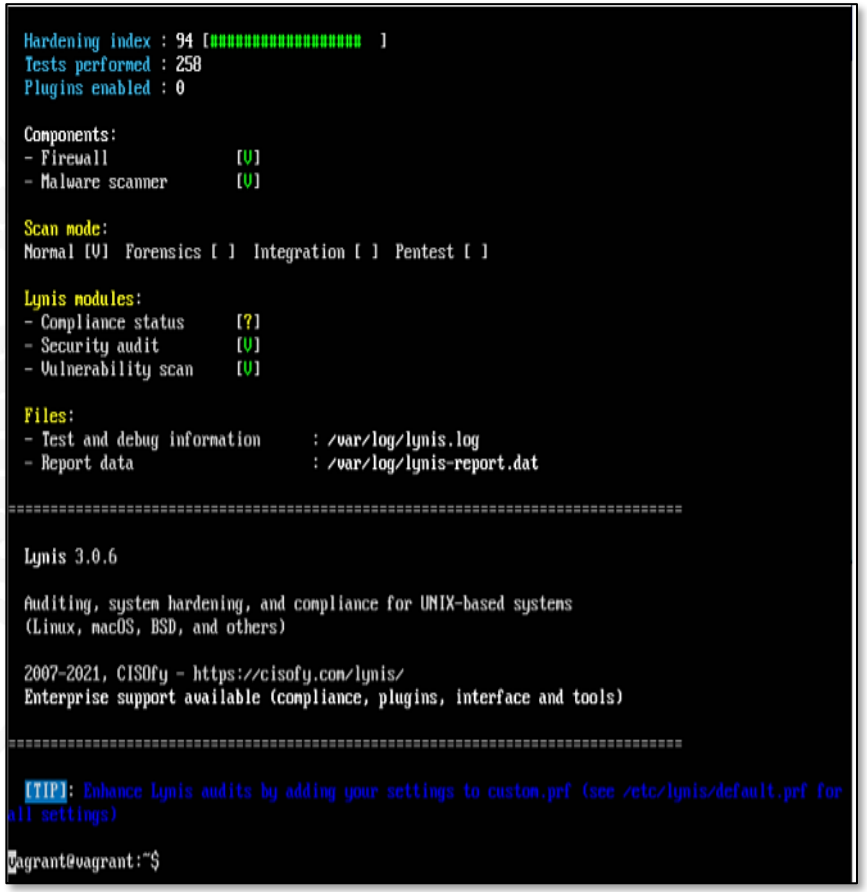
Implementando y evaluando la seguridad

- **Por lo tanto, tener una lista extensa de controles de seguridad debidamente validados es la única forma de garantizar un nivel de seguridad adecuado**
  - Pero... ¡Esta lista **no tiene que ser implementada manualmente**!
  - De hecho, ¡muchos controles de seguridad se pueden verificar y aplicar **automáticamente**!
- **Aplicar automáticamente controles de seguridad hace que en muchas situaciones se pueda lograr un alto nivel de hardening... ¡con muy poco esfuerzo!**
  - En el siguiente tema estudiaremos herramientas y protocolos que nos permiten hacer eso
  - Y, si necesitamos un control preciso, también podemos mezclar automatización y ajuste manual ☺
- **Pero ¿qué pasa si solo quiero evaluar la seguridad general de un SO?**
  - Las listas de controles de seguridad suelen permitir **un modo de solo evaluación**
  - Y podemos obtener una evaluación de seguridad automatizada con herramientas como Lynis (para sistemas Linux)
  - CLARA (https://www.ccn-cert.cni.es/es/soluciones-seguridad/clara.html) se puede usar en sistemas Windows (y algunos Linux)

LYNIS

https://github.com/CISOfy/lynis

- **Realiza un análisis de seguridad en profundidad en un Linux local, dando una "puntuación de seguridad"**
  - El objetivo principal es probar las defensas de seguridad y proporcionar consejos para un mayor hardening del sistema
  - Implementa auditoría de seguridad y detecta vulnerabilidades
  - También busca información general del sistema, paquetes de software vulnerables y posibles problemas de configuración
  - Hace pruebas de cumplimiento con políticas de seguridad (Ej.: ISO27001, PCI-DSS, HIPAA...), vistas en el **Tema 4**
- **El software (también) ayuda con**
  - Administración de parches de configuración, activos y software
  - Hardening del sistema y pentesting (escalada de privilegios)
  - Detección de intrusos
- **Tutorial**
  - https://geekytheory.com/tutorial-linux-audita-tu-sistema-con-lynis



➢ Ejercicios **L5B1_LYNIS1, L5B1_LYNIS2**

TEMA 3
SEGURIDAD DE SISTEMAS OPERATIVOS

Fuente: Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo