2026/2027

Fundamentos de Ciberseguridad

CIB

MOISÉS ALBERTO DOMÍNGUEZ CRUZ
IES LOS SAUCES – DAW2

Fundamentos de ciberseguridad

Ejercicio 1.1 2
1. Amenazas y vulnerabilidades informáticas. 2
2. Medidas de protección básicas. 12
3. Análisis de los incidentes de seguridad. 17
4. Herramientas y tecnologías de aplicación 19
5. Normativa y buenas prácticas de uso. 22
6. Auditoria de Seguridad 24
7. Control de Acceso 25
Webgrafía 34

Ejercicio 1.1

1. Amenazas y vulnerabilidades informáticas.

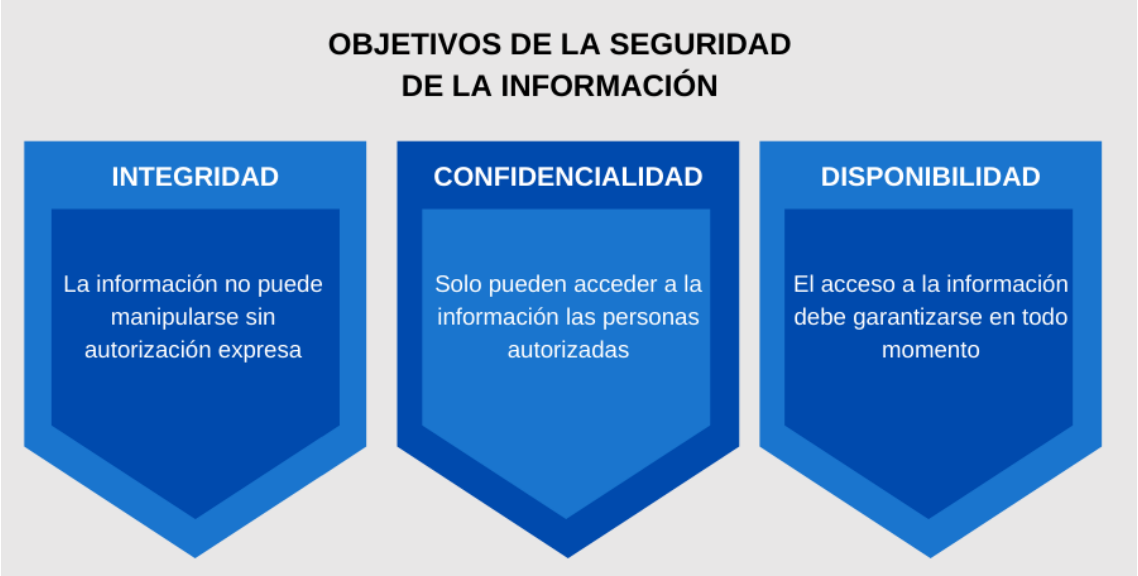
Conceptos básicos, diferencias entre amenazas activas y pasivas y su impacto.

Confidencialidad: Es la propiedad de la información que garantiza que este accesible únicamente a personal autorizado a dicha información.

Integridad: Datos o información exactos y fiables y que no han sido modificados accidentalmente o de manera intencional.

Disponibilidad: Asegura la fiabilidad y el acceso oportuno a los datos y recursos por parte de los individuos o personas autorizadas.

Diferencias → Cada uno protege un aspecto diferente del ciclo de vida de los datos.



Amenazas Pasivas

- Pérdida de privacidad y secreto industrial o comercial.
- Exposición de credenciales, datos financieros o información médica.
- Facilita la preparación de futuros ataques activos más dirigidos.

Ejemplos comunes: Sniffing, Análisis de tráfico...

Amenazas Activas

- Pérdida de integridad de los datos (información falsa o corrupta que lleva a malas decisiones operativas).
- Parálisis total o parcial de servicios críticos (pérdida económica inmediata por caída de plataformas).
- Daño reputacional severo y posibles sanciones regulatorias (por ejemplo, multas RGPD).

Ejemplos comunes: Suplantación de identidad, Modificación de mensajes, Ataques de denegación de servicio, Ransomware o inyección de malware...

Clasificación y características de las principales amenazas informáticas.

Tipos de malware

Adware: Muestra anuncios no deseados en la pantalla del ordenador o dispositivo móvil, redirige los resultados de las búsquedas a sitios web de publicidad y recopila los datos de los usuarios para venderlos a anunciantes sin su consentimiento.

Spyware: Se oculta en el dispositivo, controla tu actividad y roba información confidencial como datos financieros, información de cuentas, contraseñas y mucho más.

Ransomware: Diseñado para bloquear el acceso de los usuarios a su sistema o denegar el acceso a los datos hasta que se pague un rescate.

Troyanos: Se oculta como software legítimo con el fin de engañarte para que ejecutes software malicioso en tu equipo.

Gusanos: Son uno de los tipos de malware más comunes, se propagan en redes informáticas mediante la explotación de vulnerabilidades del sistema operativo.

Virus: Es un fragmento de código que se inserta en una aplicación y se ejecuta cuando esta se abre.

Keyloggers: Es un tipo de spyware que vigila la actividad del usuario.

Bots y botnets: Es un equipo infectado con malware que un hacker puede controlar de manera remota.

Técnicas de ingeniería social

Es una estrategia utilizada por personas o grupos para manipular y engañar a las personas con la intención de que revelen información delicada o lleven a cabo acciones que pongan en riesgo su seguridad.

Phishing



Los atacantes envían correos electrónicos, mensajes o enlaces engañosos a **sitios web que parecen legítimos** con el fin de confundir a los destinatarios para que hagan clic y revelen información delicada, como contraseñas, números de tarjetas de crédito o datos personales.

Phishing personalizado



Parecido al *phishing* pero **muy adaptado a cada persona**. Para resultar aún más convincentes, los atacantes personalizan los mensajes basándose en información concreta sobre el objetivo.

Baiting:



Los atacantes **ofrecen algo atractivo**, como descargas gratuitas de *software*, a cambio de información personal o acceso al sistema. Esta forma de actuar puede incluir archivos o enlaces infectados.

Pretexting:



Los atacantes elaboran una historia ficticia o un pretexto para obtener información. Esto implica a menudo **suplantar la identidad de alguien que inspira confianza**, como un colega o un representante de un banco.

Phishing de voz



Una forma de *phishing* que tiene lugar **mediante la comunicación por voz, habitualmente mediante una llamada telefónica**. Los atacantes se hacen pasar por entidades de confianza y convencen a las víctimas de que revelen información delicada.

Suplantación



Los atacantes **se hacen pasar por otra persona**, ya sea en línea o en persona, para ganar confianza y manipular al objetivo de forma que revele información confidencial o haga algo concreto.

Phishing en redes sociales



Los ataques **se dirigen principalmente a usuarios de redes sociales** y a menudo aprovechan un tema popular o una tendencia para crear mensajes engañosos, haciendo que parezcan pertinentes y fiables.

Programas de secuestro y extorsión



Los atacantes amenazan con **revelar información delicada** o perturbar los sistemas a menos que se pague un rescate. Infunden miedo para coaccionar a sus víctimas.

Ataques a la red

Acción deliberada para comprometer la seguridad, disponibilidad o integridad de los sistemas conectados.

Denegación de servicio (DoS/DDoS): Es un tipo de ciberataque en el que un actor malicioso tiene como objetivo que una máquina u otro dispositivo no esté disponible para los usuarios a los que va dirigido, interrumpiendo el funcionamiento normal del mismo.

Ataque de intermediario (MITM): Es un ciberataque en el que un hacker roba información confidencial espiando las comunicaciones entre dos objetivos en línea, como un usuario y una aplicación web.

Ataques a aplicaciones web

Intento por parte de agentes maliciosos de aprovechar las vulnerabilidades y debilidades de las aplicaciones web o móviles creadas durante el proceso de desarrollo de software, con el objetivo de interrumpir la actividad empresarial u obtener acceso al ecosistema de TI de una organización.

Inyección SQL: Es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una base de datos.

Cross-site scripting: es un tipo de ataque informático que permite a un actor de amenazas ejecutar código malicioso en el navegador de otro usuario.

Clasificación de vulnerabilidades en sistemas, redes y aplicaciones.

Vulnerabilidades de sistema operativo

Permiten el acceso a atacantes que dañan los dispositivos donde está instalado.

Las páginas web pueden distribuir códigos maliciosos, de manera que, cuando un usuario las visita, se descarga automáticamente un archivo que infecta el dispositivo.

Vulnerabilidades de red

Se ubican en la infraestructura del hardware, en el software o en procesos organizacionales del sistema. Este tipo de vulnerabilidad informática puede ser física o no física:

- *Físicas:* como alojar el servidor en un lugar cuyas condiciones no son aptas y pueden dañarlo —por ejemplo, en un área poco ventilada, húmeda, con riesgo de inundaciones o incendios— o donde personas no autorizadas pueden acceder de forma fácil.
- *No físicas:* como un acceso inalámbrico mal protegido, cortafuegos mal configurados o un software obsoleto que no tiene los parches adecuados para nuevas amenazas.

Vulnerabilidades de aplicaciones web

Afectan al software de uso específico, especialmente en entornos web.

Inyección de código (Injection): Inserción de comandos maliciosos (como SQL) no filtrados por la aplicación.

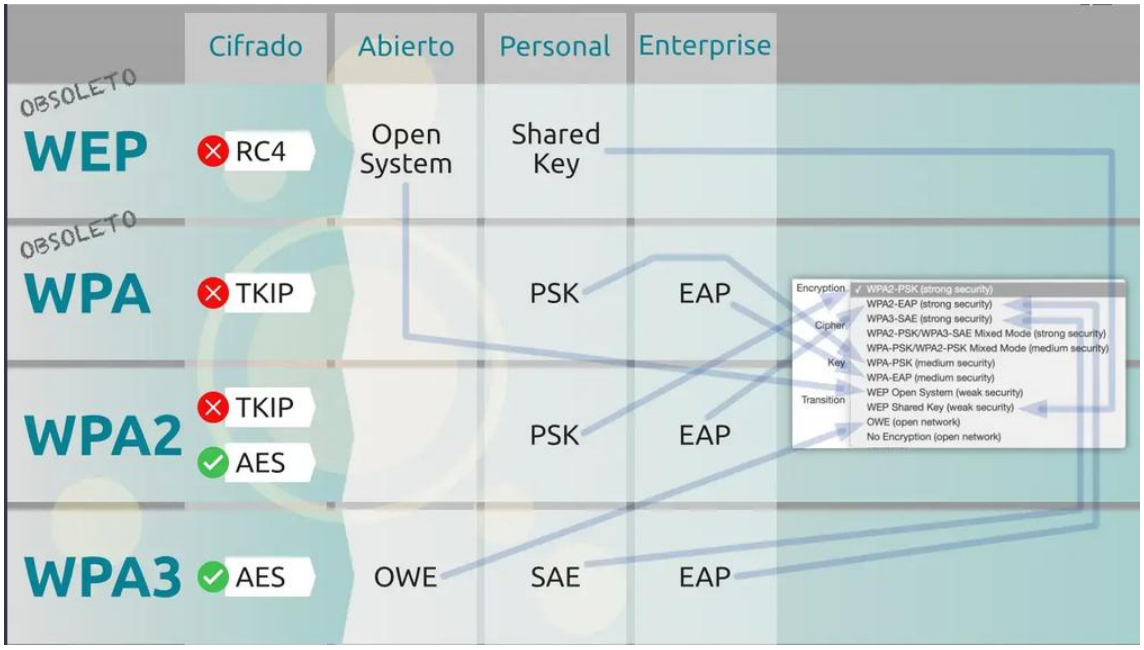
Control de acceso roto: Fallos que permiten a un usuario ver o hacer cosas sin la autorización debida.

Componentes vulnerables: Uso de librerías o dependencias de terceros desactualizadas.



Redes inalámbricas sin cifrar o con cifrados obsoletos.

Las redes inalámbricas sin cifrar o con protocolos antiguos como WEP exponen todo tu tráfico de internet a que cualquier intruso lo lea o robe en cuestión de minutos.



Relación entre amenazas y vulnerabilidades.

La relación entre amenaza y vulnerabilidad se basa en que la vulnerabilidad es la debilidad interna que permite que una amenaza externa o potencial cause un daño.

Amenaza: Un evento, agente o circunstancia externa o potencial que puede provocar un daño o ataque.

Vulnerabilidad: Un fallo, punto débil o característica propia de un sistema que lo hace susceptible a dicha amenaza.

Mecánica de explotación: identificación de vulnerabilidades por los atacantes.

Los atacantes identifican vulnerabilidades mediante análisis automatizados, investigaciones de seguridad y escaneos de red para encontrar fallos que luego puedan aprovechar.

Resumen de como DETECTAR y Explotar VULNERABILIDADES



Ejemplos prácticos de explotación: ransomware en sistemas desactualizados e inyección SQL.

Casos reales de ataques Exploit

A lo largo de los últimos años, los exploits han estado detrás de algunos de los ciberataques más sonados a nivel mundial. Estos casos muestran cómo una vulnerabilidad no corregida puede desencadenar consecuencias millonarias y afectar a miles de organizaciones.

- **WannaCry (2017)**

  Este ransomware se propagó utilizando el exploit EternalBlue, que aprovechaba una vulnerabilidad crítica en Windows. En pocos días afectó a hospitales, fábricas y empresas de todo el mundo, provocando pérdidas estimadas en más de 4.000 millones de dólares.

- **Equifax (2017)**

  Una de las mayores brechas de datos de la historia: más de 147 millones de registros personales quedaron expuestos debido a una vulnerabilidad no parcheada en Apache Struts. Los atacantes usaron un exploit conocido y ya corregido, pero la empresa no había aplicado la actualización.

- **Microsoft Exchange (2021)**

  Exploits de día cero permitieron a atacantes acceder a miles de servidores de correo corporativos en todo el mundo. El incidente derivó en robo masivo de datos y puertas traseras instaladas para acceso persistente.

- **Log4Shell (2021)**

  Una vulnerabilidad crítica en la librería Log4j, muy utilizada en aplicaciones Java, permitió la ejecución remota de código (RCE). Los exploits aparecieron en cuestión de horas y aún hoy siguen detectándose intentos de explotación.

- **SharePoint "ToolShell" (2025)**

  En 2025, varias campañas activas aprovecharon vulnerabilidades en servidores on-premises de Microsoft SharePoint (CVE-2025-49706 y CVE-2025-49704) para lograr acceso inicial y moverse lateralmente en redes corporativas.

Evaluación de las consecuencias de las amenazas y vulnerabilidades.

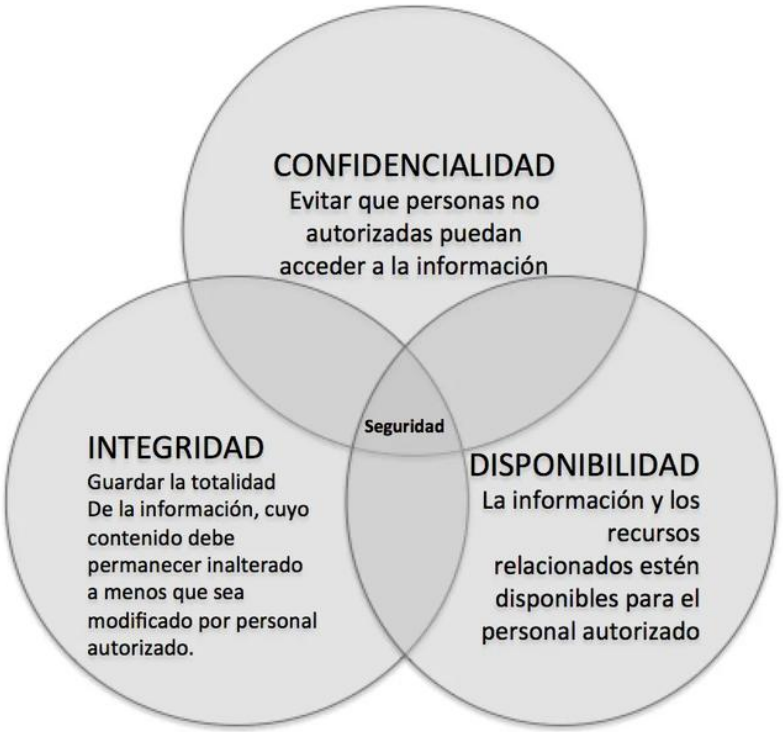
Pasos para Evaluar las Consecuencias

1. **Identificar los activos:** Reconocer qué se quiere proteger (datos, equipos, personas o infraestructura).
2. **Detectar amenazas y debilidades:** Listar los peligros posibles y las fallas de seguridad o de protección existentes.
3. **Calcular la probabilidad:** Estimar qué tan posible es que la amenaza logre explotar la vulnerabilidad.
4. **Medir el impacto:** Analizar la magnitud del daño si el evento sucede (escala de bajo a crítico o de 0 a 1).
5. **Priorizar el riesgo:** Combinar probabilidad e impacto para decidir qué problemas se deben solucionar primero.



Impacto sobre la confidencialidad, la integridad y la disponibilidad.

Determina el daño real que sufre una organización cuando un sistema o dato es atacado, robado o modificado.



Señales y síntomas de amenazas y vulnerabilidades activas.

Los principales signos y síntomas de que tu infraestructura o sistema está bajo un ataque activo o presenta fallos críticos se dividen en las siguientes categorías:

- **Lentitud extrema inexplicable:** Caídas drásticas en la velocidad del sistema, servidores o red sin un aumento real de usuarios legítimos.
- **Consumo de recursos al 100%:** Uso inusualmente alto de la CPU, memoria RAM o disco por procesos desconocidos o procesos legítimos duplicados.
- **Cuelgues y reinicios constantes:** Pantallazos azules, congelamientos o reinicios automáticos de servidores y equipos de trabajo.
- **Antivirus o firewall desactivados:** Programas de seguridad que se apagan solos o alertas de seguridad que desaparecen de los registros.
- **Modificaciones en archivos del sistema:** Archivos críticos modificados, eliminados o con permisos de acceso alterados sospechosamente.

- **Políticas de grupo cambiadas:** Alteraciones en las restricciones de red, firewalls internos o configuraciones de contraseñas sin autorización previa.
- **Bloqueos de cuentas masivos:** Múltiples usuarios bloqueados al mismo tiempo debido a intentos fallidos de inicio de sesión (síntoma de ataque de fuerza bruta).
- **Inicios de sesión anómalos:** Accesos detectados a altas horas de la noche, desde ubicaciones geográficas imposibles o desde dispositivos no registrados.
- **Nuevos usuarios administradores:** Aparición de cuentas con privilegios elevados que ningún departamento de TI ha creado.
- **Picos de tráfico saliente:** Grandes volúmenes de datos que salen de la red hacia servidores externos desconocidos (exfiltración de datos).
- **Conexiones a IPs maliciosas:** Dispositivos internos intentando comunicarse de forma repetida con direcciones IP o dominios catalogados como peligrosos.
- **Consultas DNS masivas:** Comportamiento inusual en las solicitudes de resolución de nombres, que a menudo indica comunicación con redes de bots (*botnets*).
- **Borrado de logs de auditoría:** Registros de eventos que aparecen vacíos o con saltos de tiempo inexplicables (los atacantes suelen borrar sus huellas).

2. Medidas de protección básicas.

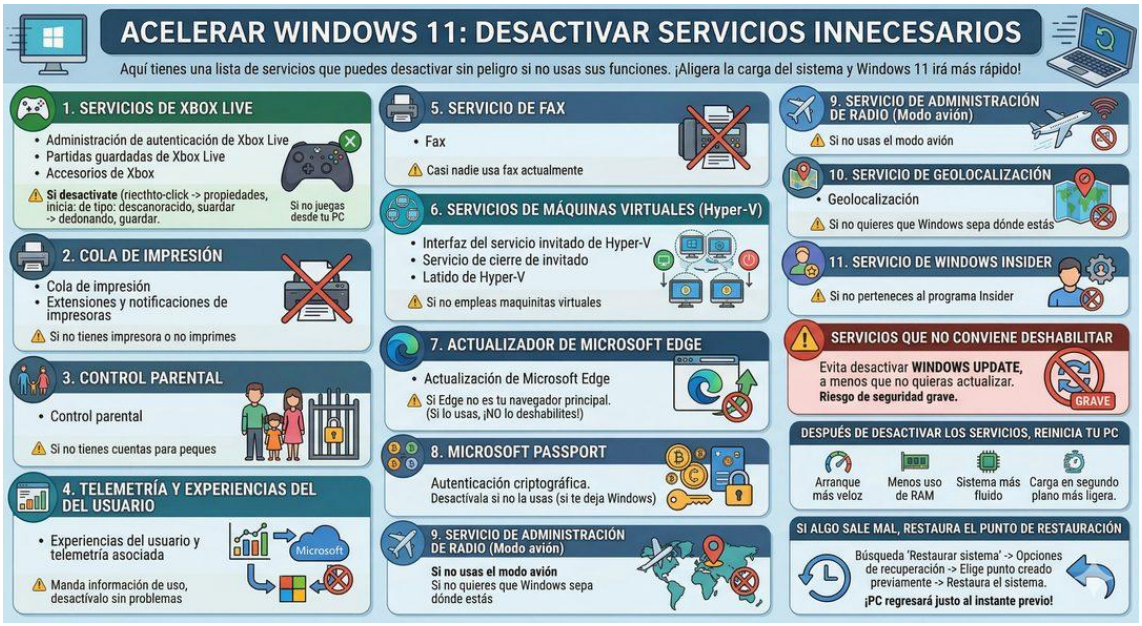
Conceptos básicos

Servicios innecesarios→ Procesos habilitados de fábrica que consumen recursos y abren puertos sin justificación operativa.

Actualizaciones→ Modificaciones de software destinadas a corregir errores funcionales y brechas de seguridad.

Vulnerabilidades conocidas→ Fallos documentados en bases de datos públicas.

Procedimientos para desactivar servicios innecesarios en sistemas operativos



Herramientas para gestionar actualizaciones y parches de seguridad

- NinjaOne Patch Management.
- ManageEngine Patch Manager Plus.
- SolarWinds Patch Manager.
- Ivanti Patch for Endpoint Manager.
- Automox.
- Patch My PC.
- PDQ Deploy.

(Fuente: https://blog.invgate.com/es/software-de-gestion-de-parches)

Gestión de contraseñas robustas

La gestión de contraseñas robustas es el conjunto de prácticas, políticas y tecnologías utilizadas para crear, almacenar, rotar y proteger las credenciales de acceso de los usuarios contra ataques de fuerza bruta, diccionario o ingeniería social.

Características de una contraseña segura

- Longitud adecuada: Mínimo de 12 a 16 caracteres. La longitud es el factor que más dificulta los ataques de fuerza bruta.
- Complejidad y variedad: Uso combinado de letras mayúsculas, minúsculas, números y caracteres especiales ($, @, %, etc.).
- Imprevisibilidad: Evitar palabras de diccionario, fechas de nacimiento, secuencias (123456) o patrones lógicos (qwerty).
- Uso de frases de contraseña (Passphrases): Frases largas compuestas por palabras aleatorias (ej. CaballoAzulVentanaGato!99), que son más fáciles de recordar para el usuario y matemáticamente muy difíciles de descifrar.

Políticas de gestión de contraseñas

- Prohibición de reutilización: Mantener un historial de contraseñas para evitar que el usuario vuelva a utilizar sus últimas 5 a 10 claves.
- Bloqueo de cuenta (Account Lockout): Bloquear temporalmente el acceso tras un número determinado de intentos fallidos (ej. 3 a 5 intentos) para mitigar ataques de fuerza bruta.
- Rotación de contraseñas: Aunque históricamente se exigía el cambio cada 90 días, estándares actuales como NIST recomiendan exigir el cambio solo si hay sospecha de compromiso, ya que la rotación forzada suele llevar a contraseñas más débiles (ej. Password1!, Password2!).
- Uso de gestores de contraseñas: Herramientas centralizadas y cifradas (como Bitwarden, 1Password o KeePass) que generan y almacenan claves únicas para cada servicio.

Métodos de autenticación alternativos: autenticación Multifactor (MFA)

- Aplicaciones autenticadoras: Programas en el móvil (como Microsoft Authenticator o Google Authenticator) que generan códigos temporales o avisos en pantalla.
- Mensajes SMS o llamadas: Códigos de un solo uso enviados al teléfono.
- Llaves de seguridad físicas: Dispositivos USB (como FIDO2) que se conectan al equipo.
- Datos biométricos: Lectura de huella o lectura de cara en el dispositivo.



Recursos multimedia:

- Video explicativo: ¿Qué es la Autenticación Multifactor (MFA)?

Control de accesos y permisos

El control de acceso es el proceso mediante el cual se restringe o se permite el uso de recursos específicos a usuarios o sistemas autenticados, basándose en un conjunto de políticas y reglas predefinidas.

- Principio de Menor Privilegio (PoLP): Otorgar a cada usuario o proceso únicamente los permisos estrictamente necesarios para realizar sus tareas, ni más ni menos.
- Necesidad de saber (Need-to-know): Incluso si un usuario tiene los privilegios técnicos para acceder a un recurso, no debe hacerlo a menos que sea necesario para su función actual.
- Separación de funciones: Dividir tareas críticas entre varios usuarios para que ninguna persona tenga control total sobre un proceso sensible, reduciendo el riesgo de fraude interno.

Gestión de roles y permisos en sistemas operativos

RBAC (Role-Based Access Control): Los permisos se asignan a roles (ej. "Administrador", "Auditor", "Usuario base") y los usuarios se asignan a esos roles, simplificando la gestión a gran escala.

- **Windows:** Utiliza NTFS para asignar permisos de lectura, escritura, modificación y control total a nivel de archivo/carpeta. A nivel de red corporativa, utiliza Active Directory (AD) mediante Grupos de Seguridad.

- **Linux/Unix:** Utiliza un sistema básico de permisos (Lectura r, Escritura w, Ejecución x) divididos en Usuario (Owner), Grupo (Group) y Otros (Others). Se gestionan con comandos como chmod y chown. Para un control más granular se utilizan las ACLs (Listas de Control de Acceso).

Configuración de accesos en redes:

VLANs (Redes de Área Local Virtuales): Segmentación lógica de la red para evitar que un departamento (ej. Invitados) acceda a los recursos de otro (ej. Finanzas).

VPNs y ZTNA (Zero Trust Network Access): Controles para usuarios externos. ZTNA asume que ninguna red es confiable y verifica la identidad y el estado del dispositivo en cada solicitud de acceso.

802.1X y NAC (Network Access Control): Autenticación requerida a nivel de puerto físico o Wi-Fi antes de que el dispositivo obtenga una IP y pueda comunicarse en la red.

Enlaces de interés:

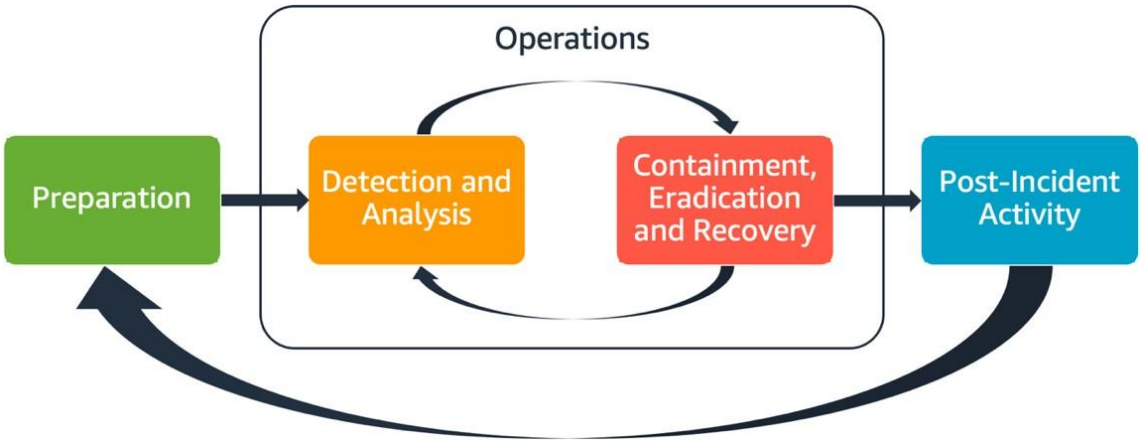
- Documentación de Microsoft sobre Control de Acceso
- Guía de administración de permisos en Linux

Implementación de reglas de firewall

Un firewall (cortafuegos) es un sistema de seguridad de red que monitoriza y controla el tráfico entrante y saliente basándose en reglas de seguridad predeterminadas. Establece una barrera entre redes confiables (internas) y redes no confiables (Internet).

3. Análisis de los incidentes de seguridad.

a) **Concepto y ciclo de vida:** Un incidente ocurre cuando una amenaza explota una vulnerabilidad, generando consecuencias negativas. Su gestión sigue el Ciclo de vida de un incidente: detección, análisis, contención, erradicación, recuperación y aprendizaje.



Proceso que sigue un equipo de seguridad para manejar un incidente:

- ***Detección:*** Identificar que ha ocurrido un incidente.
- ***Análisis:*** Determinar la causa y el alcance del incidente.
- ***Contención:*** Limitar el daño y evitar que se propague.
- ***Erradicación:*** Eliminar la amenaza de los sistemas afectados.
- ***Recuperación:*** Restaurar los sistemas y datos a un estado normal.
- ***Aprendizaje:*** Revisar el incidente para mejorar la seguridad y prevenir futuros problemas.

Enlace de interés:

- Guia oficial del NIST

b) Clasificación: Los incidentes se tipifican según su origen (internos o externos) y su magnitud (dirigidos o masivos). Para catalogarlos adecuadamente, se utilizan criterios como la naturaleza del ataque, su alcance y el objetivo final.

c) Evidencias de ataque: Los Indicadores de compromiso (IoC) son fragmentos de datos que confirman una intrusión. Se clasifican en IoCs basados en red (ej. IPs maliciosas), en sistema y en registros (ej. hashes de malware). Identificarlos permite descubrir las técnicas de ataque empleadas.

Enlace de Interés:

- Indicadores de compromiso

d) Prevención y defensa: Las Estrategias proactivas se anticipan al daño mediante la configuración de firewalls, sistemas de detección de intrusos (IDS/IPS) y una estricta gestión de contraseñas y accesos. Cada medida debe tener una justificación técnica clara.

Enlace de Interés:

- Estrategias proactivas

e) Evaluación de impacto: Toda solución defensiva debe analizarse midiendo la mitigación del riesgo frente a su viabilidad técnica y operativa. El análisis coste-beneficio garantiza que implementar una medida no sea más costoso que el activo que busca proteger.

f) Elaboración de informes: Es fundamental documentar todo el proceso utilizando herramientas y formatos estandarizados. Las buenas prácticas dicen que los hallazgos técnicos deben comunicarse de forma precisa y adaptada a la audiencia.

g) Investigación digital: El análisis forense aplica técnicas para recopilar evidencias en sistemas y redes, siendo crítica la preservación de la cadena de custodia para su validez legal. Este proceso debe regirse por principios éticos y de cumplimiento normativo, garantizando la privacidad y la confidencialidad.



Enlace de Interés:

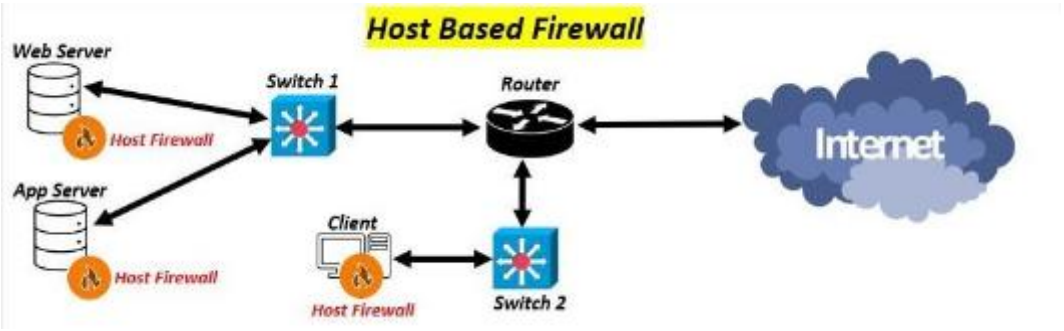
- Análisis forense

4.Herramientas y tecnologías de aplicación

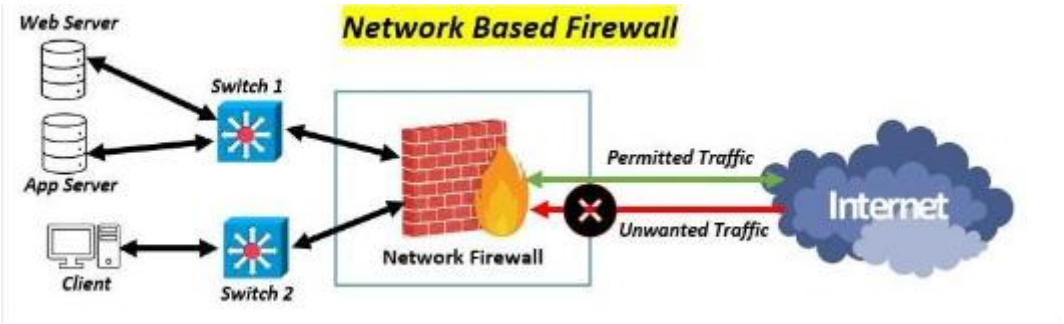
a) Introducción a las herramientas: Es fundamental conocer el concepto y tipos de herramientas de ciberseguridad, así como las funciones principales de los cortafuegos, sistemas IDS/IPS y software antivirus. De una configuración adecuada dependerá la correcta protección de los sistemas.

Contafuegos: basados en host y en red

Basados en host: Protegen un equipo específico, controlando el tráfico hacia y desde ese equipo.

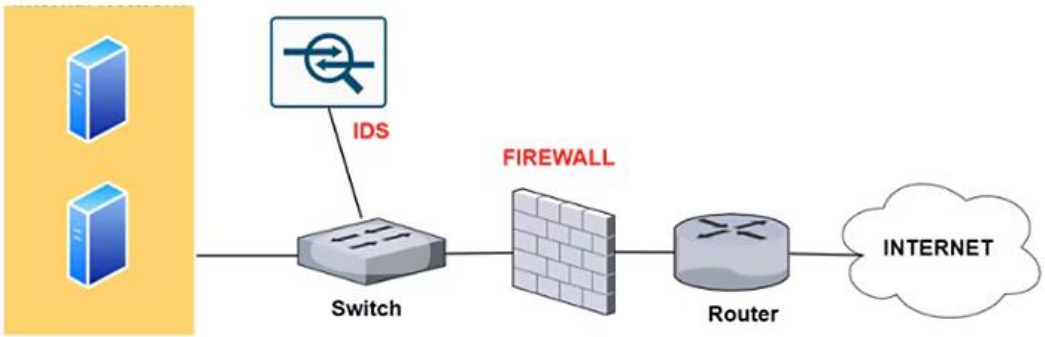


Basados en red: Protegen toda una red, filtrando el tráfico que entra y sale a nivel de red.

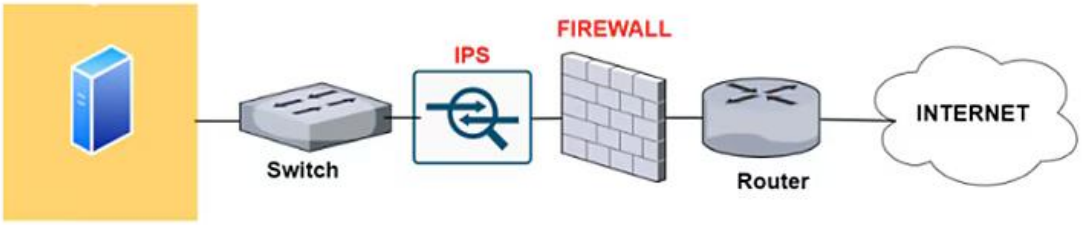


IDS/IPS

IDS: Sistema que detecta actividades sospechosas o intrusiones en una red o equipo.



IPS: Sistema de prevención de intrusiones que ayuda a identificar el tráfico malicioso y bloquea de manera proactiva el ingreso de dicho tráfico a su red.



Enlace de interés:

- IDS & IPS

Diferencias Cortafuegos vs IPS vs IDS

Parámetro	Cortafuegos	IPS	IDS
Propósito	Un dispositivo de seguridad de red que filtra el tráfico entrante y saliente basándose en reglas de seguridad predeterminadas.	Un dispositivo que inspecciona y previene las amenazas identificadas en tiempo real mediante el análisis del tráfico.	Un sistema que supervisa las actividades de la red o del sistema en busca de acciones maliciosas o violaciones de las políticas.
Operación	Filtra el tráfico basándose en reglas aplicadas a direcciones y números de puerto.	Examina el tráfico en busca de ataques en tiempo real e interviene para detenerlos en cuanto los detecta.	Observa el tráfico y busca patrones de ataque o anomalías, generando alertas.
Modo de configuración	Funciona en modo en línea o transparente en el límite de la red.	Normalmente en línea, situado después del cortafuegos dentro de la capa de red.	Generalmente funciona en modo de vigilancia, no en línea con el flujo de tráfico.
Trayectoria del tráfico	Debe ser la ruta principal del tráfico de la red.	Se coloca después del cortafuegos para examinar el tráfico filtrado.	Analiza el tráfico después de que haya atravesado el cortafuegos.
Colocación	Situado en el perímetro de la red como línea de defensa inicial.	Situado justo después de los cortafuegos, antes de la red interna.	Reside dentro de la red, normalmente después del IPS para un análisis más profundo del tráfico.
Respuesta al tráfico no autorizado	Bloquea o permite el tráfico en función de la evaluación de las reglas.	Impide activamente la progresión de las amenazas detectadas.	Emite alertas o alarmas al detectar actividades sospechosas.

Enlace de interés:

- Cortafuegos vs IPS vs IDS

Antivirus: Software diseñado para detectar, prevenir y eliminar programas maliciosos como virus, troyanos, spyware, ransomware, etc.



Enlace de interés:

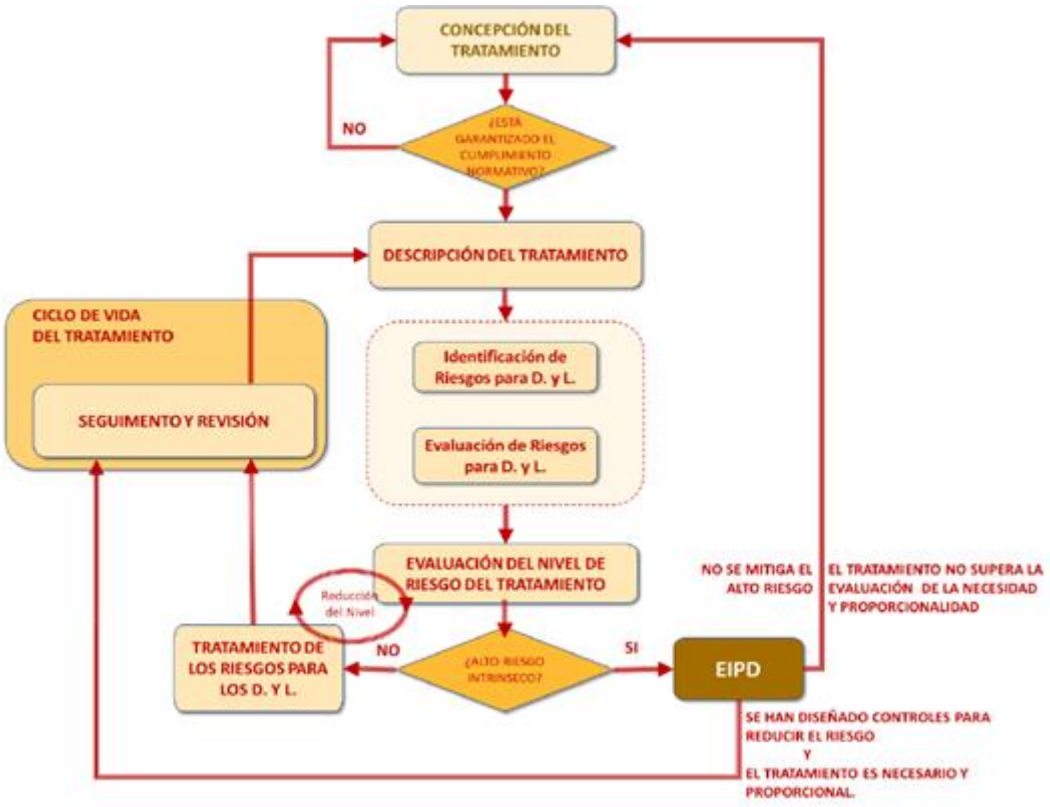
- Antivirus

5.Normativa y buenas prácticas de uso.

a) Principios de seguridad: La seguridad de la información persigue objetivos fundamentales basados en garantizar la confidencialidad, integridad y disponibilidad de los datos. Resulta de vital importancia mantener el cumplimiento normativo y aplicar buenas prácticas en todo entorno profesional.

b) Marcos reguladores: La ciberseguridad se rige por normativas y estándares tanto nacionales como internacionales. Destacan el Reglamento General de Protección de Datos (RGPD) para la privacidad, la norma ISO/IEC 27001 para sistemas de gestión, y el Esquema Nacional de Seguridad (ENS) en el ámbito español.

c) Gestión de la información: Las buenas prácticas exigen la clasificación adecuada de los datos sensibles y una estricta definición de las políticas de acceso. Asimismo, es crítico administrar adecuadamente todo el ciclo de vida de la información, desde su generación hasta su destrucción segura.



Enlaces de interés:

- Guia fuga de información INCIBE
- Control acceso INCIBE
- Gestión y evaluación del tratamiento de datos

d) Evaluación de medidas: Las medidas de seguridad deben validarse mediante herramientas de auditoría y análisis de casos en escenarios prácticos. Esto facilita un correcto Diagnóstico de fallos y la posterior formulación de Propuestas de mejora.

Enlaces de interés:

- Herramientas de auditoria INCIBE
- Guías de mejora INCIBE

e) Mejora continua y riesgos: Este proceso requiere identificar y evaluar constantemente las amenazas para desarrollar planes de acción efectivos. Para mantener la vigencia de la protección, es indispensable la revisión y actualización periódica de las políticas de seguridad.

f) Documentación y registro: Es obligatorio documentar la gestión elaborando políticas y procedimientos estructurados. Se debe mantener un Registro de incidencias utilizando formatos estándar para avalar el cumplimiento normativo, garantizando además una comunicación efectiva de las directrices a los usuarios.

Documentación Estructural

- **Políticas claras:** Definir los límites, responsabilidades y el marco normativo de la organización.
- **Procedimientos detallados:** Crear manuales paso a paso (SOP) para que el personal sepa exactamente cómo actuar ante cualquier situación.

Registro de Incidencias

- **Formatos estándar:** Utilizar plantillas homogéneas (ya sea en software especializado o formularios internos) que recojan datos críticos: fecha, descripción, impacto, acciones tomadas y responsable.
- **Evidencia auditable:** Este registro es la prueba objetiva ante auditorías externas o inspecciones de que la empresa reacciona y corrige sus fallos según la ley.

Comunicación a los Usuarios

- **Difusión efectiva:** No basta con redactar las políticas; es obligatorio capacitar a los usuarios y asegurar que las directrices sean accesibles, comprensibles y actualizadas periódicamente.

6. Auditoria de Seguridad

1. **Reconocimiento y Escaneo de Activos**
    a. **Uso de Nmap y NSE:** Empleo del escaner de redes en combinación con su motor de scripts (nse) para la exploración de infraestructura.
    b. **Inventario de activos empresariales:** Descubrimiento y catalogación de elementos críticos de la organización, abarcando direcciones Ips, puertos abiertos, URIs y Bases de datos.
    c. **Analisis de contactos:** Identificación y recopilación de contactos corporativos asociados a la superficie de ataque.

2. **Exposición Digital y Superficie de Ataque**
    a. **Seguridad de Marcas Web:** Realización de revisiones enfocadas en el análisis de seguridad de las marcas web de la entidad.
    b. **Evaluación de Exposición Digital:** Estudio del nivel de exposición digital general de la organización frente a amenazas externas e internas (MALOS / Empresa).
    c. **Inteligencia con Shodan:** Utilización de motores de búsqueda de dispositivos conectados como Shodan y sus tablas de hallazgos (Finding Table) para detectar activos expuestos en Internet.

3. **Marcos de Amenazas y Gestión de Vulnerabilidades**
    a. **Marco MITRE ATT&CK:** Aplicación de taxonomías de tácticas y técnicas basadas en el framework de MITRE ATT&CK para el análisis de vectores de ataque.
    b. **Correlación de Vulnerabilidades (NVD y CVE):** Consulta de la base de datos de vulnerabilidades NVD para la identificación y gestión de fallos de seguridad mediante sus códigos CVE.

4. **Simulación y Práctica Ofensiva**
    a. **Simulación de Ransomware en Vivo:** Análisis de escenarios de ataques avanzados y simulación de incidentes mediante ransomware live.
    b. **Laboratorios Prácticos:** Incorporación de plataformas de entrenamiento y hacking ético como Hack The Box para el desarrollo de competencias técnicas.

5. **Gestión y Documentación**
    a. **Informes de Tareas:** Elaboración de reportes de actividad orientados al registro y seguimiento de los procedimientos ejecutados (Informe Tareas).

7. Control de Acceso

El control de acceso en mi servidor **MDC-USED** (Ubuntu Server, IP `10.199.9.184`, Apache2 + PHP-FPM 8.5 + MariaDB) se organiza en **tres capas de usuarios** independientes. Cada capa tiene sus propias cuentas, su forma de autenticarse y sus permisos, y todas deben seguir el **principio de mínimo privilegio**: cada cuenta solo puede hacer lo imprescindible para su función.

7.1. Usuarios del sistema operativo – grupos – permisos

Usuarios administradores (SSH): miadmin, miadmin2, (root)

Son las cuentas que administran el servidor. Se conectan por SSH (puerto `22`, abierto en UFW) desde el CMD o MobaXterm del cliente y escalan privilegios con `sudo`. `miadmin` se creó durante la instalación de Ubuntu y `miadmin2` después, con los mismos grupos administrativos:

```
sudo useradd -m -G sudo,adm,cdrom,dip,plugdev,lxd -s /bin/bash miadmin2
sudo passwd miadmin2


ssh miadmin@10.199.9.184
```

La cuenta **root** aparece entre paréntesis porque existe, pero en Ubuntu viene **bloqueada** (sin contraseña) y solo se usa a través de `sudo`. Comprobación:

```
sudo passwd -S root
sudo sshd -T | grep -i permitrootlogin
getent group sudo
```

`passwd -S` muestra `L` (bloqueada) y por defecto SSH tiene `PermitRootLogin prohibit-password`. Para reforzarlo, en `/etc/ssh/sshd_config` se puede prohibir del todo el acceso de root y limitar qué cuentas pueden entrar por SSH:

```
PermitRootLogin no
AllowUsers miadmin miadmin2 operadorweb usuarioenjaulado
responsabledecontroldecambios


sudo systemctl restart ssh
```

Usuarios publicadores de contenidos (SFTP): operadorweb, usuarioenjaulado, responsabledecontroldecambios…

Suben y mantienen los ficheros de las aplicaciones. En MDC-USED, **operadorweb** es la cuenta de desarrollo: su home es la raíz web y su **grupo primario es www-data**:

```
sudo useradd -M -d /var/www/html -N -g www-data -s /bin/bash operadorweb
sudo passwd operadorweb
sudo chown -R operadorweb:www-data /var/www/html
```

- `-M -d /var/www/html`: no crea home nueva, su home es la raíz web.
- `-N -g www-data`: no crea un grupo propio, su grupo primario es **www-data**. Así todo lo que crea pertenece al grupo www-data y Apache puede leerlo.

- Se conecta por **SFTP** (puerto `22`) desde NetBeans, VS Code y MobaXterm, y también por SSH, porque tiene shell `/bin/bash`.

usuarioenjaulado y **responsabledecontroldecambios** solo deben usar **SFTP**, sin shell, y quedar **enjaulados (chroot)** para no poder recorrer el resto del sistema (apartado 11 de la guía del servidor):

```
sudo groupadd sftpenjaulados
sudo useradd -M -d /html/usuarioenjaulado -s /usr/sbin/nologin -g www-data -G
sftpenjaulados usuarioenjaulado
sudo passwd usuarioenjaulado
sudo mkdir /var/www/html/usuarioenjaulado
sudo chown usuarioenjaulado:www-data /var/www/html/usuarioenjaulado
sudo chmod 2750 /var/www/html/usuarioenjaulado

# /etc/ssh/sshd_config (al final del fichero)
Subsystem sftp internal-sftp
Match Group sftpenjaulados
    ChrootDirectory /var/www
    ForceCommand internal-sftp -u 0027
    AllowTcpForwarding no
    X11Forwarding no

sudo sshd -t
sudo systemctl restart ssh
```

- En Ubuntu la línea `Subsystem sftp /usr/lib/openssh/sftp-server` ya existe: hay que **sustituirla** por `Subsystem sftp internal-sftp`, no duplicarla (`sshd -t` avisa del error).
- `-d /html/usuarioenjaulado` es la home **dentro de la jaula**: al entrar, el usuario aparece directamente en `/var/www/html/usuarioenjaulado`, su carpeta de publicación. Es el propietario de esa carpeta, así que puede escribir en ella, mientras que el grupo www-data solo puede leerla.
- `-u 0027` hace que lo que suban se cree con permisos 640 (ficheros) y 750 (directorios).

Importante: el directorio debe pertenecer a root y no tener permiso de escritura para nadie más (`chown root:root /var/www` y `chmod 755 /var/www`, que es como viene en Ubuntu); si no, SSH rechaza la conexión.

Usuarios del servidor web (HTTPS): www-data

Usuario con el que se ejecuta el proceso Apache y, por tanto, el usuario que resuelve las peticiones al servicio HTTP/HTTPS. En MDC-USED hay **dos procesos** que trabajan como www-data: los hijos de **Apache2** (sirven los ficheros estáticos) y los procesos de **PHP-FPM 8.5** (ejecutan el código PHP que Apache les pasa por el socket `/run/php/php8.5-fpm.sock`). En los dos casos el proceso principal arranca como root y los que atienden las peticiones bajan a www-data.

¿En qué fichero de configuración de Apache se indica que www-data es el usuario del servicio?

En `/etc/apache2/envvars`.

En ese fichero se definen las variables, y `/etc/apache2/apache2.conf` las usa con las directivas `User` y `Group`. Como en mi servidor el PHP lo ejecuta PHP-FPM, su usuario se indica además en el pool `www`:

```
# /etc/apache2/envvars
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data

# /etc/apache2/apache2.conf
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}

# /etc/php/8.5/fpm/pool.d/www.conf
user = www-data
group = www-data
listen = /run/php/php8.5-fpm.sock
listen.owner = www-data
listen.group = www-data
```

Comprobación en MDC-USED:

```
grep APACHE_RUN /etc/apache2/envvars
grep -E '^(user|group|listen)' /etc/php/8.5/fpm/pool.d/www.conf
ps -eo user,group,cmd | grep -E 'apache2|php-fpm'
```

¿A qué grupos pertenece la cuenta www-data?

Solo a su grupo primario **www-data** (UID y GID 33 en Ubuntu).

```
id www-data
# uid=33(www-data) gid=33(www-data) grupos=33(www-data)
getent passwd www-data
# www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
```

Es una cuenta de servicio: **no tiene shell** (`/usr/sbin/nologin`), no tiene contraseña válida y **nunca debe añadirse al grupo sudo** ni a ningún otro grupo con privilegios.

¿Quién más pertenece a este grupo?

Todas las cuentas de los publicadores de contenidos, para que los contenidos que crean sean accesibles en lectura y ejecución al usuario www-data.

En MDC-USED, **operadorweb** y **usuarioenjaulado** tienen www-data como **grupo primario** (`-g www-data`). Por eso no aparecen en la lista de `getent group www-data` (que solo muestra los grupos secundarios); hay que comprobarlo con `id`:

```
id operadorweb
# uid=1002(operadorweb) gid=33(www-data) grupos=33(www-data)
getent group www-data
```

La idea clave es que **el propietario de los ficheros es el publicador y el grupo es www-data**: el publicador puede escribir y Apache/PHP-FPM, a través del grupo,

solo leer. Si se añade otro publicador que ya tiene su propio grupo, basta con `sudo usermod -aG www-data <usuario>`.

¿Alternativas al uso de www-data?

En MDC-USED todo el PHP se ejecuta en el pool por defecto `www` como www-data. Si una aplicación se compromete, el atacante puede leer los ficheros de **todas** las demás, porque todas se ejecutan con el mismo usuario. Alternativas:

- **PHP-FPM con un pool por aplicación:** como ya uso PHP-FPM 8.5, basta con crear un pool nuevo con su propio usuario sin privilegios (por ejemplo `usermiapp`) y decirle al VirtualHost de esa aplicación que use su socket. Apache sigue siendo www-data, pero el PHP de cada aplicación se ejecuta con su usuario.
- **mpm-itk:** módulo de Apache que permite asignar un usuario y grupo distintos a cada VirtualHost (`AssignUserID usuario grupo`). Es sencillo pero menos eficiente y no encaja con PHP-FPM.
- **Contenedores (Docker):** cada aplicación en su contenedor, con un usuario no root y su propio sistema de ficheros.

Ejemplo de pool propio en MDC-USED:

```
sudo useradd -r -M -s /usr/sbin/nologin -d /var/www/html/miapp usermiapp
sudo cp /etc/php/8.5/fpm/pool.d/www.conf /etc/php/8.5/fpm/pool.d/miapp.conf
sudo nano /etc/php/8.5/fpm/pool.d/miapp.conf

[miapp]
user = usermiapp
group = www-data
listen = /run/php/miapp.sock
listen.owner = www-data
listen.group = www-data
php_admin_value[open_basedir] = /var/www/html/miapp:/tmp

# En el VirtualHost de la aplicación (default-ssl.conf o el suyo propio)
<Directory /var/www/html/miapp>
    <FilesMatch "\.php$">
        SetHandler "proxy:unix:/run/php/miapp.sock|fcgi://localhost"
    </FilesMatch>
</Directory>

sudo systemctl restart php8.5-fpm
sudo systemctl reload apache2
```

Si se cambia el usuario de PHP, también hay que dar el log de Xdebug al nuevo usuario (`sudo chown usermiapp:www-data /var/log/xdebug.log`), porque ahora lo escribe `usermiapp` en lugar de www-data.

¿Permisos para la cuenta o el grupo www-data?

Actualmente en MDC-USED la raíz web tiene `chmod -R 775`. Eso significa que el **grupo www-data puede escribir en todo** `/var/www/html` y que todos los ficheros

quedan marcados como ejecutables. Es cómodo para desarrollar, pero si PHP se ve comprometido puede modificar cualquier fichero. Lo correcto es que www-data **solo lea** el código y entre en los directorios:

Elemento	Propietario:grupo	Permisos	Significado
Directorios de /var/www/html	operadorweb:www-data	2750 (rwxr-s---)	operadorweb escribe, Apache/PHP entran y leen, otros nada
Ficheros de código (.php, .html, .htaccess)	operadorweb:www-data	640 (rw-r-----)	operadorweb modifica, Apache/PHP solo leen
Fichero de credenciales de la BBDD	operadorweb:www-data	640, fuera de /var/www/html	Nunca descargable desde el navegador
uploads/, cache/ de cada aplicación	operadorweb:www-data	2770 (rwxrws---)	Únicas carpetas donde PHP escribe
/var/log/xdebug.log y /var/log/php_errors.log	www-data:www-data	640	PHP-FPM escribe sus logs

```
sudo chown -R operadorweb:www-data /var/www/html
sudo find /var/www/html -type d -exec chmod 2750 {} \;
sudo find /var/www/html -type f -exec chmod 640 {} \;

sudo touch /var/log/php_errors.log
sudo chown www-data:www-data /var/log/php_errors.log
```

El bit **setgid** (el 2 de 2750) hace que todo lo que se cree dentro herede el grupo www-data aunque lo cree otro usuario. El fichero `php_errors.log` (configurado en el `php.ini` del apartado 7.3 de la guía) debe existir y ser de www-data, o PHP-FPM no podrá escribir en él.

Además, en la configuración de Apache se limita lo que www-data puede servir. En `/etc/apache2/apache2.conf` mantengo `AllowOverride All` (lo necesito para la redirección a HTTPS por `.htaccess`), pero quito el listado de directorios:

```
# /etc/apache2/apache2.conf  (el bloque <Directory /> ya viene con Require all
denied)
<Directory /var/www/>
    Options -Indexes +SymLinksIfOwnerMatch
    AllowOverride All
    Require all granted
</Directory>

ServerName mdc-used

# /etc/apache2/conf-available/security.conf
ServerTokens Prod
ServerSignature Off

sudo apache2ctl configtest
sudo systemctl reload apache2
```

- `-Indexes`: no lista el contenido de los directorios que no tienen index.

- `SymLinksIfOwnerMatch`: impide seguir enlaces simbólicos hacia ficheros de otro propietario (por ejemplo `/etc/passwd`).
- `ServerTokens Prod` y `ServerSignature Off`: no revelan la versión de Apache ni de Ubuntu en las cabeceras ni en las páginas de error.

log de errores: en MDC-USED el `ErrorLog` del sitio HTTP está dentro de la raíz web (`/var/www/html/error/error.log`), así que cualquiera podría descargarlo en `http://10.199.9.184/error/error.log`. Se puede bloquear desde un `.htaccess` en esa carpeta, sin afectar a la página `404.html`:

```
# /var/www/html/error/.htaccess
<Files "error.log">
    Require all denied
</Files>
```

¿Cómo se controla la escritura desde la aplicación en el alojamiento de la aplicación? (no en la base de datos)

Si la aplicación puede escribir en todo su directorio (como pasa ahora con el 775), una vulnerabilidad permite al atacante modificar el propio código o dejar una webshell. Por eso se controla así:

- **Código en solo lectura para www-data** (640/2750, propietario operadorweb). La aplicación no puede modificarse a sí misma; los cambios solo llegan por SFTP con operadorweb.
- **Solo unas carpetas escribibles** por aplicación (`uploads/`, `cache/`) con grupo www-data y permisos 2770: `chmod 2770 /var/www/html/miapp/uploads`.
- **Prohibir la ejecución de PHP** en las carpetas de subida, para que un `.php` subido no se pueda ejecutar. Como tengo `AllowOverride All`, operadorweb puede hacerlo con un `.htaccess` dentro de la carpeta (con PHP-FPM no sirve `php_admin_flag engine off`, que es solo de mod_php):

```
# /var/www/html/miapp/uploads/.htaccess
Options -Indexes
<FilesMatch "\.(php|phtml|phar)$">
    Require all denied
</FilesMatch>
```

- **Restringir PHP** en `/etc/php/8.5/fpm/php.ini`: `disable_functions = exec,shell_exec,system,passthru,proc_open,popen` y límites de subida. Como tengo `upload_max_filesize = 100M`, también hay que subir `post_max_size` (por defecto `8M`) a un valor mayor, o las subidas grandes fallarán. Después: `sudo systemctl restart php8.5-fpm`.
- **Validar en la aplicación** los ficheros subidos: extensión y tipo MIME permitidos, tamaño máximo y **renombrarlos** con un nombre aleatorio (nunca usar el nombre que manda el usuario).

- **ACL** como alternativa a cambiar grupos: `sudo setfacl -m u:www-data:rwx /var/www/html/miapp/uploads`.
- **Vigilar cambios**: `inotifywait` o `auditd` sobre `/var/www/html` para detectar ficheros nuevos o modificados que no se hayan subido por SFTP. ClamAV, que ya está instalado, puede revisar la carpeta: `clamscan -r /var/www/html`.

Nota: el `php.ini` de MDC-USED tiene `display_errors = On` porque es un **servidor de desarrollo**. En producción debe estar en `Off` para no mostrar rutas y errores internos al usuario.

7.2. Usuarios del sistema gestor de base de datos (HTTPS o desde una cuenta administradora)

adminsql

Cuenta administradora de MariaDB. Se usa en lugar de `root`, porque en Ubuntu root funciona por **unix_socket**: solo puede entrar con `sudo mariadb` desde una cuenta administradora del sistema (miadmin), nunca desde la red. Para poder administrar MariaDB **desde cualquier IP** (Workbench o NetBeans en el cliente, en clase o en casa), en MDC-USED se combinan tres cosas:

- `bind-address = 0.0.0.0` en `/etc/mysql/mariadb.conf.d/50-server.cnf`: MariaDB **escucha en todas las interfaces** de red, no solo en `127.0.0.1`.
- `'adminsql'@'%'`: el comodín `%` en el host permite que adminsql se conecte **desde cualquier IP**.
- `sudo ufw allow 3306`: el cortafuegos deja pasar las conexiones al puerto de MariaDB.

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
bind-address = 0.0.0.0
sudo systemctl restart mariadb
sudo ufw allow 3306


sudo mariadb
CREATE USER 'adminsql'@'%' IDENTIFIED BY 'contraseña';
GRANT ALL PRIVILEGES ON *.* TO 'adminsql'@'%' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Formas de acceso de adminsql:

- **Desde el cliente por la red:** `mariadb -u adminsql -p -h 10.199.9.184` o un cliente gráfico al puerto `3306`.
- **Por HTTPS:** phpMyAdmin en `https://10.199.9.184/phpmyadmin`, cifrado con el certificado SSL del servidor.
- **Desde una cuenta administradora:** por SSH con miadmin y `mariadb -u adminsql -p`.

Comprobación:

```
sudo ss -tlnp | grep 3306
```

```
sudo mariadb -e "SELECT user, host, plugin FROM mysql.user;"
sudo ufw status numbered
```

Como adminsql tiene todos los privilegios y es accesible desde cualquier IP, es la cuenta más expuesta del servidor. Para mantener el acceso remoto reduciendo el riesgo:

- Usar una **contraseña robusta**, ya que cualquiera que llegue al puerto 3306 puede intentar adivinarla.
- Limitar en UFW las redes que pueden llegar al 3306 (la de clase y la de casa), en lugar de abrirlo a todo el mundo:

```
sudo ufw delete allow 3306
sudo ufw allow from 10.199.8.0/22 to any port 3306 proto tcp
sudo ufw allow from 192.168.1.0/24 to any port 3306 proto tcp
```

- Un servidor en producción no debería exponer el 3306: se dejaría `bind-address = 127.0.0.1` y se administraría por un **túnel SSH** o por phpMyAdmin con HTTPS.

UserXXXMMMMNombreAplicación

Usuario de la base de datos de la aplicación: es el que usa el código PHP para lanzar SQL contra MariaDB. Como PHP-FPM se ejecuta en el propio MDC-USED, este usuario se crea solo para **localhost**, aunque MariaDB escuche en `0.0.0.0`: así no se puede usar desde fuera aunque alguien conozca su contraseña. Tiene permisos **solo sobre su base de datos** y solo para las operaciones de datos, sin poder borrar tablas, crear usuarios ni leer ficheros del servidor:

```
CREATE DATABASE DBXXXMMMMNombreAplicacion;
CREATE USER 'UserXXXMMMMNombreAplicacion'@'localhost' IDENTIFIED BY 'contraseña';
GRANT SELECT, INSERT, UPDATE, DELETE ON DBXXXMMMMNombreAplicacion.* TO
'UserXXXMMMMNombreAplicacion'@'localhost';
FLUSH PRIVILEGES;
SHOW GRANTS FOR 'UserXXXMMMMNombreAplicacion'@'localhost';
```

Desde PHP la conexión se hace con PDO a `localhost`. Sus credenciales se guardan en un fichero de configuración con permisos 640, de operadorweb y grupo www-data, y si es posible **fuera de** `/var/www/html`, nunca en el código público ni en el repositorio.

7.3. Usuarios de la aplicación (tabla usuarios)

Son las personas que usan la aplicación web. No son usuarios de Ubuntu ni de MariaDB: se guardan como filas de la **tabla usuarios** y es la aplicación PHP la que los autentica y autoriza.

```
CREATE TABLE usuarios (
    id                INT AUTO_INCREMENT PRIMARY KEY,
    usuario           VARCHAR(50)  NOT NULL UNIQUE,
    email             VARCHAR(100) NOT NULL UNIQUE,
    password_hash     VARCHAR(255) NOT NULL,
    rol               ENUM('admin','editor','usuario') NOT NULL DEFAULT 'usuario',
    activo            TINYINT(1)   NOT NULL DEFAULT 1,
```

```
    intentos_fallidos INT          NOT NULL DEFAULT 0,
    bloqueado_hasta   DATETIME     NULL,
    ultimo_acceso     DATETIME     NULL
);
```

- **Contraseñas con hash**, nunca en texto plano: `password_hash($pass, PASSWORD_DEFAULT)` al registrar y `password_verify()` al iniciar sesión.
- **Roles**: cada página o acción comprueba en el servidor el rol del usuario, no basta con ocultar botones en la interfaz.
- **Bloqueo tras varios intentos fallidos** con `intentos_fallidos` y `bloqueado_hasta`, para frenar ataques de fuerza bruta.
- **Consultas preparadas (PDO)** para evitar inyección SQL en el login.
- **Sesiones seguras**: `session_regenerate_id(true)` tras el login y, en `/etc/php/8.5/fpm/php.ini`, cookies protegidas:

```
session.use_strict_mode = 1
session.cookie_httponly = 1
session.cookie_secure = 1
session.cookie_samesite = Lax

sudo systemctl restart php8.5-fpm
```

- **HTTPS obligatorio**: en MDC-USED ya está hecho con el `.htaccess` que redirige el puerto 80 a `https://10.199.9.184` y el sitio `default-ssl.conf` con el certificado autofirmado. Así las contraseñas y la cookie de sesión viajan cifradas (`cookie_secure` hace que la cookie solo se envíe por HTTPS).
- La cabecera **HSTS** solo debe activarse en producción con un certificado válido: con el certificado autofirmado de MDC-USED el navegador ya no dejaría aceptar la excepción de seguridad.

Resumen de las tres capas en MDC-USED

Capa	Cuentas	Acceso	Clave de seguridad
Sistema operativo – administración	miadmin, miadmin2, (root)	SSH puerto 22 (CMD / MobaXterm)	root bloqueado, grupo sudo, AllowUsers
Sistema operativo – publicación	operadorweb, usuarioenjaulado, responsabledecontroldecambios	SFTP puerto 22 (NetBeans, VS Code, MobaXterm)	Grupo primario www-data, chroot y nologin para los enjaulados
Servidor web	www-data (Apache2 y PHP-FPM 8.5)	HTTP 80 → HTTPS 443	Solo lectura salvo uploads/ y cache/
SGBD	adminsql (@'%'), UserXXXMMMMNombreAplicación (@'localhost')	3306 desde cualquier IP / phpMyAdmin HTTPS	Contraseña robusta, UFW por redes, mínimos privilegios
Aplicación	Filas de la tabla usuarios	Login web por HTTPS	Hash de contraseñas, roles, bloqueo

Webgrafía

1. Conceptos Básicos de Ciberseguridad

- Confidencialidad (Wikipedia)
- Integridad Informática (Protección Datos LOPD)
- Disponibilidad en Seguridad Informática (UNIR)
- La disponibilidad como eje de la ciberseguridad industrial (Becolve)

2. Amenazas, Malware y Tipos de Ataques

- Qué es el Sniffing y cómo evitarlo (Red Seguridad)
- Análisis de tráfico (Wikipedia)
- Definición de Spoofing (Kaspersky)
- Ataques de Denegación de Servicio - DoS (Cloudflare)
- Qué es el Ransomware (IBM)
- Tipos de Malware (Kaspersky)
- Ingeniería Social (Consejo Europeo)
- Ataques a Aplicaciones Web (Akamai)
- Qué es un Exploit en ciberseguridad (Inforges)

3. Vulnerabilidades y Análisis de Riesgos

- Tipos de vulnerabilidad informática (LinkedIn)
- Diferencias entre amenazas y vulnerabilidades (OpenWebinars)
- Vulnerabilidades, amenazas y ataques (CAIB)
- Evaluación de Vulnerabilidades (SentinelOne)
- Análisis de Riesgo en 5 pasos (LinkedIn)
- Diferencias entre amenazas, vulnerabilidades y riesgos (Sectigo)

4. Medidas de Protección y Buenas Prácticas

- Cifrado Wi-Fi y Windows 11 (Banda Ancha)
- Configuración segura de activos y software (ISC2)
- Infografía: Crea tu contraseña segura (INCIBE)
- Software de gestión de parches (Invgate)
- Documentación de Microsoft sobre Control de Acceso
- Guía de administración de permisos en Linux (RedHat)

5. Análisis y Gestión de Incidentes de Seguridad

- Alertas Tempranas e Incidentes de Seguridad (INCIBE-CERT)
- Guía oficial de Gestión de Incidentes de Seguridad Informática (NIST SP 800-61r2)
- Indicadores de compromiso - IoC (INCIBE-CERT)
- Estrategias proactivas "Shields Up" (CISA)
- Análisis Forense Digital y Respuesta a Incidentes (SANS Institute)

6. Herramientas y Tecnologías de Aplicación

- ¿Qué es un cortafuegos? (Cisco)
- Cortafuegos basados en red y en host (Palo Alto Networks)
- Diferencias entre Sistemas IDS e IPS (Fortinet)
- Definición de Sistemas de Detección de Intrusiones - IDS/IPS (Kaspersky)
- Concienciación sobre Antivirus (INCIBE)
- Software Antivirus y Antimalware (Malwarebytes)

7. Normativa, Cumplimiento y Gestión de la Información

- Reglamento General de Protección de Datos - RGPD (AEPD)
- Norma ISO/IEC 27001 para la seguridad de la información (ISO)
- Esquema Nacional de Seguridad - ENS (CCN-CERT)
- Guía sobre el ciclo de vida de los datos (AEPD)
- Guía de fuga de información y datos sensibles (INCIBE)
- Gestión del control de acceso (INCIBE)
- Gestión y evaluación del tratamiento de datos (AEPD)
- Herramientas de auditoría y diagnóstico (INCIBE)
- Guías y propuestas de mejora (INCIBE)
- Notificaciones y registro de incidencias (INCIBE-CERT)

8. Recursos Multimedia y Redes Sociales

- Vídeo: Claves para proteger tu software / OWASP (YouTube)
- Vídeo: Hacking Paso a Paso desde Cero (YouTube)
- Vídeo explicativo: ¿Qué es la Autenticación Multifactor - MFA? (YouTube)
- Publicación sobre ciberseguridad en redes sociales (X / Twitter)