Institut Puig Castellar
Santa Coloma de Gramenet

Ciberseguridad Ofensiva y Defensiva: Amenazas, Explotación y Protección (CODAEP)

Asix 2B
CFGS Administració de Sistemes Informàtics i Xarxes

Daniel Trias y Jhoan Esteban
Asix B
Curs 2025-26

Institut Puig Castellar
Santa Coloma de Gramenet

Resumen del proyecto:

La finalidad de este trabajo es investigar, conocer y poner en práctica la ciberseguridad, para esto hemos decidido seleccionar 7 ataques más comunes hoy en día hacia organizaciones con muchos datos. Estos son : Ing.Social, Virus, Redes, Apps Web, Fuerza bruta, Infraestructura y insider. Por cada ataque se realiza un método de investigación, análisis y prueba, concluyendo con lo aprendido en las pruebas de laboratorio. Aprendimos que la ciberseguridad no es solo el tecnicismo en sistemas, si no también la formación y el saber cómo actúan los usuarios, Con las bases sembradas en este proyecto, al final tendrás un conocimiento más amplio para saber actuar en futuras situaciones al trabajar para organizaciones.

Palabras clave (entre 4 i 8):

- ***Ataques***
- ***Víctima***
- ***Funcionamiento Tecnico***
- ***Ejemplos reales***
- ***Impacto y relevancia***

Institut Puig Castellar
Santa Coloma de Gramenet

Índice

3.2  Institut Puig Castellar
Santa Coloma de Gramenet

Institut Puig Castellar
Santa Coloma de Gramenet

Institut Puig Castellar
Santa Coloma de Gramenet

Institut Puig Castellar
Santa Coloma de Gramenet

Institut Puig Castellar
Santa Coloma de Gramenet

Institut Puig Castellar
Santa Coloma de Gramenet

1 Introducción

Este proyecto se centra en el estudio, análisis y simulación práctica de diferentes ataques informáticos y sus correspondientes mecanismos de defensa dentro de entornos controlados y virtualizados. El objetivo principal es comprender cómo funcionan actualmente las amenazas más relevantes en el ámbito de la ciberseguridad y qué medidas pueden aplicarse para prevenirlas, detectarlas o mitigar su impacto.

Para ello, se desarrollará un entorno de investigación basado en sistemas Linux y Windows utilizando máquinas virtuales, donde se recrearán distintos escenarios relacionados con ataques de ingeniería social, malware, ataques de red, vulnerabilidades web, explotación de sistemas y amenazas dirigidas contra infraestructuras y aplicaciones.

El proyecto combinará una parte teórica de investigación y documentación con una parte práctica basada en simulaciones controladas, permitiendo analizar tanto el comportamiento ofensivo de los ataques como las estrategias defensivas utilizadas para proteger los sistemas afectados.

Además, se emplearán diferentes herramientas relacionadas con administración de sistemas, análisis de red, monitorización y seguridad informática con el objetivo de aproximar el proyecto a entornos reales utilizados actualmente en el ámbito profesional de la ciberseguridad.

1.1 Contexto

En la actualidad, la ciberseguridad se ha convertido en uno de los pilares fundamentales dentro del sector tecnológico debido al crecimiento constante de amenazas informáticas y al aumento de servicios conectados a Internet. Empresas, instituciones y usuarios dependen diariamente de sistemas digitales, aplicaciones web, infraestructuras cloud y redes que pueden verse afectadas por ataques capaces de comprometer información, servicios y dispositivos.

Durante los últimos años, los ataques informáticos han evolucionado considerablemente tanto en complejidad como en frecuencia. Amenazas como el phishing, ransomware, ataques a aplicaciones web, robo de credenciales o explotación de vulnerabilidades afectan constantemente a organizaciones de cualquier tamaño. Además, la aparición de nuevas tecnologías como contenedores, virtualización, APIs y servicios cloud ha ampliado la superficie de ataque y ha generado nuevos desafíos en materia de seguridad.

Paralelamente, las empresas demandan cada vez más profesionales capaces de comprender no sólo la teoría de la ciberseguridad, sino también el funcionamiento práctico de los ataques y de las medidas de defensa utilizadas actualmente en entornos reales.

En este contexto surge este proyecto, orientado a la investigación y simulación práctica de diferentes tipos de ataques informáticos y mecanismos defensivos, utilizando entornos controlados y virtualizados que permitan analizar el comportamiento de las amenazas sin poner en riesgo sistemas reales.

1.2 Justificación

La principal motivación de este proyecto es profundizar de forma práctica en el ámbito de la ciberseguridad ofensiva y defensiva, aplicando conocimientos relacionados con administración de sistemas, redes y seguridad informática adquiridos durante el ciclo ASIX.

Actualmente, muchos conceptos relacionados con ataques informáticos suelen estudiarse únicamente desde una perspectiva teórica, dificultando la comprensión real de cómo funcionan las amenazas y de qué impacto pueden tener sobre sistemas e infraestructuras. Por este motivo, resulta importante desarrollar un proyecto que permita combinar investigación, documentación técnica y simulaciones prácticas en entornos seguros.

Finalmente, la ciberseguridad representa una de las áreas con mayor crecimiento y demanda laboral dentro del sector TIC, haciendo especialmente relevante la adquisición de conocimientos relacionados con análisis de vulnerabilidades, monitorización, defensa de sistemas y detección de amenazas.

1.3 Objetivos

El presente proyecto tiene como finalidad desarrollar un entorno de investigación y simulación práctica centrado en el análisis de amenazas informáticas actuales y en el estudio de las técnicas defensivas utilizadas para proteger sistemas y servicios frente a distintos tipos de ataques.

A través del proyecto se busca comprender tanto el funcionamiento técnico de los ataques como el impacto que pueden generar sobre infraestructuras reales, utilizando para ello entornos controlados y virtualizados que permitan realizar pruebas de forma segura.

1.3.1 Objetivos generales

Desarrollar un entorno práctico de investigación y simulación de ciberseguridad orientado al análisis de ataques informáticos y mecanismos de defensa, utilizando sistemas virtualizados y herramientas especializadas para estudiar el

comportamiento de amenazas actuales y las técnicas utilizadas para su prevención, detección y mitigación.

1.3.2 Objetivos específicos

Para alcanzar el objetivo general del proyecto, se plantean los siguientes objetivos específicos:

- Investigar diferentes tipos de amenazas y ataques informáticos presentes actualmente en el ámbito de la ciberseguridad.
- Analizar técnicamente ataques relacionados con ingeniería social, malware, redes, aplicaciones web y explotación de vulnerabilidades.
- Comprender el funcionamiento de las técnicas utilizadas por atacantes para comprometer sistemas y servicios.
- Estudiar mecanismos defensivos orientados a la detección, prevención y mitigación de amenazas.
- Implementar un entorno virtualizado seguro para realizar simulaciones prácticas de ataques y defensas.
- Utilizar sistemas operativos Linux y Windows durante el desarrollo de las pruebas prácticas.
- Emplear herramientas de análisis, monitorización y seguridad informática utilizadas habitualmente en entornos profesionales.
- Documentar el funcionamiento, impacto y medidas de protección asociadas a cada ataque estudiado.
- Aplicar conocimientos adquiridos durante el ciclo ASIX relacionados con administración de sistemas, redes y seguridad informática.
- Desarrollar capacidades de investigación, análisis técnico y resolución de problemas relacionados con ciberseguridad.
- Preparar una demostración práctica que permita exponer de forma visual y comprensible el funcionamiento de algunos de los ataques y defensas analizados durante el proyecto.

1.4 Estrategia y planificación del proyecto

Para el desarrollo de este proyecto se han valorado diferentes estrategias relacionadas con la investigación, simulación y análisis de amenazas informáticas. Entre las posibles opciones se encontraban el desarrollo de una herramienta completamente nueva, la adaptación de plataformas ya existentes o la creación de un entorno práctico basado en tecnologías y herramientas de ciberseguridad utilizadas actualmente en entornos profesionales.

Finalmente, la estrategia seleccionada consiste en desarrollar un entorno de investigación y simulación utilizando máquinas virtuales, sistemas Linux y Windows, y distintas herramientas especializadas de seguridad informática para recrear escenarios controlados de ataque y defensa.

Esta estrategia se considera la más adecuada debido a varios factores. En primer lugar, permite trabajar con tecnologías reales ampliamente utilizadas en el ámbito profesional de la ciberseguridad, facilitando un aprendizaje mucho más práctico y cercano a situaciones reales. Además, utilizar entornos virtualizados proporciona un entorno seguro y aislado donde pueden realizarse pruebas sin comprometer sistemas externos o infraestructuras reales.

El proyecto se dividirá en diferentes bloques temáticos relacionados con los principales tipos de amenazas estudiadas actualmente en ciberseguridad, incluyendo ataques de ingeniería social, malware, ataques de red, vulnerabilidades web, ataques contra contraseñas y explotación de infraestructuras y servicios.

Cada bloque combinará:

- Investigación teórica
- Análisis técnico
- Estudio de medidas defensivas
- Documentación de resultados

La planificación del proyecto se realizará de forma progresiva, comenzando por la investigación y preparación del entorno virtualizado, continuando con el análisis y simulación de los distintos ataques, y finalizando con la documentación, validación de resultados y preparación de la exposición final.

Desde el punto de vista de viabilidad, el proyecto resulta completamente realizable debido a que la mayor parte de las tecnologías y herramientas utilizadas son de acceso libre o educativo, permitiendo desarrollar el entorno práctico sin necesidad de una infraestructura empresarial compleja. Además, el uso de virtualización facilita repetir pruebas, restaurar sistemas y mantener un entorno estable durante todo el desarrollo del proyecto.

1.5 Metodología de trabajo

Para el desarrollo de este proyecto se seguirá una metodología de trabajo híbrida basada principalmente en un enfoque incremental y práctico, combinando características de metodologías tradicionales y metodologías ágiles.

Debido a que el proyecto se centra en investigación, pruebas técnicas y simulaciones prácticas, resulta necesario mantener cierta flexibilidad durante el desarrollo, ya que algunos escenarios, herramientas o ataques pueden requerir modificaciones o ajustes conforme avanza el trabajo.

La metodología aplicada se dividirá en varias fases principales:

1. Investigación y recopilación de información.
2. Diseño y preparación del entorno virtualizado.

3. Desarrollo de simulaciones prácticas.
4. Implementación de mecanismos defensivos.
5. Análisis de resultados.
6. Documentación técnica y preparación de la exposición final.

Para la organización y seguimiento del proyecto se utilizarán diferentes herramientas de planificación y documentación, entre ellas:

- Diagramas de Gantt y sistema Kanban (Click Up) para la planificación temporal de tareas.
- Documentación técnica estructurada.
- Herramientas de virtualización para la gestión del entorno de pruebas.
- Sistemas de captura y grabación de pantalla para documentar evidencias prácticas.
- Repositorios y almacenamiento organizado de scripts, configuraciones y resultados obtenidos.

2.1 Previsión de tareas de investigación

El desarrollo del proyecto se organizará mediante diferentes tareas de investigación y simulación práctica orientadas al análisis de amenazas informáticas y mecanismos de defensa. Cada una de estas tareas permitirá avanzar progresivamente en la comprensión técnica de los ataques estudiados y en la implementación de medidas defensivas dentro del entorno virtualizado.

Las tareas previstas para el desarrollo del proyecto son las siguientes:

2.2 Preparación del entorno de trabajo

- Instalación y configuración del software de virtualización.
- Creación de máquinas virtuales para atacantes y víctimas.
- Configuración de redes virtuales aisladas.

2.3 Investigación de amenazas y ataques

- Investigación de ataques de ingeniería social.
- Estudio de malware y técnicas de propagación.
- Investigación de ataques de red.
- Análisis de ataques a aplicaciones web.
- Investigación de ataques a contraseñas.
- Estudio de explotación de vulnerabilidades y escalada de privilegios.
- Investigación de ataques dirigidos a APIs y entornos virtualizados.

2.4 Simulación práctica de ataques

- Simulación de ataques de phishing controlados.
- Simulación controlada de malware en entornos aislados.
- Pruebas de sniffing y captura de tráfico.
- Simulación de ataques Man-in-the-Middle.
- Pruebas de fuerza bruta y diccionario.
- Pruebas de explotación de vulnerabilidades.
- Simulación de escalada de privilegios.

2.5 Implementación de mecanismos defensivos

- Configuración de firewalls y filtrado de tráfico.
- Aplicación de hardening básico en sistemas.
- Implementación de medidas de protección web.
- Configuración de herramientas de monitorización.
- Análisis de logs y eventos de seguridad.
- Aplicación de medidas de detección y mitigación.

2.6 Documentación y análisis

- Captura de evidencias de las pruebas realizadas.
- Elaboración de documentación técnica.
- Comparativa entre ataques y defensas.
- Análisis de impacto y riesgos asociados.
- Redacción de conclusiones técnicas.

2.7 Preparación de la exposición final

- Selección de pruebas prácticas más representativas.
- Organización de demostraciones visuales.
- Preparación de capturas y material gráfico.
- Preparación de la defensa oral del proyecto.

2.8 Caso de uso: atacante

El usuario atacante utilizará herramientas de análisis y explotación para realizar simulaciones controladas de distintos tipos de ataques dentro del entorno virtualizado. Entre las acciones previstas se encuentran el análisis de red, explotación de vulnerabilidades, pruebas de fuerza bruta y ataques a aplicaciones web.

2.9 Caso de uso: defensor

El usuario defensor será responsable de analizar el comportamiento de los ataques, monitorizar tráfico y eventos de seguridad, aplicar medidas defensivas

y comprobar la efectividad de las configuraciones de protección implementadas.

2.10 Tecnologías

2.10.1 Comparativa de tecnologías valoradas

Durante el desarrollo del proyecto se han valorado distintas tecnologías y herramientas relacionadas con virtualización, sistemas operativos, análisis de red y seguridad informática.

VirtualBox vs VMware

VirtualBox destaca por ser una solución gratuita, ligera y sencilla de utilizar en entornos educativos, permitiendo crear y gestionar máquinas virtuales fácilmente.

VMware ofrece un mayor rendimiento y opciones avanzadas de virtualización, aunque requiere más recursos y algunas funcionalidades son de pago.

Debido a la facilidad de acceso y compatibilidad con el entorno disponible, VirtualBox ha sido la opción seleccionada para el proyecto.

Kali Linux vs Parrot Security OS

Kali Linux es una de las distribuciones más utilizadas en ciberseguridad ofensiva y cuenta con una gran cantidad de herramientas preinstaladas y documentación disponible.

Parrot Security OS ofrece funcionalidades similares, incluyendo herramientas de privacidad y menor consumo de recursos.

Finalmente, se ha seleccionado Kali Linux debido a su amplia adopción profesional y abundante documentación técnica.

Wireshark vs tcpdump

Wireshark permite analizar tráfico de red mediante una interfaz gráfica muy visual y sencilla de interpretar.

tcpdump funciona desde línea de comandos y resulta más ligero y rápido en determinados escenarios.

Para este proyecto se utilizará principalmente Wireshark debido a la facilidad de visualización y análisis durante las simulaciones prácticas.

2.2.2 Tecnologías escogidas

Virtualización

- VirtualBox

Sistemas operativos

- Kali Linux
- Ubuntu Server
- Windows

Herramientas de análisis y seguridad

- Wireshark
- Nmap
- Burp Suite
- Hydra
- John the Ripper
- Metasploit Framework

2.11 Estructura del proyecto

El proyecto estará basado en un entorno virtualizado diseñado para simular distintos escenarios de ataque y defensa dentro de una infraestructura controlada y aislada. La estructura general del laboratorio permitirá recrear comunicaciones entre sistemas, análisis de tráfico, explotación de vulnerabilidades y aplicación de medidas defensivas sin afectar sistemas reales externos.

La arquitectura del proyecto estará compuesta por diferentes máquinas virtuales conectadas mediante una red interna virtual, donde cada sistema tendrá una función específica dentro de las simulaciones prácticas.

La estructura general del entorno estará formada por:

- Máquina atacante.
- Máquinas víctimas.
- Red virtual aislada.
- Herramientas ofensivas.
- Herramientas defensivas.
- Sistemas de monitorización y análisis.

El sistema atacante será utilizado para ejecutar simulaciones relacionadas con análisis de red, explotación de vulnerabilidades, ataques web y pruebas de acceso no autorizado. Las máquinas víctimas permitirán recrear distintos escenarios de compromiso sobre sistemas Linux y Windows.

La comunicación entre los distintos componentes se realizará mediante redes virtuales internas configuradas dentro del software de virtualización, permitiendo controlar completamente el tráfico generado durante las pruebas.

Además, se implementarán herramientas de análisis y monitorización capaces de capturar tráfico de red, registrar eventos y analizar el comportamiento de los ataques durante las simulaciones prácticas.

Conceptualmente, la estructura del proyecto seguirá un modelo dividido en tres áreas principales:

- Entorno ofensivo.
- Entorno vulnerable.
- Entorno defensivo y de monitorización.

2.12 Descripción de los componentes

2.12.1 Máquina atacante

La máquina atacante será el sistema principal utilizado para realizar las simulaciones ofensivas del proyecto. Este componente estará basado principalmente en Kali Linux debido a la gran cantidad de herramientas de ciberseguridad que incorpora por defecto.

Entre sus principales funciones destacan:

- Escaneo de redes y servicios.
- Análisis de vulnerabilidades.
- Simulación de ataques web.
- Captura y análisis de tráfico.
- Pruebas de fuerza bruta.
- Simulación de ataques de red.

La máquina atacante incluirá herramientas como:

- Nmap
- Wireshark
- Burp Suite
- Hydra

2.12.2 Máquina víctima Windows

La máquina víctima Windows estará orientada principalmente a pruebas relacionadas con malware, ingeniería social, ataques a contraseñas y análisis de configuraciones inseguras en entornos Windows.

Las principales funciones de este componente serán:

- Simulación de ataques de phishing.
- Ejecución controlada de malware en entornos aislados.
- Pruebas de credenciales y autenticación.
- Análisis de permisos y privilegios.
- Aplicación de medidas defensivas y monitorización.

Este sistema permitirá recrear escenarios similares a los utilizados habitualmente en entornos empresariales reales.

2.12.3 Red virtual

La red virtual será el componente encargado de conectar todas las máquinas utilizadas durante el proyecto dentro de un entorno aislado y seguro.

Sus funciones principales serán:

- Permitir comunicación entre máquinas virtuales.
- Facilitar simulaciones de tráfico y ataques de red.
- Aislar completamente el laboratorio del exterior.
- Controlar el flujo de comunicaciones.
- Permitir análisis y captura de paquetes.

La red se configurará utilizando las herramientas de virtualización proporcionadas por VirtualBox, creando redes internas independientes para evitar riesgos externos.

2.12.4 Herramientas de monitorización y defensa

Este componente incluirá distintas herramientas y mecanismos destinados a detectar, registrar y analizar las actividades realizadas durante las simulaciones.

Entre sus principales funciones destacan:

- Captura de tráfico de red.
- Análisis de logs.
- Detección de actividad sospechosa.
- Monitorización de eventos.
- Aplicación de medidas defensivas.

2.12.5 Sistema de documentación y evidencias

Este componente estará orientado a registrar todas las pruebas realizadas durante el proyecto mediante documentación técnica y evidencias visuales.

Sus funciones principales serán:

- Captura de pantallas.
- Grabación de simulaciones.
- Organización de resultados.
- Elaboración de documentación técnica.
- Preparación del contenido para la exposición final.

3. Ingeniería Social

3.1.1 Definición de Ingeniería Social

La ingeniería social puede definirse como el conjunto de técnicas utilizadas para manipular a una persona con el objetivo de obtener información confidencial, acceso a sistemas o la realización de determinadas acciones que ayuden al atacante. En lugar de atacar directamente la infraestructura informática, el atacante intenta aprovechar la confianza, el desconocimiento o la distracción de una víctima.

En la mayoría de los casos, los sistemas de seguridad pueden encontrarse muy bien configurados, pero un simple error o desconocimiento humano puede comprometer toda la protección de una empresa. Por este motivo, la ingeniería social se considera actualmente una de las amenazas más difíciles de prevenir. Quien sabe si el día de mañana un usuario abre un link con malware.

También los atacantes suelen apoyarse en elementos psicológicos como la urgencia, la autoridad o el miedo para aumentar las probabilidades de éxito. De esta manera, consiguen que la víctima actúe impulsivamente sin analizar con detalle la situación.

3.1.2 Evolución de la Ingeniería Social

Las primeras formas de ingeniería social aparecieron muchísimo antes del Internet. Antiguamente, muchos ataques eran en llamadas telefónicas, falsificación de identidades o intentos de acceso físico a instalaciones restringidas. El objetivo principal seguía siendo lo mismo: engañar a las personas para obtener algún tipo de beneficio.

Con el crecimiento de la comunicación digital, estas técnicas evolucionaron rápidamente. El correo electrónico permitió realizar campañas masivas de fraude, spam, etc. A su vez las redes sociales facilitaron el acceso a información personal y profesional de millones de usuarios.

En la actualidad, los ataques son considerablemente más profundos. Los delincuentes analizan perfiles públicos, investigan hábitos y recopilan datos antes de iniciar el ataque. Esto les permite crear mensajes más personalizados mucho más creíbles que los ataques genéricos.

Y también con la aparición de tecnologías como la IA (Inteligencia Artificial) o los deepfakes está incrementando el nivel de realismo de las suplantaciones. Hoy en día es posible falsificar voces, imágenes e incluso videollamadas con una calidad bastante aceptable como para engañar a muchísimos usuarios.

3.1.3 Objetivos de la Ingeniería Social

Los ataques de ingeniería social persiguen distintos objetivos dependiendo del interés del atacante. Uno de los más frecuentes es el robo de credenciales de acceso, como contraseñas, datos bancarios o información corporativa sensible.

Otra finalidad habitual es la distribución de malware. Muchos ataques de ransomware comienzan mediante correos electrónicos fraudulentos o enlaces maliciosos que convencen a la víctima para ejecutar archivos infectados.

También existen campañas orientadas al espionaje empresarial o institucional. En estos casos, el atacante busca acceder a documentos internos, comunicaciones privadas o información estratégica que posteriormente pueda utilizarse con fines económicos o políticos.

Por otro lado, algunos ataques tienen como objetivo provocar daños reputacionales o generar pérdidas económicas dentro de una organización mediante sabotaje, filtraciones de datos o difusión de información falsa. Como el famoso caso de Anonymous.

3.1.4 Factores Psicológicos Utilizados

La ingeniería social funciona porque abusa de nuestras emociones y comportamientos comunes. Algunos de los factores psicológicos más utilizados son:

- **Autoridad:** el atacante se hace pasar por una figura importante, como un jefe, un técnico o una entidad bancaria.
- **Urgencia:** se presiona a la víctima para actuar rápidamente y evitar que piense con calma.
- **Miedo:** amenazas de bloqueo de cuentas, sanciones o pérdida de datos.
- **Curiosidad:** archivos o mensajes llamativos que despiertan interés.
- **Confianza:** suplantación de contactos conocidos o empresas legítimas.
- **Recompensa:** promesas de premios, descuentos o beneficios falsos.

Estos factores explican por qué incluso usuarios experimentados pueden caer en un engaño si el contexto emocional es adecuado.

3.1.5 Público Objetivo

Cualquier persona puede convertirse en víctima de un ataque de ingeniería social, aunque algunas personas resultan especialmente más atractivas para los atacantes. Por ejemplo personas mayores poco informadas

Los usuarios domésticos suelen ser objetivo de fraudes bancarios, robos de cuentas personales o estafas relacionadas con compras online y redes sociales. Los atacantes aprovechan el desconocimiento técnico/informático o la falta de medidas de protección bastantes básicas.

Dentro de las empresas, los empleados son un objetivo especialmente apetecible debido al acceso que pueden tener a sistemas internos o información sensible. Departamentos como recursos humanos, administración o soporte técnico suelen recibir ataques dirigidos con frecuencia.

Las administraciones públicas y organismos gubernamentales también son objetivos habituales debido al valor de la información que manejan y al impacto que genera una filtración.

En ataques más avanzados, conocidos como **spear phishing**, los delincuentes seleccionan cuidadosamente a las víctimas al realizar investigaciones previas sobre su cargo, funciones o nivel de acceso.

3.1.6 Frecuencia y Relevancia Actual

Actualmente, la ingeniería social es uno de los métodos de ataque más utilizados en el ámbito de la ciberseguridad. incidentes graves que comienzan con un correo fraudulento, una llamada falsa o un mensaje diseñado para engañar a la víctima.

La gran efectividad se debe a que manipular a una persona suele ser más sencillo que vulnerar sistemas protegidos técnicamente. Incluso organizaciones con infraestructuras avanzadas pueden verse comprometidas si un usuario entrega sus credenciales o ejecuta un archivo malicioso.

El teletrabajo y la comunicación digital también ha incrementado considerablemente este tipo de amenazas. Los usuarios reciben diariamente una gran cantidad de mensajes, enlaces y notificaciones, lo que facilita cometer errores por distracción

3.1.7 Facilidad del Ataque

Los aspectos más preocupantes de la ingeniería social es que muchos ataques no requieren conocimientos avanzados. simplemente basta con crear un mensaje convincente o utilizar información pública para construir un engaño bastante creíble. Ya lo veremos en la práctica.

Existen herramientas automatizadas capaces de enviar mensajes de phishing en cuestión de minutos. y con lo simple que es en Internet encontrarse kits preparados con plantillas falsas, páginas clonadas y sistemas destinados al robo de credenciales.

Las redes sociales también facilitan enormemente el trabajo de los atacantes. Información aparentemente inofensiva, como fotografías, ubicaciones, horarios o relaciones laborales, puede utilizarse para personalizar ataques y aumentar su credibilidad.

Como consecuencia, cualquier usuario conectado a Internet puede convertirse en un posible objetivo, independientemente de su nivel técnico o experiencia.

3.1.8 Importancia de la Defensa

La protección frente a la ingeniería social requiere combinar diferentes soluciones tanto técnicas así como la formación continua. Ninguna medida técnica resulta completamente eficaz si los usuarios no son capaces de identificar comportamientos sospechosos.

La concienciación es uno de los elementos más importantes dentro de la prevención. Enseñar es la clave. desconfiar de solicitudes urgentes y evitar compartir información sensible. Con esto las probabilidades de sufrir un ataque son muy bajas.

A nivel técnico, las organizaciones utilizan filtros antiphishing, autenticación multifactor, herramientas de detección de amenazas y sistemas de monitorización para reforzar la seguridad.

También es fundamental establecer protocolos internos claros. Por ejemplo, muchas empresas verifican mediante un segundo canal cualquier solicitud relacionada con transferencias bancarias o cambios de contraseñas.

La realidad demuestra que el factor humano continúa siendo el puntos más vulnerable dentro de cualquier sistema de seguridad. Por ello, la formación y la cultura preventiva han adquirido una importancia cada vez mayor.

3.2 Phishing

3.2.1 Definición

El phishing es una técnica de ingeniería social basada en la suplantación de identidad. El atacante se hace pasar por una empresa, servicio o entidad legítima con el objetivo de engañar a la víctima y obtener información confidencial.

A veces, estos ataques se realizan mediante correos electrónicos, aunque también pueden aparecer en mensajes SMS, aplicaciones de mensajería instantánea o redes sociales.

El propósito es generar suficiente confianza para que la víctima acceda a un enlace falso, descargue un archivo malicioso o introduzca sus datos personales.

3.2.2 Funcionamiento Técnico

Hay muchos métodos técnicos pero generalmente el atacante crea páginas web falsas con una apariencia muy similar a la original o simplemente la misma página pero con diferentes URL. después, envía mensajes diseñados para generar preocupación o urgencia, obligando al usuario a actuar rápidamente.

Cuando la víctima introduce sus credenciales en la página fraudulenta, la información queda almacenada directamente en poder del atacante.

En otras situaciones, los enlaces incluidos en el mensaje suelen descargar malware automáticamente o redirigen al usuario hacia sitios preparados para explotar vulnerabilidades del navegador.

Hoy en día, muchos ataques utilizan dominios casi idénticos a los originales y certificados HTTPS válidos, dificultando todavía más encontrarlos.

3.2.3 Tipos de Phishing

Existen diferentes variantes de phishing dependiendo de los métodos que el atacante haya utilizado y de la víctima seleccionada.

- **El phishing** masivo consiste en campañas enviadas simultáneamente a miles de usuarios sin una personalización específica.
- **El spear phishing,** es lo contrario, se dirige a víctimas concretas tras realizar investigaciones previas para aumentar la credibilidad del ataque.

- **El whaling** se enfoca principalmente en directivos o altos cargos empresariales debido al elevado valor de la información a la que tienen acceso.

También existen modalidades como el smishing, que es mediante mensajes SMS, y el vishing, basado en llamadas telefónicas fraudulentas.

3.2.4 Impacto y Prevención

El phishing continúa siendo una de las amenazas más utilizadas debido a su alta efectividad. Muchas infecciones de ransomware y robos de información comienzan mediante este tipo de engaños.

Para reducir la probabilidad, es importante verificar siempre el remitente de los mensajes, evitar acceder a enlaces sospechosos y utilizar autenticación multifactor.

Las empresas también aplican filtros de seguridad, análisis automatizados y simulaciones de ataques para mejorar la preparación de los empleados.

3.2.5 Casos Reales de Phishing

En los últimos años, han habido numerosos incidentes de ciberseguridad que nos han demostrado el enorme impacto que puede llegar a tener el phishing tanto en usuarios particulares como en grandes organizaciones.

1. **Primer Caso**

Uno de los casos más conocidos ocurrió en Julio de 2020, Twitter sufrió uno de los incidentes de ingeniería social más conocidos de los últimos años. Los atacantes consiguieron engañar a empleados de la compañía mediante llamadas telefónicas y técnicas de suplantación de identidad.obteniendo acceso a herramientas internas, lograron acceder a cuentas verificadas de personalidades públicas y empresas reconocidas con millones de seguidores. publicando mensajes fraudulentos relacionados con criptomonedas para estafar a los usuarios

2. **Segundo Caso**

Uno de los fraudes más famosos relacionados con phishing empresarial afectó a Google y Facebook entre los años 2013 y 2015. Un atacante lituano llamado Evaldas Rimasauskas consiguió engañar a las dos compañías mediante correos electrónicos falsificados y facturas fraudulentas. El atacante se hizo pasar por un proveedor legítimo con el que ambas empresas trabajan habitualmente.Los empleados responsables de pagos autorizaron

transferencias millonarias creyendo que las solicitudes eran reales. En total, el fraude superó los 100 millones de dólares.

3. **Último Caso interesante**

Como último caso tenemos el de Sony Pictures en el año 2014, Varios atacantes usando métodos phishing enviaban correos maliciosos, consiguiendo credenciales e información muy confidencial, filtrando películas, datos de empleados, documentos importantes, etc. La empresa tuvo demasiadas pérdidas al punto de casi caer en quiebra y dañando toda su reputación.

Obviamente hay muchísimos más casos, pero con esto corroboramos que si el factor humano no se corrige hasta los más grandes y adinerados son sencillamente atacables.

3.2.6 El phishing moderno

El phishing ha evolucionado notablemente en los últimos años debido al desarrollo de nuevas tecnologías y al aumento de información pública disponible en Internet. Inicialmente, muchos ataques eran fácilmente identificables debido a errores ortográficos, diseños poco profesionales o dominios sospechosos. Sin embargo, los atacantes actuales utilizan herramientas mucho más avanzadas capaces de replicar páginas legítimas con gran precisión.

El uso de IA también está transformando este tipo de ataques. Hoy en día, algunos sistemas permiten generar mensajes personalizados de forma automática, adaptando el contenido según la víctima seleccionada. Esto incrementa considerablemente las probabilidades de éxito, ya que el mensaje puede parecer redactado específicamente para cada usuario.

Existen grupos organizados que venden kits completos de phishing en foros clandestinos, incluyendo páginas falsas, plantillas de correos y sistemas automatizados para recopilar credenciales robadas. Esto ha reducido enormemente la dificultad técnica necesaria para lanzar ataques. Como consecuencia, el phishing continúa siendo una de las amenazas más utilizadas dentro del panorama actual de la ciberseguridad.

3.3 Pretexting

3.3.1 Definición y Funcionamiento

El pretexting consiste en crear una identidad o situación falsa con el fin de convencer a la víctima para que proporcione información sensible o realice

determinadas acciones.El atacante suele investigar previamente a la persona objetivo para hacer más creíble la conversación. Posteriormente, puede hacerse pasar por un técnico informático, un empleado de la empresa o incluso una entidad bancaria.La clave de esta técnica reside en la capacidad de generar confianza y construir un contexto aparentemente legítimo.

3.3.2 Riesgos y Prevención

Este tipo de ataques puede resultar especialmente peligroso porque muchas veces no deja evidencias técnicas visibles. La manipulación ocurre directamente a través de conversaciones telefónicas, correos o mensajes. Para prevenir estos casos, resulta fundamental verificar siempre la identidad de quien solicita información y evitar compartir credenciales mediante canales inseguros.

Las organizaciones suelen implementar protocolos de validación y sistemas de doble comprobación para acciones críticas relacionadas con accesos o transferencias económicas.

3.4 Baiting

3.4.1 Definición y Características

El baiting consiste en atraer a la víctima mediante algún elemento aparentemente beneficioso con el objetivo de provocar una acción insegura. Uno de los ejemplos más conocidos es el uso de memorias USB infectadas abandonadas en lugares públicos para despertar la curiosidad de las personas.

También son frecuentes las descargas falsas de software, premios inexistentes o contenido pirateado que incluye malware oculto.

3.4.2 Impacto y Defensa

Aunque pueda parecer una técnica simple, continúa siendo efectiva debido a factores psicológicos como la curiosidad o el interés por obtener recompensas rápidas. Para reducir el riesgo, las empresas suelen restringir el uso de dispositivos externos y aplicar controles sobre el software autorizado. La formación de los usuarios sigue siendo esencial, especialmente en relación con los riesgos asociados al uso de dispositivos desconocidos o descargas no verificadas.

3.5 Tailgating

3.5.1 Definición

El tailgating es una técnica de ingeniería social física que consiste en acceder a zonas restringidas aprovechando la entrada de personas autorizadas.En muchos casos, el atacante aparenta ser trabajador, repartidor o técnico para no despertar sospechas.

3.5.2 Riesgos y Medidas de Seguridad

El acceso físico a instalaciones puede permitir el robo de dispositivos, la instalación de malware o el acceso directo a redes internas.Por esta razón, muchas organizaciones aplican controles de acceso estrictos, sistemas biométricos y supervisión de visitantes.

Además, la formación en seguridad física resulta tan importante como la protección informática, ya que ambos ámbitos están estrechamente relacionados.

3.6 Ingeniería Social en Redes Sociales

3.6.1 Uso de Redes Sociales

Las redes sociales hoy en día son una fuente de información extremadamente valiosa para los atacantes. Fotografías, ubicaciones, relaciones personales o datos laborales. datos que pueden utilizarse para preparar ataques mucho más personalizados.

Los atacantes analizan perfiles públicos y recopilan información mediante técnicas OSINT para identificar posibles objetivos y construir engaños más creíbles.

3.6.2 Riesgos y Prevención

La sobreexposición de información personal facilita considerablemente el trabajo de los atacantes. Muchas veces, los propios usuarios publican datos que posteriormente pueden utilizarse en campañas de phishing o suplantación de identidad. Para reducir riesgos, se recomienda limitar la información pública, revisar periódicamente las configuraciones de privacidad y desconfiar de perfiles desconocidos.

Las empresas también desarrollan políticas de uso responsable de redes sociales para evitar filtraciones involuntarias de información corporativa.

3.7 Simulación laboratorio

La implementación de este entorno permite extraer las siguientes conclusiones

- Eficacia de la Ingeniería Social: El uso de HTML5 y CSS3 demuestra que el realismo visual es el factor determinante para anular la desconfianza del usuario. La apariencia profesional de la web legitima el contenido malicioso, convirtiendo el diseño en el vector de ataque principal.
- Accesibilidad de la Infraestructura: La rapidez con la que se despliega un servidor funcional mediante Apache2 en Ubuntu evidencia que un atacante puede montar infraestructuras de phishing de forma masiva y económica, sin necesidad de recursos complejos.
- Validación del Entorno: La interacción exitosa entre la máquina servidor (Ubuntu) y la víctima (Windows) en la red aislada confirma que el escenario es técnicamente viable y funcional para realizar pruebas de penetración y estudios de concienciación en un entorno seguro.

Para una visualización detallada de la ejecución del ataque y los resultados obtenidos en el laboratorio, consulte el apartado 2 en la memoria práctica.

3.8 Conclusión

La ingeniería social demuestra que la seguridad informática no depende únicamente de herramientas tecnológicas, sino también del comportamiento humano. Los atacantes han comprendido que manipular personas suele resultar más rentable y sencillo que vulnerar sistemas complejos mediante ataques técnicos tradicionales.

La evolución de tecnologías como la inteligencia artificial y los deep fakes está incrementando el nivel de sofisticación de estos ataques, dificultando cada vez más distinguir entre comunicaciones legítimas y fraudulentas. Por eso la formación continua sólida son elementos fundamentales dentro de cualquier estrategia de ciberseguridad.

En definitiva, mientras las personas continúen formando parte de los sistemas digitales, la ingeniería social seguirá representando una de las amenazas más importantes dentro de la seguridad informática.

4. Malware

4.1.1 Definición de Malware

El término malware proviene del inglés *malicious software*, y se utiliza para describir cualquier programa o código informático creado con el propósito de causar daño, infiltrarse o aprovecharse de un sistema sin autorización. A diferencia del software legítimo, que busca aportar una función útil o resolver una necesidad, el malware actúa de manera oculta y malintencionada, buscando robar información, romper la integridad del sistema o incluso obtener beneficios económicos.

Hoy en día, el malware representa una de las amenazas más comunes y peligrosas dentro del ámbito de la ciberseguridad. Ningún entorno está completamente a salvo: puede afectar tanto a usuarios domésticos como a empresas o incluso a infraestructuras críticas. Lo más preocupante es que muchas infecciones ocurren sin que el usuario llegue a darse cuenta, gracias a las técnicas de ocultación y engaño que estos programas utilizan.

4.1.2 Evolución del Malware

La historia del malware ha evolucionado al mismo ritmo que la tecnología. En los años 80 y 90 los virus eran relativamente simples y se propagaban principalmente a través de disquetes o archivos compartidos. Esos primeros casos solían tener fines curiosos o demostrativos, más que delictivos. Sin embargo, con la expansión masiva de Internet a partir de los 2000, el panorama cambió por completo: aparecieron nuevas familias de malware, mucho más sofisticadas y con fines claramente económicos.

A medida que los sistemas de protección fueron mejorando, los atacantes empezaron a desarrollar técnicas más avanzadas como la ofuscación de código, el cifrado de comunicaciones o la evasión de antivirus. Hoy el malware no se limita a infectar ordenadores aislados, sino que forma parte de redes criminales organizadas capaces de coordinar miles de equipos al mismo tiempo. Se ha pasado del virus experimental al ataque diseñado como un negocio rentable y profesionalizado.

4.1.3 Objetivos del Malware

Los objetivos del malware dependen del tipo de ataque y del perfil del atacante, pero casi siempre giran alrededor de tres grandes ejes: robo, control y beneficio.

Por un lado, muchos ataques buscan robar información sensible, como contraseñas, datos bancarios o documentos internos de una empresa. Otros se centran en tomar el control de los equipos infectados, bien para espiarlos o para integrarlos en una red de dispositivos comprometidos (botnets) que luego se usan en otros ataques.

Finalmente, el beneficio económico se ha consolidado como uno de los principales motores: el ransomware, por ejemplo, bloquea archivos y exige un rescate para

recuperarlos, mientras que otras variantes como los troyanos bancarios buscan vaciar cuentas o interceptar transacciones. También existen casos de malware utilizado con fines políticos o de sabotaje, centrados más en dañar o desestabilizar que en ganar dinero.

4.1.4 Vectores de Infección

Un sistema puede infectarse de muchas formas diferentes. El factor humano sigue siendo el más determinante: la ingeniería social es el medio más efectivo para propagar malware. Los ataques de *phishing* o los mensajes falsos que convencen al usuario para abrir un archivo infectado o hacer clic en un enlace malicioso siguen funcionando muy bien porque se aprovechan de la confianza.

Otro vector común es la descarga de software desde fuentes no oficiales. En muchos casos, el propio usuario instala el malware pensando que está bajando una aplicación útil o “activando” un programa de pago. También siguen existiendo infecciones por dispositivos USB contaminados, sobre todo en entornos corporativos o industriales, donde la red suele estar más controlada.

Por último, la explotación de vulnerabilidades es otra vía importante. Muchos malware aprovechan fallos en sistemas o programas que no están actualizados, permitiendo que la infección ocurra de forma automática, sin que el usuario haga nada. Este tipo de fallos demuestra la importancia de mantener los equipos y las aplicaciones siempre parcheados.

4.1.5 Público objetivo

El malware no discrimina: puede afectar a cualquier usuario, pero la motivación del atacante determina a quién se dirige con más frecuencia. Los usuarios domésticos suelen ser el objetivo más común, principalmente porque suelen tener un nivel de protección y de concienciación menor. En estos casos, los ataques se centran en robar información personal, datos de tarjetas o credenciales de redes sociales.

En cambio, las empresas resultan mucho más atractivas desde el punto de vista económico. Atacar una organización que maneja datos valiosos o acceso a redes corporativas puede generar beneficios mucho mayores, especialmente mediante ransomware, espionaje industrial o robo de propiedad intelectual.

También las administraciones públicas han pasado a ser un objetivo habitual, especialmente en ataques cuyo objetivo es el sabotaje o la obtención de información sensible. Los atacantes pueden actuar por motivaciones económicas, políticas o incluso ideológicas.
En resumen, hoy en día el malware se usa tanto de forma masiva (para infectar al

mayor número de víctimas posible) como en operaciones específicas (*ataques dirigidos* o *APT*), en las que la víctima es seleccionada estratégicamente.

4.1.6 Frecuencia y relevancia actual

En la actualidad, el malware es una constante en el panorama de la ciberseguridad. Se detectan millones de nuevas muestras cada año, y muchas de ellas son variaciones mínimas de otras ya existentes, diseñadas para evadir los sistemas de detección tradicionales. El crecimiento del ransomware es un claro ejemplo de esta tendencia: se ha convertido en uno de los tipos de malware más rentables y, por tanto, más activos.

Otro aspecto importante es que una gran parte de las infecciones actuales no se debe a vulnerabilidades técnicas, sino a errores humanos: abrir un adjunto sospechoso, usar contraseñas débiles o no actualizar el sistema. Este punto es clave porque demuestra que incluso con buenas soluciones técnicas, la seguridad sigue dependiendo del comportamiento de los usuarios.

Por eso, la formación y concienciación se consideran hoy herramientas de defensa tan importantes como los antivirus o los firewalls. No basta con tener protección; hay que saber usarla correctamente.

4.1.7 Facilidad de infección

Una de las principales razones por las que el malware sigue siendo tan exitoso es la facilidad con la que puede infiltrarse. No hace falta ser un objetivo “importante” para ser víctima. Acciones cotidianas como abrir un correo, descargar un programa o visitar una página aparentemente legítima pueden bastar para comprometer un sistema.

La confianza, el descuido y la falta de actualizaciones son factores que abren la puerta a los atacantes. Además, el desarrollo de herramientas automatizadas ha reducido la barrera técnica para crear y distribuir malware. Hoy, un atacante sin grandes conocimientos puede adquirir un kit completo por Internet, listo para usar.

Esto hace que el usuario final se convierta en el eslabón más débil de la cadena de seguridad. De hecho, la mayoría de incidentes no se originan en ataques muy sofisticados, sino en descuidos básicos. La sencillez con la que puede producirse una infección refuerza la necesidad de adoptar buenas prácticas de seguridad tanto a nivel personal como organizativo.

4.1.8 Importancia de la defensa

Defenderse del malware requiere una visión integral y no depender de una sola medida. El enfoque más efectivo es el de seguridad en capas, combinando prevención, detección y respuesta.

La prevención incluye la educación de los usuarios, el uso de buenas prácticas (no descargar de fuentes no confiables, mantener todo actualizado, usar contraseñas seguras) y la reducción de la superficie de ataque mediante configuraciones seguras. La detección se apoya en antivirus tradicionales, pero también en soluciones más avanzadas como los sistemas de monitorización o los analizadores de comportamiento, capaces de reconocer actividades sospechosas incluso si el malware es nuevo.

Finalmente, la respuesta implica estar preparado para el peor escenario: disponer de copias de seguridad fiables, planes de recuperación ante incidentes y procedimientos de contención.

La realidad es que eliminar completamente el riesgo es imposible, pero una defensa bien planificada puede reducir drásticamente tanto la probabilidad como el impacto de una infección. La ciberseguridad no consiste sólo en evitar ataques, sino en saber responder cuando ocurren.

4.2 Virus Informáticos

4.2.1 Definición

Un virus informático es un tipo de malware capaz de copiarse e insertarse en otros archivos o programas legítimos, de forma similar a cómo los virus biológicos infectan organismos. Su activación depende generalmente de que el usuario ejecute el archivo infectado, por lo que la distribución del virus suele aprovechar la interacción humana.

Una vez activo, puede alterar el funcionamiento normal del sistema, modificar archivos, ralentizar procesos o incluso abrir la puerta a otros ataques más complejos. Aunque los virus clásicos han perdido protagonismo frente a amenazas más automatizadas, su estructura es la base sobre la que se desarrollaron otros tipos de malware.

4.2.2 Funcionamiento técnico

El funcionamiento de un virus se basa en insertar código malicioso dentro de archivos o programas legítimos. Al ejecutarse el archivo infectado, el sistema carga tanto el

código original como el código del virus, lo que permite su propagación.
Los virus pueden infectar diferentes tipos de archivos, aunque los más comunes son los ejecutables (.exe), los documentos con macros (como los de Office) o incluso sectores de arranque del disco.

Algunos permanecen latentes hasta que se cumple una condición específica, como una fecha concreta o la ejecución de una aplicación determinada. Otros, en cambio, comienzan a replicarse de inmediato.

En sistemas modernos, esta replicación puede provocar un alto consumo de recursos, corrupción de datos o propagación a unidades extraíbles, generando un comportamiento anómalo general en el equipo.

4.2.3 Tipos y características

Los virus pueden clasificarse según su comportamiento o la forma en que se propagan:

- Virus de archivo: se adhieren a programas ejecutables y se activan al iniciarse el programa.
- Virus de macros: se insertan en documentos de ofimática que ejecutan código (por ejemplo, en Word o Excel).
- Virus de arranque: afectan al sector de inicio de los discos y se cargan antes del sistema operativo.
- Virus residentes: permanecen en memoria después de infectar el sistema, permitiendo la infección de otros archivos sin intervención del usuario.
- Virus polimórficos o metamórficos: modifican su código cada vez que se replican, dificultando su detección mediante firmas.

La característica común de todos ellos es su capacidad de replicación y su dependencia del usuario para activarse.

4.2.4 Ejemplos reales

Algunos de los virus más conocidos en la historia de la informática son ILOVEYOU y Melissa, ambos distribuidos a través del correo electrónico a principios de los 2000. Estos virus demostraron cómo la combinación de ingeniería social y propagación automatizada podía colapsar redes enteras.

Otros casos, como CIH (Chernobyl) o Michelangelo, fueron diseñados para causar daños directos al sistema operativo, llegando incluso a inutilizar equipos.

En la actualidad, los virus puros son menos frecuentes, pero sus técnicas se siguen empleando como parte de infecciones más complejas o como componentes dentro de troyanos y gusanos.

4.2.5 Impacto y relevancia actual

Aunque los virus ya no son tan comunes como hace años, siguen siendo relevantes dentro del panorama del malware. La mayoría de ataques actuales incorporan módulos de infección inspirados en estos mecanismos clásicos, sobre todo dentro de kits de malware más avanzados o para iniciar cargas maliciosas.

Su impacto depende del contexto: en entornos domésticos, puede limitarse a una ralentización del equipo o corrupción de archivos; pero en redes corporativas mal protegidas, puede actuar como punto de entrada a operaciones mayores, especialmente si se combina con ingeniería social.

4.2.6 Medidas de prevención y defensa

La prevención sigue siendo la medida más importante. Algunas buenas prácticas clave son:

- Mantener actualizado el sistema operativo y el software.
- Utilizar antivirus o EDR con bases de firmas y análisis heurístico.
- Evitar ejecutar archivos de origen desconocido o adjuntos sospechosos.
- Analizar siempre los dispositivos externos antes de su uso.
- Configurar políticas de restricción de macros en entornos corporativos.

En entornos profesionales, se recomienda además utilizar monitorización centralizada y políticas de aislamiento de red para evitar la propagación.

4.2.7 Defensa avanzada

La defensa moderna contra virus se basa en una combinación de análisis de comportamiento, control de ejecución y protección de puntos críticos del sistema. Las herramientas actuales ya no dependen únicamente de firmas, sino que utilizan heurísticas para detectar patrones anómalos, como la replicación de archivos, la modificación masiva de documentos o el uso sospechoso de APIs del sistema.

En entornos corporativos, las macros deben estar restringidas por defecto, permitiendo solo aquellas firmadas digitalmente. Además, los sistemas EDR monitorizan continuamente los procesos activos e interceptan comportamientos típicos de virus residentes.

Otra capa importante es el sandboxing, que permite ejecutar archivos desconocidos en entornos aislados antes de permitirlos en el sistema real.

Finalmente, limitar el uso de dispositivos extraíbles y activar su escaneo automático reduce significativamente las infecciones por soportes externos.

4.2.8 Análisis crítico

A pesar de ser una de las formas más antiguas de malware, los virus siguen siendo una herramienta eficaz cuando se combina con la ingenuidad del usuario. Este tipo de amenaza demuestra que, por muy avanzado que sea el software de protección, el factor humano sigue siendo el punto débil.

Su gran limitación es que necesita acción del usuario para activarse, lo que ha hecho que los atacantes migren hacia gusanos o troyanos, más automatizados y versátiles. Sin embargo, como base de aprendizaje y análisis, los virus son fundamentales para entender la evolución del malware y las estrategias de defensa actuales.

4.3 Troyanos

4.3.1 Definición

Un troyano (del inglés *Trojan horse*) es un tipo de malware que se disfraza de programa legítimo o inofensivo para engañar al usuario y conseguir que lo ejecute voluntariamente. Su nombre hace referencia al famoso “caballo de Troya” de la mitología griega: algo que aparenta ser un regalo, pero que en realidad contiene una amenaza oculta.

A diferencia de los virus o gusanos, los troyanos no se replican por sí mismos, sino que se instalan mediante engaño o descuido. Su función principal suele ser abrir la puerta a otros ataques, establecer comunicación con el atacante y permitir el control remoto del sistema infectado.

4.3.2 Funcionamiento técnico

El funcionamiento de un troyano se basa en la ejecución de un archivo aparentemente inofensivo que, además de realizar la acción esperada por el usuario, ejecuta en paralelo un conjunto de rutinas maliciosas.

Una vez dentro del sistema, el troyano puede:

- Establecer comunicación constante con un servidor de comando y control (C2) para recibir instrucciones.
- Crear mecanismos de persistencia, como modificar el registro de Windows o programar tareas para ejecutarse en cada inicio.
- Descargar y ejecutar otros tipos de malware (ransomware, spyware, mineros, etc.).
- Otorgar al atacante acceso remoto al equipo comprometiendo la seguridad total del sistema.

En general, los troyanos actúan como un “caballo de entrada” dentro de una red: su objetivo es establecer una base de control para futuras acciones más complejas.

4.3.3 Tipos y características

Existen múltiples variantes, clasificadas según su función principal:

- Troyanos de acceso remoto (RATs): permiten al atacante controlar completamente el sistema, usar el teclado, cámara o archivos. Ejemplo: *DarkComet* o *NjRAT*.
- Troyanos bancarios: diseñados para interceptar credenciales financieras y manipular transacciones.

- Troyanos descargadores (Downloaders): su función principal es instalar otros componentes de malware.
- Troyanos espía: capturan información sensible, como contraseñas o archivos personales.
- Troyanos proxy o botnet: convierten el equipo en un nodo remoto para lanzar ataques distribuidos.

Una de sus características más peligrosas es su discreción: suelen ejecutarse en segundo plano sin generar síntomas visibles, lo que les permite mantenerse activos durante mucho tiempo sin ser detectados.

4.3.4 Ejemplos reales

Uno de los troyanos más conocidos es Zeus, diseñado para robar credenciales bancarias. Su impacto fue enorme porque combinaba técnicas de inyección de código con paneles de administración profesionales, lo que marcó un punto de inflexión en el cibercrimen.

Otro caso relevante es Emotet, que comenzó como un troyano bancario, pero evolucionó hasta convertirse en una plataforma modular que distribuía ransomware y otros tipos de malware a gran escala. De hecho, su infraestructura llegó a ser uno de los mayores ecosistemas de ciberataques entre 2017 y 2021.

Estos ejemplos muestran la capacidad de los troyanos modernos de actualizarse y transformarse según los objetivos de los atacantes, convirtiéndose en herramientas extremadamente versátiles.

4.3.5 Impacto y relevancia actual

En la actualidad, los troyanos siguen siendo una de las amenazas más relevantes debido a que son la principal vía de acceso inicial en muchos ataques complejos. A menudo no actúan solos, sino como la “primera fase” previa a la instalación de ransomware o spyware.

Su combinación de engaño, persistencia y control remoto los convierte en un arma eficaz tanto para cibercriminales como para grupos de espionaje avanzados (APT).

Según datos del INCIBE y de Europol, una gran parte de las infecciones corporativas modernas tienen su origen en un archivo troyanizado descargado por el usuario o recibido por correo electrónico.

4.3.6 Medidas de prevención y defensa

La prevención de infecciones por troyanos se basa en la seguridad proactiva:

- Descargar software únicamente desde fuentes oficiales o verificadas.
- No ejecutar archivos adjuntos ni instaladores desconocidos.
- Mantener el sistema operativo y los navegadores actualizados.
- Implementar soluciones de antivirus y EDR con análisis de comportamiento y control sobre conexiones salientes.
- En entornos empresariales, limitar los privilegios de usuario y controlar las políticas de ejecución (AppLocker, GPO, etc.).

Además, la monitorización del tráfico de red puede revelar patrones característicos, como conexiones persistentes a servidores externos o actividad fuera del horario habitual.

4.3.7 Defensa avanzada

La defensa frente a troyanos combina control técnico con reducción del riesgo humano. Las organizaciones suelen aplicar listas blancas de aplicaciones, permitiendo únicamente la ejecución de programas aprobados. Esto bloquea de forma muy efectiva la instalación accidental de troyanos.

Los firewalls de nueva generación detectan conexiones hacia servidores de comando y control desconocidos, bloqueando comunicación saliente sospechosa. Herramientas como Sysmon o Autoruns ayudan a identificar cambios en persistencia, servicios o tareas programadas que delatan la presencia de un troyano.

Los navegadores y sistemas operativos modernos integran comprobaciones de reputación que alertan si un archivo descargado tiene poca confianza. Finalmente, el control de descargas en el navegador y el filtrado DNS ayudan a evitar que los usuarios accedan a sitios que distribuyen troyanos.

4.3.8 Análisis crítico

Los troyanos siguen siendo una de las herramientas más efectivas dentro del arsenal del ciberdelincuente porque aprovechan el eslabón más débil del sistema: el usuario. No necesitan vulnerar sistemas complejos, basta con convencer a alguien de que ejecute un archivo.

Su éxito demuestra que la ciberseguridad no depende solo de la tecnología, sino también del factor humano. Mientras existan usuarios dispuestos a confiar en lo que ven, el troyano seguirá siendo una amenaza.

A su vez, esta simplicidad es también su mayor limitación: sin interacción humana, la mayoría de ellos no logran penetrar. Por eso, la defensa más eficaz no siempre es un firewall, sino la concienciación y el sentido crítico ante cualquier archivo o correo sospechoso.

4.4 Gusanos

4.4.1 Definición

Un gusano informático es un tipo de malware diseñado para replicarse y propagarse automáticamente a través de redes sin necesidad de intervención del usuario. A diferencia de los virus, los gusanos no necesitan infectar otros archivos para multiplicarse, ya que son programas autónomos capaces de copiarse a sí mismos y distribuirse mediante protocolos de red, servicios expuestos o vulnerabilidades del sistema.

Su objetivo principal suele ser la expansión rápida, aunque en muchos casos incluyen una “carga útil” que puede dañar el sistema, instalar otros tipos de malware o reclutar los equipos afectados dentro de una red de bots.

4.4.2 Funcionamiento técnico

El método más común de propagación de un gusano consiste en escanear la red en busca de dispositivos vulnerables —por ejemplo, sistemas sin parches o con contraseñas por defecto—. Una vez localizado un objetivo vulnerable, el gusano explota la debilidad y se copia en el nuevo sistema.

Este proceso es completamente automático. Mientras infecta nuevos equipos, cada uno de ellos continúa repitiendo el proceso, generando un crecimiento exponencial del número de infecciones.

Los gusanos pueden utilizar distintos medios de propagación:

- Redes locales o compartición de archivos.
- Internet, aprovechando puertos y servicios expuestos.
- Mensajería instantánea o correos electrónicos automatizados.
- Dispositivos extraíbles conectados a distintos equipos.

Su capacidad de replicarse rápidamente puede saturar redes y servidores, provocando caídas de rendimiento o denegaciones de servicio incluso sin haber ejecutado acciones destructivas.

4.4.3 Tipos y características

Según su forma de propagación, podemos distinguir varios tipos de gusanos:

- Gusanos de red: aprovechan fallos de seguridad en sistemas conectados (como vulnerabilidades en SMB o RDP).
- Gusanos de correo electrónico: se distribuyen mediante mensajes automáticos con archivos adjuntos infectados.
- Gusanos de mensajería o redes sociales: se propagan enviando enlaces o archivos a los contactos del usuario.
- Gusanos híbridos: combinan varios métodos, e incluso integran componentes de virus o troyanos.

Entre sus características más notables destacan la rapidez de propagación, el uso autónomo de los recursos del sistema y su capacidad para actuar como vector de otros tipos de malware, multiplicando el impacto del ataque inicial.

4.4.4 Ejemplos reales

Uno de los gusanos más famosos fue WannaCry (2017), que explotaba una vulnerabilidad en sistemas Windows (EternalBlue) y combinaba la propagación automática con una carga de ransomware. En pocas horas logró afectar a cientos de miles de equipos en todo el mundo, incluyendo hospitales, empresas y administraciones públicas.
Otro ejemplo es Conficker (2008), que se expandió a través de redes corporativas explotando fallos de servicio y contraseñas débiles. Llegó a infectar millones de dispositivos y permaneció activo durante años debido a la falta de actualizaciones en muchos sistemas.

Estos casos demostraron que un gusano puede producir un impacto global en cuestión de horas, especialmente cuando encuentra redes mal configuradas o vulnerabilidades sin parches.

4.4.5 Impacto y relevancia actual

Aunque los gusanos clásicos ya no son tan comunes como hace una década, su principio básico sigue muy presente en amenazas actuales. Muchos ataques modernos combinan el comportamiento de gusano con otras funciones, como el cifrado de archivos o la instalación de puertas traseras.

El impacto de un gusano no solo reside en el daño directo que pueda causar, sino también en el efecto colateral sobre los sistemas y las redes: congestión del tráfico, consumo de ancho de banda y propagación no controlada.

En entornos corporativos grandes o con sistemas críticos, un gusano puede paralizar operaciones enteras en cuestión de minutos si no existen políticas de segmentación ni sistemas de detección de intrusiones.

4.4.6 Medidas de prevención y defensa

La forma más efectiva de prevenir infecciones por gusanos es mantener los sistemas actualizados y aplicar parches de seguridad tan pronto como estén disponibles. Además, es importante:

- Limitar los servicios expuestos a la red y cerrar puertos innecesarios.
- Implementar firewalls, IDS/IPS y segmentación de red para reducir el alcance de la propagación.
- Controlar los permisos de ejecución y el uso de dispositivos externos.
- Monitorizar de forma continua la red en busca de actividad anómala o escaneos automáticos.
  En entornos empresariales, disponer de políticas de actualización centralizada y copias de seguridad desconectadas es esencial para mitigar un posible brote.

4.4.7 Defensa avanzada

Los gusanos dependen casi siempre de vulnerabilidades, por lo que las estrategias de defensa más efectivas consisten en reducir la superficie de ataque. Mantener los sistemas actualizados y aplicar parches críticos de forma inmediata es la barrera principal para impedir su propagación.

Las redes modernas también utilizan segmentación: si un gusano infecta un segmento, no puede moverse fácilmente a otros. Los IDS/IPS detectan comportamientos típicos como escaneos masivos o conexiones repetidas en puertos vulnerables.

El cierre de servicios innecesarios (como SMB o RDP cuando no son imprescindibles) elimina rutas de entrada comunes. En entornos más avanzados, los honeypots internos permiten identificar gusanos antes de que lleguen a sistemas reales.

4.4.8 Análisis crítico

Los gusanos representan uno de los ejemplos más claros de cómo una vulnerabilidad sin corregir puede convertirse en un punto de desastre a gran escala. Su efecto multiplicador hace que un pequeño fallo técnico se convierta en una crisis masiva en cuestión de horas.

Su principal fortaleza, la autonomía, también es su debilidad: al depender de vulnerabilidades conocidas, pueden ser neutralizados rápidamente si los sistemas están bien protegidos y actualizados.

En definitiva, el gusano es un recordatorio de que la seguridad preventiva y la gestión de actualizaciones no son opcionales. Basta con un solo sistema desactualizado para comprometer toda una red.

4.5 Ransomware

4.5.1 Definición

El ransomware es un tipo de malware que limita o bloquea el acceso a la información de un sistema, normalmente cifrando los archivos del usuario, y exige un pago económico —el “rescate”— a cambio de la recuperación de los datos.

Su nombre procede de *ransom* (rescate) y *software*. Este tipo de ataque combina componentes técnicos con una estrategia de extorsión digital, convirtiéndose en una de las amenazas más graves tanto para usuarios como para organizaciones.

A diferencia de otros malware que buscan pasar desapercibidos, el ransomware quiere ser visible: el atacante informa directamente a la víctima de que sus datos han sido bloqueados y le presiona psicológicamente para pagar. Esta mezcla de daño financiero y manipulación emocional explica su gran efectividad.

4.5.2 Funcionamiento técnico

El proceso de infección del ransomware suele iniciarse con la ejecución de un archivo malicioso que llega por correo, descarga o a través de una vulnerabilidad. Una vez activo, el malware realiza las siguientes acciones:

1. Exploración del sistema: localiza archivos de interés (documentos, imágenes, bases de datos…).
2. Cifrado: utiliza algoritmos como AES o RSA para bloquear los archivos, sustituyendo sus extensiones y eliminando copias de sombra.
3. Notificación: muestra un mensaje informando del secuestro e indicando el método de pago, habitualmente en criptomonedas.
4. Comunicación externa: en muchos casos, contacta con un servidor remoto para registrar la infección y enviar claves de cifrado.

Algunas variantes incluso combinan el cifrado con técnicas de movimiento lateral, extendiéndose por redes completas. Por este motivo, un solo equipo infectado puede comprometer toda una organización.

4.5.3 Tipos y características

Existen distintas variantes de ransomware según su forma de actuar:

- Ransomware de cifrado: bloquea los archivos del usuario y exige un pago por la clave de descifrado.
- Ransomware de bloqueo: impide el acceso al sistema sin cifrar los archivos (más común en ataques domésticos).

- Ransomware de doble extorsión: además de cifrar, exfiltra los datos y amenaza con publicarlos si no se paga.
- Ransomware como servicio (RaaS): modelo donde los desarrolladores alquilan el malware a terceros a cambio de una comisión del rescate.

Una característica común de todos ellos es que usan cifrado fuerte, lo que hace casi imposible recuperar los datos sin la clave. Además, la profesionalización de las bandas cibercriminales ha hecho que estos ataques sean metódicos, con atención al detalle y estructuras similares a empresas reales.

4.5.4 Ejemplos reales

El caso más conocido es WannaCry (2017), que afectó a más de 200.000 equipos en 150 países aprovechando una vulnerabilidad de Windows (EternalBlue). Su impacto fue masivo: sistemas sanitarios, empresas y organismos públicos quedaron completamente bloqueados.

Otro ejemplo relevante es Ryuk, una variante dirigida principalmente a grandes corporaciones. Su táctica consistía en infiltrar redes empresariales mediante phishing o troyanos y lanzar el cifrado en el momento más crítico, generando pérdidas millonarias.

Ambos casos ilustran cómo el ransomware ha pasado de ser un ataque oportunista a una operación planificada con objetivos claros y capacidad de devastación empresarial.

4.5.5 Impacto y relevancia actual

Hoy en día, el ransomware es una de las principales preocupaciones en ciberseguridad, especialmente en entornos empresariales y gubernamentales. Su impacto va mucho más allá del daño técnico:

- Económico: paraliza operaciones y genera pérdidas directas e indirectas.
- Legal: puede implicar responsabilidad por pérdida o filtración de datos personales (según el RGPD).
- Reputacional: daña la confianza de clientes y socios.

Además, este tipo de ataques se ha profesionalizado, con estructuras de “servicio” que permiten a cualquiera lanzar campañas sin conocimientos técnicos, lo que ha multiplicado su frecuencia.

4.5.6 Medidas de prevención y defensa

La defensa frente al ransomware se basa en anticiparse:

- Copias de seguridad periódicas: deben mantenerse desconectadas o en entornos aislados para evitar que también sean cifradas.
- Actualizaciones constantes: aplicar parches de seguridad en sistemas y servicios expuestos.
- Control de privilegios: limitar el alcance de los permisos y segmentar la red.
- Educación del usuario: evitar abrir adjuntos sospechosos o habilitar macros desconocidas.
- Herramientas de detección: soluciones con monitorización de comportamiento y respuesta automática ante cifrados masivos.

En caso de infección, la recomendación general es no pagar el rescate, ya que no garantiza la recuperación de los datos y suele reforzar la estructura delictiva detrás de estos ataques.

4.5.7 Defensa avanzada

La defensa contra ransomware se basa en una estrategia multicapa. Las copias de seguridad inmutables son fundamentales: no deben ser accesibles desde el propio sistema para evitar que el ransomware las cifre. Muchas empresas utilizan almacenamiento en la nube con versiones protegidas.

Los sistemas EDR detectan patrones de cifrado masivo en tiempo real y pueden detener procesos automáticamente. También es esencial bloquear macros o scripts no firmados y limitar el uso de PowerShell para evitar ejecuciones peligrosas.

El enfoque Zero Trust aplica restricciones incluso entre procesos internos, impidiendo que un malware se mueva lateralmente con facilidad. Finalmente, separar credenciales y evitar administradores “globales” reduce la expansión del daño cuando un equipo es comprometido.

4.5.8 Análisis crítico

El ransomware refleja de forma muy clara cómo la seguridad tecnológica y la seguridad humana están profundamente conectadas. No solo se basa en la vulnerabilidad técnica, sino en la reacción emocional de la víctima: el miedo, la urgencia y la presión son las herramientas más efectivas del atacante.

Su éxito no se debe únicamente a su sofisticación técnica, sino a su estrategia psicológica. La defensa, por tanto, tiene que contemplar tanto el plano técnico (protección, copias, actualizaciones) como el plano organizativo (protocolos de respuesta, formación y comunicación interna).

En definitiva, el ransomware no es solo un problema informático: es una prueba de madurez global en ciberseguridad. Las organizaciones que se preparan antes de que ocurra el ataque son las únicas capaces de reducir su impacto real.

4.6 Spyware

4.6.1 Definición

El spyware es un tipo de malware diseñado para recopilar información de un sistema o de su usuario sin su conocimiento ni consentimiento. Su finalidad no suele ser destruir o dañar el equipo directamente, sino observar, registrar y enviar datos al atacante.

El término viene de *spy* (espía) y refleja su naturaleza discreta: intenta permanecer activo el máximo tiempo posible sin levantar sospechas. Los datos obtenidos pueden ir desde credenciales y hábitos de navegación hasta información financiera o de actividad laboral.

El spyware representa una amenaza directa a la privacidad, y en algunos casos puede servir de punto de partida para otras acciones más graves, como fraudes, chantajes o espionaje corporativo.

4.6.2 Funcionamiento técnico

El spyware suele llegar al sistema de manera encubierta, integrado dentro de programas aparentemente inofensivos o paquetes de instalación gratuitos (*bundled software*). Una vez ejecutado, se instala de forma silenciosa y comienza su actividad en segundo plano.

Sus funciones técnicas más comunes incluyen:

- Monitoreo de actividad: registra webs visitadas, uso de aplicaciones y tiempo de conexión.
- Captura de información sensible: guarda pulsaciones de teclado (*keylogging*), historial del navegador o contenido de formularios.
- Toma de control parcial: algunos pueden activar la cámara o el micrófono para espiar al usuario.
- Transmisión de datos: la información se envía periódicamente a un servidor controlado por el atacante.

Los programas más avanzados implementan técnicas de persistencia, asegurándose de ejecutarse automáticamente al iniciar el sistema y ocultando su presencia de gestores de tareas y antivirus.

4.6.3 Tipos y características

Existen distintas variantes de spyware según los datos que recopilan o el método utilizado:

- Keyloggers: registran las pulsaciones del teclado y capturan contraseñas o mensajes.
- Trackers de navegación: recopilan el historial web y los hábitos de uso de Internet.
- Adware espía: combinan la recopilación de información con la visualización de publicidad.
- Stalkerware: diseñado para vigilar de forma encubierta a una persona, habitualmente en entornos de acoso o control.
- Spyware corporativo o gubernamental: utilizado con fines de espionaje industrial o de vigilancia estatal.

Una característica común a todos ellos es su discreción. A diferencia del ransomware o los gusanos, el spyware busca pasar desapercibido, aprovechando el tiempo para recopilar la mayor cantidad de datos posible.

4.6.4 Ejemplos reales

Uno de los casos más mediáticos fue el spyware Pegasus, desarrollado por la empresa NSO Group. Este software fue identificado en dispositivos iOS y Android y se utilizó para espiar a periodistas, políticos y activistas a través de vulnerabilidades no conocidas (*zero-day*).

En entornos más cotidianos, también se han detectado versiones de spyware distribuidas a través de aplicaciones de móvil aparentemente legítimas —juegos, linternas, fondos de pantalla— que pedían permisos excesivos y transmitían datos personales a servidores externos.

Estos ejemplos muestran una tendencia preocupante: el spyware ya no es exclusivo del cibercrimen, sino una herramienta que también puede aparecer en contextos de vigilancia institucional o social.

4.6.5 Impacto y relevancia actual

En la era del dato y la hiperconectividad, el spyware es una de las amenazas más significativas para la privacidad. No solo afecta a usuarios particulares, sino también a empresas, donde la fuga de información puede tener consecuencias legales y económicas graves.

El mayor problema del spyware es que no genera síntomas evidentes, lo que complica su detección. Puede permanecer meses o incluso años recopilando información sin ser identificado.

Actualmente, muchos de estos programas aprovechan la recopilación masiva de datos a través del marketing digital, lo que difumina la línea entre “análisis legítimo de comportamiento” y espionaje.

4.6.6 Medidas de prevención y defensa

Protegerse del spyware requiere una combinación de prevención y monitoreo activo:

- Descargar siempre software desde fuentes oficiales y verificar permisos antes de instalar.
- Evitar instaladores que incluyan múltiples programas (bundles).
- Mantener actualizado el sistema operativo y los navegadores.
- Utilizar antivirus y antimalware con capacidades de detección de comportamiento.
- En móviles, revisar regularmente los permisos de aplicaciones y la actividad de consumo de batería o red.

En entornos empresariales, además, es recomendable usar sistemas de detección de intrusiones (IDS) y auditorías periódicas para identificar tráfico sospechoso o accesos no autorizados.

4.6.7 Defensa avanzada

El spyware es difícil de detectar porque trabaja en silencio, así que la estrategia se basa en la monitorización continua. Los sistemas operativos modernos avisan cuando una aplicación accede al micrófono, cámara o ubicaciones sensibles, ayudando a identificar actividad sospechosa.

Analizar el tráfico saliente es otra medida clave, ya que el spyware siempre necesita enviar datos al exterior. Los EDR detectan llamadas sospechosas a funciones del sistema relacionadas con captura de teclado o pantalla.

En dispositivos móviles y en PC es importante revisar permisos otorgados y desinstalar aplicaciones que pidan acceso excesivo. En empresas se usan auditorías periódicas y herramientas específicas anti-spyware para detectar procesos que actúan sin permiso.

4.6.8 Análisis crítico

El spyware demuestra que la ciberseguridad no solo tiene que ver con ataques destructivos, sino también con la explotación silenciosa de la información. Su poder radica en la discreción: mientras el ransomware grita, el spyware escucha.

Esta amenaza plantea un dilema ético importante. En algunos contextos, herramientas de monitorización o telemetría legítimas pueden usarse con fines cuestionables, lo que complica establecer una frontera clara entre uso profesional y abuso.

En resumen, el spyware es una demostración de cómo la información se ha convertido en el activo más buscado del mundo digital, y cómo la privacidad —más que los sistemas— es el principal objetivo a proteger.

4.7 Rootkits

4.7.1 Definición

Un rootkit es un tipo de malware diseñado para ocultar su presencia o la de otros programas dentro de un sistema, permitiendo además el acceso privilegiado del atacante de manera persistente y sin ser detectado.

El término proviene de “root” (usuario con máximo nivel de permisos en sistemas Unix/Linux) y “kit” (conjunto de herramientas), y describe perfectamente su objetivo: mantener el control total del sistema de forma invisible.

A diferencia de otros tipos de malware que buscan causar daño inmediato, los rootkits priorizan el sigilo. Su meta es permanecer el mayor tiempo posible sin ser descubiertos, manipulando procesos, archivos o funciones del sistema operativo para fingir un entorno limpio y funcional.

4.7.2 Funcionamiento técnico

La técnica principal de un rootkit consiste en alterar el funcionamiento interno del sistema operativo. En lugar de crear procesos visibles, intercepta llamadas del sistema (syscalls) o modifica componentes críticos como el kernel o los controladores para esconder su actividad.

Por ejemplo, cuando el sistema muestra una lista de procesos activos o directorios, el rootkit puede filtrar su propia información, haciendo creer al usuario o al antivirus que no está presente.
Dependiendo del nivel al que actúe, el rootkit puede operar en diferentes capas del sistema:

- Modo usuario: se carga junto a aplicaciones normales y manipula utilidades como el Administrador de tareas o “ps”.
- Modo kernel: se integra dentro del núcleo del sistema operativo, lo que le permite un control total sobre la gestión de procesos, memoria y dispositivos.
- Modo arranque (bootkit): se ejecuta antes que el propio sistema operativo, garantizando persistencia incluso tras reinicios.

Esta capacidad de interceptar y modificar el comportamiento del sistema hace que su detección sea extremadamente compleja: a menudo, un sistema infectado aparenta estar en perfecto estado.

4.7.3 Tipos y características

Los rootkits se pueden clasificar según el nivel de acceso y su técnica de ocultación:

- Rootkits de usuario: se ejecutan en modo usuario y alteran herramientas del sistema. Son los más fáciles de eliminar.
- Rootkits de kernel: actúan a bajo nivel, modificando funciones internas del sistema operativo. Su eliminación puede requerir reinstalar el sistema.
- Bootkits: infectan el cargador de arranque o el firmware, provocando que el rootkit se ejecute incluso antes del sistema.
- Rootkits de hardware o firmware: se alojan en componentes físicos (BIOS, UEFI, tarjetas de red o de video), lo que los vuelve extremadamente persistentes.

Una característica común a todos es su intención de permanencia y sigilo, ocultándose no solo de los usuarios, sino también de los antivirus y sistemas de monitorización.

4.7.4 Ejemplos reales

Uno de los casos más famosos fue el rootkit de Sony BMG (2005), incluido accidentalmente en algunos CD de música. Su función era ocultar el software anticopia, pero terminó generando una grave vulnerabilidad que podía ser aprovechada por atacantes.

Otro ejemplo destacado es ZeroAccess, un rootkit avanzado que combinaba ocultamiento con la creación de una botnet, permitiendo a los atacantes controlar equipos infectados en todo el mundo para minado de criptomonedas o fraudes de clics.

Estos ejemplos demostraron que los rootkits no solo son herramientas de cibercrimen, sino también que pueden infiltrarse en productos comerciales o usarse con fines aparentemente legítimos, lo que los hace especialmente peligrosos.

4.7.5 Impacto y relevancia actual

Los rootkits siguen siendo una amenaza real, especialmente en ataques avanzados y dirigidos (APT). Su capacidad para mantenerse ocultos durante largos periodos los convierte en un recurso clave para el espionaje y la persistencia a largo plazo en redes corporativas.

Sin embargo, su desarrollo requiere un alto conocimiento técnico del sistema operativo, por lo que no son tan comunes como los troyanos o ransomware. Aun así, los grupos de amenaza más sofisticados continúan utilizándolos como parte de cadenas de ataque más amplias.

Además, con la expansión del hardware programable y los sistemas embebidos, el riesgo de rootkits a nivel de firmware o UEFI está aumentando, ya que pueden sobrevivir incluso a formateos completos.

4.7.6 Medidas de prevención y defensa

Evitar una infección por rootkit requiere una estrategia de seguridad avanzada, ya que no basta con un antivirus convencional. Las medidas más efectivas incluyen:

- Mantener los sistemas y controladores actualizados para reducir vulnerabilidades explotables.
- Limitar el uso de cuentas de administrador o *root* a tareas estrictamente necesarias.
- Utilizar herramientas de detección de integridad del sistema (por ejemplo, rkhunter, chkrootkit, o OSSEC en entornos Linux).
- Analizar equipos desde entornos externos (como sistemas “Live CD”) para evitar la manipulación del análisis desde el sistema infectado.
- Implementar en entornos corporativos sistemas de monitorización continua y gestión de logs (SIEM), que ayuden a detectar comportamientos anómalos.

En casos graves, la eliminación completa puede requerir reinstalar el sistema operativo e incluso verificar el firmware de los dispositivos afectados.

4.7.7 Defensa avanzada

Los rootkits son los más complejos de combatir, por lo que requieren técnicas específicas basadas en integridad del sistema. El arranque seguro (Secure Boot) evita

que se carguen componentes no firmados durante el inicio, bloqueando rootkits de arranque.

Las herramientas de verificación como rkhunter, chkrootkit u OSSEC revisan archivos del sistema y detectan modificaciones sospechosas. En muchos casos, el análisis debe realizarse desde un entorno externo (LiveCD) para evitar que el rootkit manipule la inspección.

La protección del firmware es crucial: BIOS y UEFI deben mantenerse actualizados y firmados digitalmente. Los sistemas EDR modernos utilizan hooks en el kernel para detectar modificaciones no autorizadas, bloqueando rootkits que intenten ocultarse a bajo nivel.

4.7.8 Análisis crítico

El rootkit representa una de las formas más sofisticadas de malware no por el daño directo que causa, sino por su capacidad de pasar completamente desapercibido. Su desarrollo exige gran conocimiento del funcionamiento interno del sistema operativo, lo que lo vincula normalmente a ataques avanzados o a desarrolladores con experiencia.

A diferencia de otras amenazas más evidentes, el rootkit no busca protagonismo, sino persistencia. Su existencia plantea un desafío constante a las herramientas de seguridad tradicionales, que basan su eficacia en la visibilidad del malware.

En términos de defensa, el rootkit nos recuerda un principio esencial en ciberseguridad: no se puede proteger lo que no se puede ver. Por eso, la transparencia del sistema, la monitorización continua y el control del acceso privilegiado son pilares imprescindibles para prevenir este tipo de amenazas.

4.8 Simulación Laboratorio

La fase de desarrollo y arquitectura del ejecutable permite extraer las siguientes conclusiones críticas:

- Eficacia de la Carga Útil Transparente: La técnica de combinar un instalador legítimo (Adobe Reader) con un script malicioso en segundo plano demuestra que la funcionalidad no es enemiga de la infección. El hecho de que el usuario obtenga el programa que deseaba elimina el "factor de sospecha", permitiendo que el malware opere durante periodos prolongados sin ser detectado.
- Abuso de Funcionalidades Nativas del Sistema: El uso de parámetros de compilación como *--noconsole* y comandos de sistema como *attrib +h +s* evidencia que no siempre es necesario software externo complejo para evadir la atención del usuario. El aprovechamiento de la carpeta Startup y el Registro de Windows subraya la dificultad de erradicar amenazas que utilizan mecanismos de persistencia estándar del sistema operativo.
- Resiliencia mediante la Redundancia: La implementación de un bucle de balizamiento (beaconing) y la doble persistencia aseguran la supervivencia del ataque ante reinicios o caídas temporales del servidor de control. Esto demuestra que el malware moderno está diseñado bajo un principio de autonomía, priorizando la reconexión constante sobre la ejecución única.
- Compilación Nativa como Medida de Evasión: La decisión de compilar directamente en Windows para evitar corrupciones de librerías y falsos positivos resalta la importancia de la compatibilidad binaria en el éxito de un exploit.

Para una visualización detallada de la ejecución del ataque y los resultados obtenidos en el laboratorio, consulte el apartado 3 en la memoria práctica.

4.9 Conclusión

Tras el análisis exhaustivo de las diversas tipologías de malware —desde los virus y gusanos clásicos hasta las sofisticadas arquitecturas de ransomware, spyware y rootkits— se puede concluir que el malware ha dejado de ser una simple pieza de código dañino para convertirse en un ecosistema de explotación integral.

De este estudio se desprenden cuatro conclusiones estratégicas fundamentales:

- La especialización y modularidad del ataque: El malware moderno ya no cumple una sola función. El atacante utiliza una "cadena de infección" donde cada tipo de malware cumple un rol: el Troyano garantiza el acceso inicial, el Gusano expande la presencia en la red, el Spyware

exfiltra información valiosa en silencio y, finalmente, el Ransomware ejecuta la extorsión económica. Esta modularidad permite al cibercrimen maximizar el beneficio de una sola intrusión.

- El sigilo como activo estratégico: Mientras que el Ransomware utiliza la visibilidad como arma de presión, el Spyware y los Rootkits demuestran que, en ciberseguridad, lo que no se ve es a menudo lo más peligroso. La capacidad de un Rootkit para subvertir el propio sistema operativo y volverse invisible a los ojos del administrador redefine el concepto de "compromiso total", obligando a las defensas a evolucionar hacia la verificación de integridad de bajo nivel (firmware y kernel).
- La erosión de la privacidad frente a la monetización del dato: El auge del Spyware subraya que la información personal y corporativa es el activo más codiciado. La transición de ataques destructivos a ataques de vigilancia silenciosa refleja un cambio de paradigma: el objetivo ya no es romper el sistema, sino convertirlo en una fuente continua de datos para el espionaje industrial, político o financiero.
- La insuficiencia de la defensa estática: El análisis técnico de estas amenazas demuestra que el antivirus tradicional basado en firmas es insuficiente. Ante amenazas polimórficas y técnicas de persistencia avanzada, la defensa debe ser proactiva y multicapa. Conceptos como el *Sandboxing*, el análisis heurístico, el *EDR* (Endpoint Detection and Response) y el *Secure Boot* son hoy requisitos mínimos para garantizar una protección mínima aceptable.

En resumen, el malware es el reflejo de la carrera armamentista digital. La conclusión final es que no existe el sistema invulnerable, sino el sistema resiliente. Una organización madura en ciberseguridad es aquella que no solo invierte en evitar la infección, sino que diseña su infraestructura bajo la premisa de que el malware eventualmente entrará, enfocando sus esfuerzos en la detección temprana, la segmentación para evitar el movimiento lateral y una capacidad de recuperación rápida y garantizada.

5. Ataques de Red

5.1.1 Definición de ataques de red

Los ataques de red constituyen un conjunto de técnicas maliciosas orientadas a comprometer la seguridad de los sistemas informáticos mediante la explotación de sus comunicaciones. A diferencia de otros tipos de amenazas que actúan directamente sobre el sistema, estos ataques se centran en los datos en tránsito, es decir, en la información que circula entre dispositivos a través de la red.

Desde un punto de vista técnico, su objetivo es vulnerar alguno de los tres pilares fundamentales de la seguridad de la información: la confidencialidad, la integridad y la disponibilidad. Para ello, los atacantes aprovechan debilidades en protocolos de red, configuraciones incorrectas o la ausencia de mecanismos de protección como el cifrado o la autenticación.

En este contexto, los ataques que se analizan en este trabajo —denegación de servicio (DoS/DDoS), sniffing, spoofing, Man-in-the-Middle y session hijacking— representan distintos enfoques para explotar la red, ya sea interrumpiendo el servicio, interceptando comunicaciones o suplantando identidades digitales.

5.1.2 Evolución de los ataques de red

La evolución de los ataques de red está directamente relacionada con el desarrollo de Internet y la creciente complejidad de las infraestructuras tecnológicas. En sus primeras etapas, cuando las comunicaciones carecían de mecanismos de seguridad robustos, era relativamente sencillo interceptar tráfico o analizar paquetes mediante técnicas básicas de escucha. Esto favoreció la aparición de ataques pasivos como el sniffing.

Con el tiempo, los atacantes comenzaron a desarrollar técnicas activas capaces de modificar o manipular la comunicación. Así surgieron ataques como el spoofing, que permite falsificar la identidad de un dispositivo dentro de la red, o el Man-in-the-Middle, donde el atacante se sitúa entre dos extremos de comunicación sin que estos lo detecten.

Posteriormente, con el aumento del tráfico en Internet y la dependencia de servicios online, los ataques de denegación de servicio cobraron especial relevancia. Estos evolucionaron hacia modelos distribuidos (DDoS), donde múltiples sistemas comprometidos actúan coordinadamente para saturar un objetivo.

En la actualidad, estos ataques no suelen ejecutarse de forma aislada, sino que se combinan en cadenas complejas. Por ejemplo, un ataque de spoofing puede facilitar un escenario de Man-in-the-Middle, que a su vez permite el secuestro de sesión. Esta

integración de técnicas refleja un alto grado de sofisticación en el panorama actual de la ciberseguridad.

5.1.3 Objetivos de los ataques de red

Los ataques de red responden principalmente a la intención de vulnerar los principios básicos de seguridad. En primer lugar, la disponibilidad se ve comprometida mediante ataques de denegación de servicio, cuyo objetivo es impedir que los usuarios legítimos accedan a un recurso o servicio. Este tipo de ataques no busca necesariamente acceder a la información, sino bloquear su acceso mediante la saturación de recursos.

Por otro lado, la confidencialidad se ve afectada en ataques como el sniffing o el Man-in-the-Middle, donde el atacante intercepta la comunicación para obtener información sensible, como credenciales, datos personales o información corporativa. Este tipo de ataques es especialmente efectivo en entornos donde no se emplea cifrado adecuado.

Finalmente, la integridad y la autenticidad se ven comprometidas en ataques como el spoofing o el session hijacking. En estos casos, el atacante no solo accede a la información, sino que puede modificarla o hacerse pasar por un usuario legítimo, alterando el comportamiento normal de la comunicación.

En conjunto, estos objetivos suelen estar alineados con finalidades más amplias como el fraude económico, el espionaje o el sabotaje de sistemas.

5.1.4 Vectores de ataque

Los vectores de ataque en el ámbito de red hacen referencia a los puntos de entrada o condiciones que permiten al atacante llevar a cabo sus acciones. En muchos casos, estos vectores están relacionados con la falta de seguridad en las comunicaciones o con configuraciones inadecuadas.

Uno de los principales factores es el uso de redes inseguras, especialmente redes WiFi abiertas o mal configuradas, que facilitan la interceptación de tráfico y la ejecución de ataques como el sniffing o el Man-in-the-Middle. En estos entornos, el atacante puede acceder fácilmente a la comunicación sin necesidad de comprometer directamente los dispositivos.

Otro vector importante es la debilidad inherente de ciertos protocolos de red. Protocolos como ARP, DNS o HTTP fueron diseñados sin mecanismos de autenticación o cifrado, lo que los hace vulnerables a ataques de suplantación o manipulación. Esta falta de seguridad estructural sigue siendo explotada en la actualidad.

Además, la ausencia de cifrado en las comunicaciones permite que los datos puedan ser capturados y analizados con relativa facilidad. Esto resulta especialmente crítico en aplicaciones web que no utilizan HTTPS o en sistemas que no implementan correctamente la protección de sesiones.

Por último, la exposición de servicios a Internet también constituye un vector relevante, especialmente en el caso de ataques de denegación de servicio, donde el objetivo es saturar dichos servicios mediante un volumen masivo de tráfico.

5.1.5 Público objetivo

Los ataques de red pueden dirigirse a una amplia variedad de objetivos, aunque el tipo de víctima suele influir en la técnica empleada. Los usuarios domésticos representan un objetivo frecuente debido a su menor nivel de protección, especialmente en redes WiFi o dispositivos mal configurados. En estos casos, los ataques suelen centrarse en la interceptación de tráfico o el robo de credenciales.

En el ámbito empresarial, el interés radica principalmente en el acceso a redes internas y la obtención de información sensible. Las empresas suelen ser objetivo de ataques más elaborados, que combinan varias técnicas de red para lograr persistencia o acceso privilegiado.

Por otro lado, los servicios online y plataformas digitales son objetivos habituales de ataques de denegación de servicio, ya que su disponibilidad es crítica para su funcionamiento. En estos casos, el impacto no solo es técnico, sino también económico y reputacional.

Finalmente, las infraestructuras críticas representan uno de los objetivos más sensibles, ya que un ataque exitoso puede tener consecuencias graves en sectores como la sanidad, la energía o el transporte.

5.1.6 Frecuencia y relevancia actual

En el contexto actual, los ataques de red son constantes y forman parte del tráfico habitual en Internet. Cualquier sistema conectado está expuesto a intentos continuos de escaneo, conexiones no autorizadas y tráfico malicioso automatizado.

La creciente digitalización, junto con el aumento de dispositivos conectados y el uso de redes distribuidas, ha incrementado significativamente la superficie de ataque. Esto ha provocado que técnicas clásicas como el sniffing o el Man-in-the-Middle sigan siendo relevantes, especialmente cuando no se aplican medidas de seguridad adecuadas.

Además, muchos de estos ataques no generan un impacto inmediato visible, lo que dificulta su detección y permite que el atacante mantenga su actividad durante largos periodos sin ser descubierto.

5.1.7 Facilidad de explotación

Uno de los aspectos más preocupantes de los ataques de red es la relativa facilidad con la que pueden llevarse a cabo. En la actualidad, existen numerosas herramientas que permiten analizar tráfico, interceptar comunicaciones o simular ataques de forma

automatizada. Esto reduce considerablemente la barrera técnica necesaria para ejecutar este tipo de acciones.

A esta situación se suma la persistencia de errores comunes en la configuración de redes y sistemas, como el uso de protocolos inseguros, la falta de cifrado o una gestión deficiente de sesiones. Estos factores hacen que incluso ataques básicos puedan tener una alta tasa de éxito.

Como resultado, el usuario y el administrador de sistemas se convierten en elementos clave dentro de la seguridad, ya que muchas vulnerabilidades no se deben a fallos complejos, sino a malas prácticas o descuidos.

5.1.8 Importancia de la defensa

La protección frente a ataques de red requiere un enfoque integral que combine múltiples medidas de seguridad. No existe una solución única capaz de prevenir todos los tipos de ataque, por lo que es necesario aplicar una estrategia basada en capas.

En primer lugar, la prevención se basa en asegurar las comunicaciones mediante el uso de cifrado, la correcta configuración de servicios y la segmentación de la red. En segundo lugar, la detección implica la monitorización continua del tráfico para identificar comportamientos anómalos o patrones de ataque. Finalmente, la capacidad de respuesta permite contener y mitigar el impacto de un incidente una vez que se ha producido.

Además, en el contexto actual cobra especial relevancia el modelo de seguridad “Zero Trust”, que parte de la premisa de no confiar en ningún elemento de la red por defecto, incluso si se encuentra dentro del perímetro interno. Este enfoque resulta especialmente eficaz frente a ataques que explotan la propia comunicación entre sistemas.

5.2 Denegación de servicio (DoS / DDoS)

5.2.1 Definición

Un ataque de denegación de servicio (DoS, Denial of Service) es una técnica cuyo objetivo es interrumpir o degradar el funcionamiento de un sistema, red o servicio, impidiendo que los usuarios legítimos puedan acceder a él. Este tipo de ataque no busca acceder a la información, sino afectar directamente a la disponibilidad, uno de los pilares fundamentales de la seguridad.

Cuando este ataque se realiza desde múltiples dispositivos de forma coordinada, se denomina ataque distribuido de denegación de servicio (DDoS, Distributed Denial of Service). En este caso, el atacante utiliza una red de sistemas comprometidos conocida como botnet para generar un volumen masivo de tráfico hacia el objetivo, dificultando su mitigación.

5.2.2 Funcionamiento técnico

El funcionamiento de un ataque DoS se basa en la saturación de los recursos del sistema objetivo. Estos recursos pueden incluir el ancho de banda de red, la capacidad de procesamiento del servidor o el número de conexiones simultáneas que puede gestionar.

En un escenario típico, el atacante envía una gran cantidad de solicitudes al servidor con el objetivo de consumir sus recursos. Si el volumen de tráfico supera la capacidad del sistema, este deja de responder correctamente a las peticiones legítimas.

En ataques DDoS, este proceso se amplifica mediante el uso de múltiples dispositivos distribuidos geográficamente. Estos dispositivos, que pueden ser ordenadores, servidores o incluso dispositivos IoT, han sido previamente comprometidos y controlados por el atacante. La coordinación de estos nodos permite generar un tráfico mucho más difícil de filtrar o bloquear.

Existen distintos mecanismos técnicos para llevar a cabo estos ataques, entre los que destacan:

- Saturación de ancho de banda mediante tráfico masivo
- Consumo de recursos del servidor (CPU, memoria)
- Explotación de protocolos para amplificar el tráfico (amplification attacks)
- Ataques a nivel de aplicación que simulan tráfico legítimo

5.2.3 Tipos y características

Los ataques DoS/DDoS pueden clasificarse según la capa del modelo de red a la que afectan y la técnica utilizada.

Ataques volumétricos:
Se centran en saturar el ancho de banda de la red mediante grandes volúmenes de tráfico. Su objetivo es colapsar la infraestructura de red antes de que el tráfico llegue al servidor.

Ataques de protocolo:
Explotan debilidades en protocolos como TCP, enviando solicitudes maliciosas o incompletas para consumir recursos del sistema. Un ejemplo típico es el SYN flood, donde se inician conexiones que nunca se completan.

Ataques a nivel de aplicación:
Se dirigen a servicios específicos como servidores web. Simulan tráfico legítimo (por ejemplo, múltiples peticiones HTTP), lo que dificulta su detección y mitigación.

Una característica clave de los ataques modernos es su capacidad de adaptación, combinando diferentes técnicas para aumentar su efectividad y evadir sistemas de defensa.

5.2.4 Ejemplos reales

Uno de los ataques DDoS más conocidos ocurrió en 2016 contra el proveedor de servicios DNS Dyn. Este ataque utilizó la botnet Mirai, compuesta principalmente por dispositivos IoT vulnerables, y logró afectar a servicios como Twitter, Netflix o GitHub, provocando interrupciones masivas en Internet.

Otro caso relevante es el ataque contra GitHub en 2018, que alcanzó un pico de 1.35 Tbps mediante una técnica de amplificación basada en servidores memcached. Este ataque destacó por su intensidad y por el uso de infraestructuras legítimas para amplificar el tráfico.

Estos ejemplos evidencian la capacidad de los ataques DDoS para generar impactos a gran escala utilizando recursos relativamente accesibles.

5.2.5 Impacto y relevancia actual

En la actualidad, los ataques de denegación de servicio representan una de las amenazas más frecuentes para servicios online. Su impacto puede ser significativo, especialmente en entornos donde la disponibilidad es crítica.

A nivel económico, estos ataques pueden provocar pérdidas debido a la interrupción de servicios, la caída de plataformas de comercio electrónico o la pérdida de confianza por parte de los usuarios. Además, también pueden utilizarse como parte de estrategias más amplias, por ejemplo, como distracción mientras se ejecutan otros ataques.

El crecimiento de las botnets y la disponibilidad de servicios de DDoS como servicio (DDoS-as-a-Service) han contribuido a la proliferación de este tipo de ataques, reduciendo la barrera de entrada para los atacantes.

5.2.6 Medidas de prevención y defensa

La defensa frente a ataques DoS/DDoS requiere una combinación de medidas técnicas y organizativas.

Entre las principales estrategias se encuentran:

- Uso de sistemas de mitigación DDoS basados en filtrado y análisis de tráfico
- Implementación de balanceadores de carga para distribuir las peticiones
- Limitación de tasas (rate limiting) para controlar el número de solicitudes
- Uso de redes de distribución de contenido (CDN) para absorber tráfico
- Configuración de firewalls y sistemas de detección de intrusiones

Además, es fundamental contar con proveedores de servicios preparados para gestionar grandes volúmenes de tráfico, así como planes de contingencia ante incidentes.

5.2.7 Defensa avanzada

Las soluciones avanzadas frente a DDoS incluyen sistemas capaces de analizar el tráfico en tiempo real y distinguir entre tráfico legítimo y malicioso mediante técnicas de aprendizaje automático y análisis de comportamiento.

El uso de infraestructuras distribuidas permite absorber el impacto del ataque, mientras que tecnologías como Anycast ayudan a repartir el tráfico entre múltiples nodos geográficamente dispersos.

Asimismo, los proveedores de servicios en la nube ofrecen mecanismos de mitigación automática que detectan patrones de ataque y aplican contramedidas sin intervención manual.

5.2.8 Análisis crítico

Los ataques DoS/DDoS ponen de manifiesto una debilidad fundamental de los sistemas conectados: su dependencia de la disponibilidad continua. A diferencia de otros ataques más sofisticados, no requieren necesariamente vulnerabilidades complejas, sino simplemente una capacidad suficiente para saturar recursos.

Su eficacia radica en la asimetría entre atacante y defensor: generar tráfico malicioso puede ser relativamente sencillo y barato, mientras que defenderse requiere infraestructuras robustas y costosas.

Además, la proliferación de dispositivos IoT inseguros ha ampliado significativamente la capacidad de los atacantes para lanzar ataques distribuidos. Esto plantea un desafío estructural en la seguridad de Internet, donde la protección no depende únicamente del objetivo, sino del estado global de la red.

5.3 Sniffing

5.3.1 Definición

El sniffing es una técnica de interceptación de tráfico de red que consiste en capturar y analizar los paquetes de datos que circulan entre dispositivos conectados a una red. Su objetivo principal es obtener información transmitida a través de la comunicación, como credenciales, sesiones, datos personales o información corporativa sensible.

Desde un punto de vista técnico, el sniffing puede realizarse de forma legítima con fines de administración y diagnóstico de redes, aunque también puede utilizarse de forma maliciosa para espiar comunicaciones sin autorización.

Este tipo de ataque afecta principalmente a la confidencialidad de la información, ya que permite al atacante acceder a datos que deberían permanecer privados durante la transmisión.

5.3.2 Funcionamiento técnico

El funcionamiento del sniffing se basa en la captura de paquetes de red mediante interfaces configuradas en modo promiscuo. En este modo, la tarjeta de red deja de filtrar únicamente el tráfico destinado al propio dispositivo y comienza a capturar todos los paquetes que circulan por la red local.

En redes tradicionales basadas en hubs, esta técnica permitía interceptar fácilmente todo el tráfico compartido. Sin embargo, en redes modernas con switches, los atacantes suelen combinar el sniffing con técnicas adicionales como ARP spoofing o ataques Man-in-the-Middle para redirigir el tráfico hacia el sistema atacante.

Una vez capturados los paquetes, el atacante utiliza herramientas de análisis capaces de reconstruir sesiones, extraer credenciales o interpretar protocolos de comunicación.

Los datos más sensibles suelen obtenerse en comunicaciones sin cifrar, especialmente mediante protocolos como HTTP, FTP o Telnet.

5.3.3 Tipos y características

El sniffing puede clasificarse según la forma en la que se captura el tráfico.

Sniffing pasivo

Consiste en escuchar el tráfico de red sin modificar la comunicación. Es más común en redes compartidas o mal segmentadas y resulta difícil de detectar, ya que no altera el comportamiento de la red.

Sniffing activo

Implica manipular el tráfico o la infraestructura de red para redirigir los paquetes hacia el atacante. Suele utilizar técnicas como:

- ARP spoofing
- MAC flooding
- DHCP spoofing

Este tipo de sniffing es habitual en redes conmutadas modernas.

Características principales

- Captura silenciosa de tráfico
- Obtención de credenciales y sesiones
- Dependencia de comunicaciones no cifradas
- Dificultad de detección en ataques pasivos

5.3.4 Ejemplos reales

Uno de los casos más conocidos relacionados con sniffing ocurrió en redes WiFi públicas donde los usuarios accedían a servicios sin cifrado HTTPS. En estos entornos, atacantes conectados a la misma red podían capturar credenciales, cookies de sesión y tráfico sensible utilizando herramientas de análisis de paquetes.

Otro ejemplo relevante fue la utilización de herramientas como Firesheep en 2010, una extensión para navegadores que permitía secuestrar sesiones web mediante la captura de cookies transmitidas sin cifrado. Este caso evidenció la importancia de implementar HTTPS de forma generalizada en servicios web.

En entornos corporativos, el sniffing también ha sido utilizado como parte de ataques más complejos orientados al espionaje interno, permitiendo recopilar información sobre credenciales, servidores y estructura de red antes de ejecutar otras fases del ataque.

5.3.5 Impacto y relevancia actual

Aunque actualmente muchas comunicaciones utilizan cifrado mediante HTTPS o VPN, el sniffing continúa siendo una técnica relevante dentro de la ciberseguridad ofensiva. Su efectividad depende principalmente del nivel de protección aplicado a la red y a las comunicaciones.

En redes inseguras o mal configuradas, el sniffing puede permitir el robo de credenciales, información financiera, datos corporativos o sesiones autenticadas. Además, en ataques dirigidos, esta técnica suele utilizarse como fase previa de reconocimiento para recopilar información sobre la infraestructura objetivo.

La proliferación de dispositivos IoT y redes inalámbricas también ha incrementado la superficie de exposición, especialmente en entornos domésticos donde las configuraciones de seguridad suelen ser deficientes.

Actualmente, el sniffing sigue siendo una técnica ampliamente utilizada tanto en auditorías de seguridad legítimas como en actividades maliciosas relacionadas con espionaje y robo de información.

5.3.6 Medidas de prevención y defensa

La principal medida de protección frente al sniffing es el uso de comunicaciones cifradas. Protocolos como HTTPS, SSH o VPN impiden que un atacante pueda interpretar fácilmente los datos capturados durante la transmisión.

Entre las medidas defensivas más importantes destacan:

- Uso obligatorio de HTTPS en aplicaciones web
- Implementación de redes privadas virtuales (VPN)
- Segmentación de red y aislamiento de dispositivos

- Configuración segura de redes WiFi mediante WPA2 o WPA3
- Desactivación de protocolos inseguros como Telnet o FTP
- Monitorización del tráfico de red
- Uso de sistemas IDS/IPS para detectar comportamientos anómalos

5.3.7 Defensa avanzada

Las estrategias avanzadas frente al sniffing incluyen mecanismos de cifrado extremo a extremo y arquitecturas de red basadas en el modelo Zero Trust.

Las organizaciones modernas implementan sistemas capaces de detectar anomalías relacionadas con ataques de interceptación, incluyendo:

- Detección de ARP spoofing
- Monitorización de cambios en tablas MAC
- Análisis de tráfico sospechoso
- Network Access Control (NAC)

Asimismo, las soluciones EDR y NDR (Network Detection and Response) permiten correlacionar eventos de red para identificar actividades de captura o manipulación de tráfico en tiempo real.

El uso de certificados digitales, autenticación mutua y túneles cifrados también reduce significativamente la efectividad de este tipo de ataques.

5.3.8 Análisis crítico

El sniffing demuestra cómo la seguridad de una comunicación no depende únicamente de los sistemas finales, sino también del canal por el que circula la información. Aunque interceptar tráfico puede parecer una técnica relativamente simple, su impacto puede ser muy elevado cuando las comunicaciones no están correctamente protegidas.

La evolución hacia protocolos cifrados ha reducido considerablemente la efectividad del sniffing tradicional. Sin embargo, los atacantes continúan desarrollando técnicas complementarias para superar estas limitaciones, especialmente mediante ataques Man-in-the-Middle o suplantación de red.

Además, muchos usuarios siguen utilizando redes públicas o dispositivos mal configurados, lo que mantiene vigente esta amenaza en la actualidad.

Desde una perspectiva defensiva, el sniffing pone de manifiesto la importancia del cifrado como elemento esencial de la seguridad moderna, así como la necesidad de adoptar una estrategia integral de protección de comunicaciones.

5.4 Spoofing

5.4.1 Definición

El spoofing es una técnica de suplantación utilizada en ciberseguridad mediante la cual un atacante falsifica la identidad de un dispositivo, usuario o servicio con el objetivo de engañar a otros sistemas dentro de una red.

Esta técnica puede aplicarse sobre distintos elementos de comunicación, como direcciones IP, direcciones MAC, servidores DNS, correos electrónicos o páginas web. El objetivo principal suele ser obtener acceso no autorizado, interceptar información, redirigir tráfico o facilitar otros ataques más complejos.

Desde el punto de vista de la seguridad, el spoofing compromete principalmente la autenticidad y la integridad de las comunicaciones, ya que el receptor confía en una identidad falsificada creyendo que pertenece a una fuente legítima.

5.4.2 Funcionamiento técnico

El funcionamiento del spoofing depende del tipo de identidad que se desea falsificar. En términos generales, el atacante modifica determinados parámetros de los paquetes de red o de la comunicación para aparentar ser otro sistema legítimo.

Uno de los casos más comunes es el IP spoofing, donde se falsifica la dirección IP de origen de los paquetes enviados. Esto permite ocultar la identidad real del atacante o hacerse pasar por otro dispositivo dentro de la red.

Otro ejemplo frecuente es el ARP spoofing, utilizado en redes locales. En este caso, el atacante envía respuestas ARP falsas para asociar su dirección MAC con la dirección IP de otro dispositivo, normalmente el router. Como consecuencia, el tráfico de la víctima pasa a través del sistema atacante.

También existen variantes como el DNS spoofing, donde se manipulan respuestas DNS para redirigir al usuario hacia sitios fraudulentos, o el email spoofing, utilizado en campañas de phishing para aparentar que un correo proviene de una fuente legítima.

Estas técnicas suelen emplearse como fase previa para ataques más avanzados, como Man-in-the-Middle, robo de credenciales o distribución de malware.

5.4.3 Tipos y características

El spoofing puede clasificarse según el elemento que se suplanta.

IP Spoofing

Consiste en falsificar la dirección IP de origen de un paquete de red. Es habitual en ataques DDoS y técnicas de ocultación.

ARP Spoofing

Manipula las tablas ARP de los dispositivos de una red local para redirigir el tráfico hacia el atacante.

DNS Spoofing

Permite modificar respuestas DNS para redirigir a la víctima hacia páginas fraudulentas o servidores maliciosos.

Email Spoofing

Consiste en falsificar el remitente de un correo electrónico para engañar al receptor.

MAC Spoofing

Se modifica la dirección MAC de un dispositivo para hacerse pasar por otro equipo autorizado dentro de la red.

Características principales

- Suplantación de identidad digital
- Manipulación de comunicaciones
- Facilita ataques posteriores
- Dificultad de detección en ciertos entornos
- Aprovechamiento de protocolos inseguros

5.4.4 Ejemplos reales

Uno de los usos más frecuentes del spoofing se produce en campañas de phishing mediante email spoofing. Los atacantes falsifican direcciones de correo de entidades legítimas, como bancos o empresas tecnológicas, para engañar a los usuarios y obtener credenciales o información sensible.

Otro caso habitual es el ARP spoofing en redes WiFi públicas. En este escenario, el atacante se sitúa entre la víctima y el router para interceptar el tráfico y capturar datos transmitidos sin cifrado.

También existen ataques DNS spoofing dirigidos contra infraestructuras corporativas, donde los usuarios son redirigidos a páginas falsas diseñadas para robar credenciales o distribuir malware.

En ataques DDoS, el IP spoofing se utiliza frecuentemente para ocultar el origen del tráfico malicioso y dificultar las tareas de mitigación.

5.4.5 Impacto y relevancia actual

El spoofing continúa siendo una técnica muy utilizada debido a que muchos protocolos de red fueron diseñados originalmente sin mecanismos robustos de autenticación.

Su impacto puede ser elevado, especialmente cuando se utiliza para facilitar otros ataques más complejos. La capacidad de suplantar identidades permite al atacante obtener la confianza de usuarios y sistemas, aumentando significativamente la probabilidad de éxito.

En la actualidad, esta técnica sigue presente tanto en ataques dirigidos como en campañas masivas automatizadas. Además, el crecimiento de redes inalámbricas, servicios cloud y dispositivos IoT ha ampliado las posibilidades de explotación.

El spoofing representa una amenaza especialmente crítica en entornos donde la autenticación y la validación de comunicaciones no están correctamente implementadas.

5.4.6 Medidas de prevención y defensa

La prevención frente al spoofing requiere combinar mecanismos de autenticación, cifrado y validación de tráfico.

Entre las principales medidas defensivas destacan:

- Uso de protocolos seguros con autenticación
- Implementación de HTTPS, SSH y VPN
- Configuración de filtros anti-spoofing en routers y firewalls
- Uso de DNSSEC para proteger resoluciones DNS
- Implementación de SPF, DKIM y DMARC en correo electrónico
- Segmentación de red
- Monitorización de tráfico anómalo
- Protección ARP dinámica en switches gestionables

Además, la concienciación de los usuarios resulta fundamental para reducir el impacto de ataques basados en su plantación.

5.4.7 Defensa avanzada

Las soluciones avanzadas frente al spoofing incluyen sistemas capaces de validar automáticamente la autenticidad de dispositivos y comunicaciones dentro de la red.

Las infraestructuras modernas utilizan mecanismos como:

- Network Access Control (NAC)
- Autenticación mutua mediante certificados digitales
- Zero Trust Network Architecture
- Sistemas IDS/IPS especializados
- Inteligencia artificial para detección de anomalías

En redes empresariales, la correlación de eventos y el análisis de comportamiento permiten detectar intentos de suplantación incluso cuando el atacante utiliza técnicas sofisticadas.

Asimismo, el uso de cifrado extremo a extremo reduce considerablemente la utilidad del tráfico interceptado mediante spoofing.

5.4.8 Análisis crítico

El spoofing evidencia una de las debilidades históricas de Internet: la confianza implícita entre sistemas y protocolos diseñados originalmente sin considerar amenazas modernas.

Aunque actualmente existen numerosos mecanismos de protección, la compatibilidad con tecnologías antiguas y las malas configuraciones continúan facilitando este tipo de ataques.

Además, el spoofing rara vez actúa de forma aislada. Su verdadero potencial aparece cuando se combina con técnicas como sniffing, Man-in-the-Middle o phishing, permitiendo ataques mucho más complejos y difíciles de detectar.

Desde una perspectiva defensiva, esta técnica pone de manifiesto la necesidad de implementar modelos de seguridad basados en validación continua y desconfianza por defecto, especialmente en infraestructuras modernas conectadas a Internet.

5.5 Man-in-the-Middle (MitM)

5.5.1 Definición

Un ataque Man-in-the-Middle (MitM) es una técnica de interceptación en la que un atacante se sitúa de forma encubierta entre dos sistemas que se están comunicando, con el objetivo de capturar, modificar o manipular la información intercambiada sin que las víctimas lo detecten.

En este tipo de ataque, ambas partes creen estar comunicándose directamente entre sí, cuando en realidad toda la información pasa previamente por el sistema controlado por el atacante.

Los ataques MitM comprometen principalmente la confidencialidad, integridad y autenticidad de las comunicaciones, ya que permiten tanto el espionaje como la alteración del contenido transmitido.

5.5.2 Funcionamiento técnico

El funcionamiento de un ataque Man-in-the-Middle se basa en interceptar el flujo de comunicación entre dos dispositivos o servicios.

Para lograrlo, el atacante debe posicionarse en medio de la comunicación mediante distintas técnicas, como:

- ARP spoofing
- DNS spoofing
- Rogue Access Points
- IP spoofing
- Secuestro de sesiones

Una vez situado entre ambas partes, el atacante puede actuar de diferentes formas:

- Escuchar el tráfico sin modificarlo
- Capturar credenciales o cookies
- Alterar paquetes transmitidos
- Redirigir a páginas falsas
- Inyectar contenido malicioso

En redes locales, uno de los métodos más utilizados es el ARP spoofing, donde el atacante engaña tanto a la víctima como al router para que envíen el tráfico a través de su sistema.

En entornos web, el atacante puede intentar degradar conexiones HTTPS o utilizar certificados falsos para interceptar comunicaciones cifradas.

5.5.3 Tipos y características

Los ataques MitM pueden clasificarse según la técnica utilizada para interceptar la comunicación.

ARP MitM

Se realiza en redes locales mediante manipulación de tablas ARP para redirigir el tráfico.

DNS MitM

Consiste en alterar respuestas DNS para dirigir a la víctima hacia servidores controlados por el atacante.

HTTPS MitM

Busca interceptar comunicaciones cifradas mediante certificados falsos o ataques SSL stripping.

WiFi Evil Twin

El atacante crea un punto de acceso falso con apariencia legítima para que las víctimas se conecten a él.

Session Hijacking

Consiste en capturar sesiones autenticadas para asumir la identidad del usuario.

Características principales

- Interceptación silenciosa de comunicaciones
- Posibilidad de manipular tráfico en tiempo real
- Robo de credenciales y sesiones
- Dependencia de redes inseguras o usuarios desprevenidos
- Alta efectividad combinada con ingeniería social

5.5.4 Ejemplos reales

Uno de los escenarios más habituales de ataques MitM ocurre en redes WiFi públicas. Los atacantes crean puntos de acceso falsos o utilizan ARP spoofing para interceptar el tráfico de los usuarios conectados.

Otro ejemplo conocido es el uso de SSL stripping, técnica que fuerza conexiones HTTP inseguras para evitar el cifrado HTTPS y permitir la lectura del tráfico.

También se han documentado ataques MitM dirigidos contra infraestructuras corporativas mediante manipulación DNS o certificados comprometidos.

En algunos casos, incluso aplicaciones móviles y redes IoT han sido vulnerables a este tipo de ataques debido a validaciones incorrectas de certificados o comunicaciones sin cifrar.

5.5.5 Impacto y relevancia actual

Los ataques Man-in-the-Middle continúan representando una amenaza importante debido a la gran cantidad de comunicaciones que se realizan constantemente a través de Internet y redes inalámbricas.

Su impacto puede ser extremadamente elevado, ya que permiten acceder a información sensible en tiempo real, incluyendo:

- Credenciales
- Datos bancarios
- Cookies de sesión
- Información corporativa
- Comunicaciones privadas

Además, estos ataques son especialmente peligrosos porque muchas veces las víctimas no perciben ningún comportamiento extraño durante la comunicación.

Aunque la adopción de HTTPS y protocolos seguros ha reducido ciertos riesgos, todavía existen numerosos sistemas mal configurados o aplicaciones vulnerables que pueden ser explotadas.

5.5.6 Medidas de prevención y defensa

La protección frente a ataques MitM requiere asegurar tanto la red como las comunicaciones.

Entre las principales medidas defensivas destacan:

- Uso obligatorio de HTTPS y TLS
- Verificación de certificados digitales
- Implementación de VPN
- Uso de HSTS para evitar SSL stripping
- Protección frente a ARP spoofing
- Evitar redes WiFi públicas inseguras
- Segmentación de red
- Autenticación multifactor (MFA)
- Monitorización de tráfico sospechoso

También es importante que los usuarios verifiquen alertas de certificados y eviten ignorar advertencias de seguridad del navegador.

5.5.7 Defensa avanzada

Las estrategias avanzadas frente a MitM incluyen arquitecturas de seguridad basadas en autenticación continua y cifrado extremo a extremo.

Las organizaciones modernas utilizan:

- Zero Trust Architecture
- Sistemas NAC
- Detección avanzada de anomalías de red
- Certificate Pinning
- DNS seguro (DoH / DoT)
- IDS/IPS especializados

Además, las soluciones EDR y NDR permiten detectar patrones relacionados con manipulación de tráfico o comportamiento anómalo en tiempo real.

El uso de autenticación mutua mediante certificados digitales también dificulta considerablemente este tipo de ataques.

5.5.8 Análisis crítico

Los ataques Man-in-the-Middle reflejan la importancia crítica de proteger las comunicaciones y validar la identidad de los sistemas implicados.

Aunque muchas tecnologías modernas incorporan mecanismos de cifrado robustos, la seguridad final sigue dependiendo de la correcta implementación y configuración de estos sistemas.

Además, los atacantes suelen combinar técnicas MitM con spoofing, sniffing o ingeniería social para aumentar sus probabilidades de éxito.

La evolución constante de las redes inalámbricas, dispositivos IoT y servicios cloud hace que este tipo de ataques continúe siendo relevante en la actualidad.

Desde una perspectiva defensiva, los ataques MitM evidencian que la seguridad de una infraestructura no depende únicamente de proteger los dispositivos finales, sino también de garantizar la integridad y autenticidad de toda la comunicación entre sistemas.

5.6 Session Hijacking

5.6.1 Definición

El session hijacking, o secuestro de sesión, es una técnica de ataque mediante la cual un atacante obtiene y utiliza de forma ilegítima una sesión autenticada de un usuario para acceder a sistemas o servicios sin necesidad de conocer sus credenciales.

Este tipo de ataque se basa en el robo o manipulación de identificadores de sesión, normalmente almacenados en cookies o tokens utilizados por aplicaciones web para mantener la autenticación del usuario una vez iniciado sesión.

Desde el punto de vista de la seguridad, el session hijacking compromete principalmente la autenticidad y confidencialidad, ya que el atacante puede actuar como si fuese el usuario legítimo y acceder a información o funcionalidades restringidas.

5.6.2 Funcionamiento técnico

El funcionamiento del session hijacking depende de la obtención del identificador de sesión válido de la víctima.

En aplicaciones web, cuando un usuario inicia sesión correctamente, el servidor genera un token o cookie de sesión que identifica la conexión autenticada. Mientras ese identificador siga siendo válido, el usuario puede interactuar con la aplicación sin volver a introducir sus credenciales.

El atacante busca capturar o manipular dicho identificador mediante distintas técnicas, como:

- Sniffing de tráfico no cifrado
- Ataques Man-in-the-Middle
- Cross-Site Scripting (XSS)
- Malware o keyloggers
- Predicción de tokens débiles
- Robo de cookies almacenadas

Una vez obtenido el token de sesión, el atacante puede reutilizarlo para acceder directamente a la cuenta de la víctima.

En algunos casos, el usuario legítimo continúa utilizando la sesión sin percibir que está siendo comprometida simultáneamente.

5.6.3 Tipos y características

El session hijacking puede clasificarse según la técnica utilizada para comprometer la sesión.

Session Sidejacking

Consiste en capturar cookies de sesión mediante sniffing en redes inseguras.

Session Fixation

El atacante fuerza al usuario a autenticarse utilizando un identificador de sesión previamente conocido por el atacante.

Session Prediction

Se basa en predecir tokens de sesión generados mediante algoritmos débiles o poco aleatorios.

Session Replay

Implica reutilizar identificadores de sesión capturados previamente para restaurar el acceso autenticado.

Características principales

- Uso ilegítimo de sesiones válidas
- No requiere conocer la contraseña
- Alta efectividad en aplicaciones mal protegidas
- Dependencia de la seguridad de las cookies y tokens
- Difícil detección en algunos escenarios

5.6.4 Ejemplos reales

Uno de los ejemplos más conocidos relacionados con session hijacking fue el uso de la herramienta Firesheep, que permitía capturar cookies de sesión en redes WiFi públicas y acceder a cuentas de servicios web sin necesidad de contraseña.

También se han detectado ataques contra aplicaciones web vulnerables a XSS, donde scripts maliciosos robaban automáticamente cookies de sesión de usuarios autenticados.

En entornos corporativos, el robo de sesiones VPN o paneles administrativos ha permitido a atacantes acceder a recursos internos sin activar mecanismos tradicionales de autenticación.

Asimismo, muchas aplicaciones antiguas transmitían identificadores de sesión mediante HTTP sin cifrado, facilitando enormemente este tipo de ataques.

5.6.5 Impacto y relevancia actual

El session hijacking continúa siendo una amenaza importante debido al uso masivo de aplicaciones web y servicios online basados en sesiones autenticadas.

Su impacto puede ser especialmente grave porque permite al atacante acceder directamente a cuentas legítimas, evitando mecanismos de protección basados únicamente en contraseñas.

Entre las posibles consecuencias destacan:

- Acceso no autorizado a cuentas
- Robo de información personal o corporativa
- Suplantación de identidad
- Fraude financiero
- Escalada de privilegios

Aunque actualmente el uso generalizado de HTTPS ha reducido ciertos escenarios de explotación, siguen existiendo aplicaciones vulnerables por mala gestión de sesiones o implementación incorrecta de cookies.

5.6.6 Medidas de prevención y defensa

La protección frente al session hijacking requiere asegurar tanto las comunicaciones como la gestión de sesiones.

Entre las principales medidas defensivas destacan:

- Uso obligatorio de HTTPS
- Cookies seguras (Secure y HttpOnly)
- Implementación de SameSite en cookies
- Regeneración de tokens tras autenticación
- Expiración automática de sesiones
- Autenticación multifactor (MFA)
- Validación de actividad sospechosa
- Protección frente a XSS
- Cierre automático de sesiones inactivas

Además, resulta importante evitar el uso de redes públicas inseguras sin protección adicional mediante VPN.

5.6.7 Defensa avanzada

Las estrategias avanzadas frente al session hijacking incluyen mecanismos de autenticación adaptativa y monitorización continua del comportamiento del usuario.

Las soluciones modernas implementan:

- Token binding
- Validación contextual de sesiones
- Detección de anomalías de comportamiento
- Rotación dinámica de tokens
- Sistemas Zero Trust
- Análisis de geolocalización e IP
- EDR y monitorización avanzada de accesos

Asimismo, muchas plataformas cloud utilizan inteligencia artificial para detectar comportamientos incompatibles con la actividad normal del usuario y bloquear automáticamente sesiones sospechosas.

El uso de autenticación multifactor reduce considerablemente el impacto potencial de sesiones comprometidas.

5.6.8 Análisis crítico

El session hijacking pone de manifiesto que la autenticación no finaliza cuando el usuario introduce correctamente sus credenciales, sino que debe mantenerse protegida durante toda la sesión.

Aunque las tecnologías modernas han mejorado significativamente la seguridad de las comunicaciones, muchos riesgos continúan relacionados con errores de implementación en aplicaciones web y configuraciones inseguras.

Además, este tipo de ataque demuestra cómo técnicas aparentemente indirectas, como XSS o sniffing, pueden terminar comprometiendo completamente la identidad digital del usuario.

Desde una perspectiva defensiva, el secuestro de sesión resalta la importancia de implementar mecanismos robustos de gestión de sesiones, autenticación continua y protección de tokens en cualquier sistema conectado a Internet.

5.7 Simulación Laboratorio

En esta práctica se desarrolló un laboratorio virtualizado utilizando Kali Linux y Windows dentro de una red aislada con el objetivo de demostrar el funcionamiento de un ataque Man in the Middle mediante ARP Spoofing. A través de herramientas como Bettercap y Wireshark se consiguió posicionar la máquina atacante entre la víctima y el router, interceptando el tráfico de red generado por la máquina Windows.

Durante la simulación se analizaron paquetes ARP, consultas DNS y tráfico HTTP no cifrado, demostrando cómo un atacante puede capturar información sensible cuando se utilizan protocolos inseguros. Además, se configuró el reenvío de paquetes mediante IP Forwarding para permitir que la máquina Kali actuase temporalmente como intermediario entre ambos dispositivos.

Para una visualización detallada de la ejecución del ataque y los resultados obtenidos en el laboratorio, consulte el apartado 4 en la memoria práctica.

5.8 Conclusiones del bloque de ataques de red

Los ataques de red representan una de las amenazas más relevantes dentro del ámbito de la ciberseguridad debido a que afectan directamente a las comunicaciones entre sistemas. A diferencia de otros tipos de ataques centrados únicamente en vulnerabilidades locales, estos ataques aprovechan debilidades presentes en protocolos, configuraciones de red y mecanismos de autenticación.

A lo largo de este bloque se ha podido observar cómo técnicas como el sniffing, el spoofing o los ataques Man-in-the-Middle pueden combinarse entre sí para comprometer la confidencialidad, integridad y disponibilidad de la información.

Asimismo, se ha comprobado que muchos de estos ataques continúan siendo efectivos en la actualidad debido a configuraciones inseguras, falta de cifrado o malas prácticas tanto por parte de usuarios como de administradores de sistemas.

Desde una perspectiva defensiva, la protección frente a ataques de red requiere un enfoque multicapa basado en cifrado, segmentación, monitorización continua y modelos de seguridad modernos como Zero Trust.

Finalmente, este bloque demuestra la importancia de comprender el funcionamiento interno de las comunicaciones de red para poder identificar amenazas, aplicar medidas de mitigación adecuadas y mejorar la seguridad global de las infraestructuras tecnológicas.

6. Ataques a Aplicaciones Web

6.1.1 Definición de ataques a aplicaciones web

Los ataques a aplicaciones web son técnicas utilizadas para explotar vulnerabilidades presentes en páginas, servicios y aplicaciones accesibles mediante navegadores o sistemas conectados a Internet.

Actualmente, las aplicaciones web constituyen una parte fundamental de la infraestructura digital moderna, ya que permiten gestionar servicios bancarios, plataformas empresariales, comercio electrónico, redes sociales y múltiples sistemas corporativos. Debido a esta exposición constante, se han convertido en uno de los principales objetivos de los atacantes.

A diferencia de otros ataques centrados en la red o en el sistema operativo, los ataques web se dirigen específicamente contra errores de programación, fallos de validación o configuraciones inseguras presentes en la lógica de la aplicación.

Dentro de este bloque se analizarán algunas de las vulnerabilidades más relevantes y frecuentes en el ámbito web, como SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Path Traversal y File Inclusion.

6.1.2 Evolución de los ataques web

La evolución de los ataques web ha estado directamente relacionada con el crecimiento de Internet y la complejidad creciente de las aplicaciones modernas.

En las primeras etapas del desarrollo web, muchas aplicaciones carecían de mecanismos básicos de validación y seguridad, permitiendo ataques relativamente simples como inyecciones SQL o ejecución de scripts maliciosos.

Con el tiempo, el aumento de servicios online, aplicaciones dinámicas y APIs incrementó considerablemente la superficie de ataque. Los atacantes comenzaron a explotar no solo vulnerabilidades técnicas, sino también errores lógicos relacionados con autenticación, sesiones y control de acceso.

Actualmente, las aplicaciones web modernas utilizan tecnologías avanzadas y arquitecturas distribuidas, aunque continúan apareciendo vulnerabilidades derivadas de errores de programación, malas configuraciones o validaciones insuficientes.

Además, muchas amenazas actuales combinan diferentes técnicas web para aumentar su efectividad, integrando robo de sesiones, evasión de autenticación y exfiltración de información.

6.1.3 Objetivos de los ataques web

Los ataques a aplicaciones web persiguen diferentes objetivos dependiendo del tipo de vulnerabilidad explotada y del valor del sistema comprometido.

Entre los principales objetivos destacan:

- Acceso no autorizado a información
- Robo de credenciales y sesiones
- Ejecución de código malicioso
- Manipulación de bases de datos
- Escalada de privilegios
- Interrupción de servicios
- Distribución de malware
- Compromiso de usuarios legítimos

En muchos casos, las aplicaciones web representan un punto de entrada inicial hacia infraestructuras corporativas más amplias.

6.1.4 Vectores de ataque

Los vectores de ataque en aplicaciones web suelen estar relacionados con errores de desarrollo, validación insuficiente de datos o configuraciones inseguras.

Entre los más habituales destacan:

- Entradas de usuario sin validar
- Consultas SQL inseguras
- Gestión incorrecta de sesiones
- Permisos mal configurados
- Subida insegura de archivos
- Dependencias vulnerables
- APIs expuestas
- Falta de sanitización de datos

Muchos de estos problemas se producen durante el desarrollo de la aplicación y pueden permanecer ocultos durante largos periodos si no se realizan auditorías de seguridad adecuadas.

6.1.5 Público objetivo

Las aplicaciones web pueden ser objetivo de ataques independientemente de su tamaño o finalidad.

Los objetivos más habituales incluyen:

- Plataformas de comercio electrónico
- Servicios bancarios

- Aplicaciones corporativas
- Redes sociales
- APIs públicas
- Paneles administrativos
- Plataformas cloud
- Sistemas gubernamentales

Además, los usuarios finales también pueden verse afectados indirectamente mediante robo de credenciales, secuestro de sesiones o distribución de contenido malicioso.

6.1.6 Frecuencia y relevancia actual

En la actualidad, los ataques web representan una de las amenazas más frecuentes dentro de la ciberseguridad debido a la enorme cantidad de aplicaciones expuestas a Internet.

Muchas organizaciones dependen completamente de servicios web para su funcionamiento diario, lo que convierte estas plataformas en objetivos prioritarios para atacantes y grupos organizados.

Además, el desarrollo rápido de aplicaciones y la integración constante de nuevas funcionalidades incrementan el riesgo de introducir vulnerabilidades involuntariamente.

Actualmente, vulnerabilidades como SQL Injection o XSS continúan apareciendo incluso en aplicaciones modernas, especialmente cuando no se aplican buenas prácticas de desarrollo seguro.

6.1.7 Facilidad de explotación

Uno de los aspectos más preocupantes de los ataques web es la facilidad con la que muchas vulnerabilidades pueden explotarse.

Existen numerosas herramientas automatizadas capaces de detectar y explotar vulnerabilidades web sin requerir conocimientos extremadamente avanzados.

Asimismo, frameworks, laboratorios vulnerables y documentación pública facilitan enormemente el aprendizaje y automatización de este tipo de ataques.

Esto convierte las aplicaciones web mal protegidas en objetivos especialmente vulnerables frente a atacantes con distintos niveles de experiencia.

6.1.8 Importancia de la defensa

La protección de aplicaciones web requiere integrar la seguridad en todas las fases del desarrollo y mantenimiento del sistema.

Las estrategias defensivas modernas incluyen:

- Validación y sanitización de entradas
- Uso de consultas parametrizadas
- Gestión segura de sesiones
- Control de acceso robusto
- Hardening de servidores web
- Monitorización continua
- WAF (Web Application Firewall)
- Auditorías y pentesting periódico

Además, metodologías como DevSecOps buscan integrar la seguridad directamente en el ciclo de desarrollo de software.

La protección de aplicaciones web resulta especialmente crítica debido a que muchas veces representan la principal puerta de acceso a servicios e información sensible.

6.2 SQL Injection (SQLi)

6.2.1 Definición

SQL Injection (SQLi) es una vulnerabilidad de seguridad que permite a un atacante manipular consultas SQL ejecutadas por una aplicación web contra una base de datos.

Este tipo de ataque se produce cuando la aplicación no valida correctamente los datos introducidos por el usuario y estos son insertados directamente dentro de consultas SQL dinámicas.

El objetivo principal suele ser acceder, modificar o eliminar información almacenada en la base de datos, aunque en algunos casos también puede permitir autenticación no autorizada o incluso ejecución remota de código en el servidor.

Desde el punto de vista de la seguridad, SQL Injection compromete principalmente la confidencialidad, integridad y disponibilidad de la información almacenada.

6.2.2 Funcionamiento técnico

El funcionamiento de SQL Injection se basa en insertar instrucciones SQL maliciosas dentro de campos de entrada controlados por el usuario.

Cuando la aplicación construye consultas SQL concatenando directamente estos datos sin validación adecuada, el atacante puede alterar la lógica original de la consulta.

Por ejemplo, un formulario de login vulnerable podría permitir modificar la condición de autenticación para acceder sin conocer credenciales válidas.

Las vulnerabilidades SQLi suelen aparecer en:

- Formularios de login
- Buscadores web
- Parámetros URL
- APIs
- Campos de entrada dinámicos

Dependiendo del nivel de acceso y configuración de la base de datos, el atacante puede:

- Leer información sensible
- Modificar registros
- Eliminar datos
- Enumerar tablas y usuarios
- Obtener hashes de contraseñas
- Escalar el ataque hacia el sistema operativo

6.2.3 Tipos y características

SQL Injection puede clasificarse según la técnica utilizada y la respuesta obtenida.

In-band SQLi

El atacante obtiene resultados directamente en la respuesta de la aplicación.

Error-based SQLi

Se aprovechan mensajes de error SQL para obtener información sobre la base de datos.

Union-based SQLi

Utiliza la sentencia UNION para combinar resultados legítimos con datos obtenidos por el atacante.

Blind SQLi

La aplicación no devuelve errores visibles, por lo que el atacante deduce información mediante respuestas lógicas o temporales.

Time-based SQLi

Se utilizan retrasos en la respuesta del servidor para inferir información.

Características principales

- Explotación de entradas no validadas
- Acceso directo a bases de datos
- Alto impacto potencial
- Automatización sencilla

- Muy frecuente en aplicaciones inseguras

6.2.4 Ejemplos reales

SQL Injection ha sido responsable de numerosos incidentes de seguridad a gran escala.

Uno de los casos más conocidos afectó a Sony Pictures en 2011, donde vulnerabilidades SQLi permitieron acceder a información personal de millones de usuarios.

También han existido ataques contra plataformas gubernamentales, tiendas online y servicios bancarios debido a consultas SQL inseguras.

A pesar de ser una vulnerabilidad conocida desde hace décadas, continúa apareciendo en aplicaciones modernas desarrolladas sin prácticas adecuadas de desarrollo seguro.

6.2.5 Impacto y relevancia actual

SQL Injection sigue siendo una de las vulnerabilidades web más peligrosas debido al acceso directo que proporciona a la base de datos.

Las consecuencias pueden incluir:

- Robo masivo de información
- Filtración de credenciales
- Modificación o destrucción de datos
- Acceso administrativo
- Interrupción de servicios
- Compromiso total de aplicaciones web

Además, muchas bases de datos contienen información extremadamente sensible, aumentando considerablemente el impacto potencial de este tipo de ataques.

6.2.6 Medidas de prevención y defensa

La principal defensa frente a SQL Injection consiste en evitar la construcción insegura de consultas SQL.

Entre las medidas más importantes destacan:

- Uso de consultas parametrizadas
- Prepared Statements
- Validación y sanitización de entradas
- Principio de mínimo privilegio en bases de datos
- Gestión segura de errores
- WAF (Web Application Firewall)

- Auditorías de código
- Escaneo periódico de vulnerabilidades

Además, resulta fundamental evitar mostrar errores SQL detallados al usuario.

6.2.7 Defensa avanzada

Las soluciones avanzadas frente a SQLi incluyen mecanismos capaces de detectar patrones anómalos en consultas y tráfico web.

Entre las tecnologías más utilizadas destacan:

- WAF con reglas anti-SQLi
- Monitorización de consultas SQL
- Detección basada en comportamiento
- Segmentación de bases de datos
- Sistemas de análisis dinámico de aplicaciones
- Protección RASP (Runtime Application Self-Protection)

Asimismo, metodologías DevSecOps permiten detectar vulnerabilidades SQLi durante el desarrollo antes de desplegar la aplicación.

6.2.8 Análisis crítico

SQL Injection demuestra cómo errores aparentemente simples de programación pueden generar vulnerabilidades extremadamente críticas.

Aunque actualmente existen mecanismos efectivos para prevenir este tipo de ataques, muchas aplicaciones continúan siendo vulnerables debido a malas prácticas de desarrollo o reutilización de código inseguro.

Además, la facilidad de automatización y el impacto potencial convierten SQLi en una de las técnicas más relevantes dentro de la ciberseguridad ofensiva.

Desde una perspectiva defensiva, esta vulnerabilidad evidencia la importancia de integrar la seguridad desde las primeras fases del desarrollo de aplicaciones web.

6.3 Cross-Site Scripting (XSS)

6.3.1 Definición

Cross-Site Scripting (XSS) es una vulnerabilidad web que permite a un atacante inyectar scripts maliciosos dentro de páginas web visualizadas por otros usuarios.

El objetivo principal suele ser ejecutar código JavaScript en el navegador de la víctima para robar sesiones, manipular contenido, redirigir usuarios o realizar acciones no autorizadas.

Este tipo de vulnerabilidad aparece cuando la aplicación web muestra contenido proporcionado por el usuario sin validarlo o sanitizarlo correctamente.

Desde el punto de vista de la seguridad, XSS compromete principalmente la confidencialidad e integridad de la interacción entre usuario y aplicación.

6.3.2 Funcionamiento técnico

El funcionamiento de XSS se basa en insertar código ejecutable dentro de contenido procesado por la aplicación web.

Cuando la aplicación devuelve este contenido al navegador sin filtrado adecuado, el script se ejecuta automáticamente en el contexto de la página legítima.

El atacante puede utilizar este comportamiento para:

- Robar cookies de sesión
- Capturar credenciales
- Modificar contenido web
- Redirigir usuarios
- Ejecutar acciones en nombre de la víctima

Las vulnerabilidades XSS suelen encontrarse en:

- Formularios
- Comentarios
- Chats
- Parámetros URL
- Paneles administrativos
- APIs con contenido HTML dinámico

6.3.3 Tipos y características

XSS puede clasificarse según la forma en que el código malicioso es almacenado o ejecutado.

Stored XSS

El script malicioso queda almacenado permanentemente en la aplicación y afecta a múltiples usuarios.

Reflected XSS

El código se refleja inmediatamente en la respuesta del servidor mediante parámetros manipulados.

DOM-based XSS

La vulnerabilidad se produce directamente en el navegador mediante manipulación insegura del DOM.

Características principales

- Ejecución de código en el navegador
- Robo de sesiones y credenciales
- Dependencia de validación insuficiente
- Alta interacción con el usuario
- Muy frecuente en aplicaciones dinámicas

6.3.4 Ejemplos reales

Numerosas redes sociales, foros y plataformas web han sufrido vulnerabilidades XSS a lo largo de los años.

Uno de los casos más conocidos ocurrió en MySpace, donde un gusano basado en XSS logró propagarse automáticamente entre perfiles de usuarios.

También se han detectado vulnerabilidades XSS en plataformas bancarias, paneles administrativos y aplicaciones corporativas capaces de comprometer sesiones autenticadas.

Actualmente, XSS continúa siendo una de las vulnerabilidades web más frecuentes en auditorías de seguridad.

6.3.5 Impacto y relevancia actual

El impacto de XSS depende del contexto de la aplicación comprometida y de los privilegios de la víctima afectada.

Entre las principales consecuencias destacan:

- Robo de cookies y sesiones
- Suplantación de usuarios
- Captura de credenciales
- Manipulación visual de páginas
- Distribución de malware
- Redirección a sitios fraudulentos

Además, las aplicaciones modernas basadas en JavaScript y contenido dinámico aumentan considerablemente la superficie de ataque relacionada con XSS.

6.3.6 Medidas de prevención y defensa

La protección frente a XSS requiere validar y sanitizar correctamente todo contenido introducido por usuarios.

Entre las principales medidas defensivas destacan:

- Escape de caracteres especiales
- Sanitización de entradas y salidas
- Uso de Content Security Policy (CSP)
- Cookies HttpOnly
- Validación de contenido HTML
- Frameworks seguros
- Protección frente a DOM manipulation insegura

Asimismo, resulta fundamental evitar insertar directamente contenido dinámico dentro del HTML o JavaScript sin tratamiento adecuado.

6.3.7 Defensa avanzada

Las soluciones avanzadas frente a XSS incluyen mecanismos capaces de bloquear ejecución de scripts no autorizados y detectar comportamiento sospechoso.

Entre las tecnologías más utilizadas destacan:

- CSP avanzada
- WAF con protección XSS
- Sanitización automática
- Trusted Types
- Monitorización de scripts dinámicos
- RASP (Runtime Application Self-Protection)

Además, muchos frameworks modernos incorporan mecanismos automáticos de protección frente a XSS.

6.3.8 Análisis crítico

Cross-Site Scripting evidencia cómo la interacción dinámica entre usuario y navegador puede convertirse en un vector crítico de ataque cuando no se aplican controles adecuados.

Aunque las tecnologías web modernas han mejorado considerablemente la protección frente a XSS, muchas aplicaciones siguen siendo vulnerables debido a validaciones insuficientes o manipulación insegura del DOM.

Además, este tipo de vulnerabilidad demuestra que la seguridad web no depende únicamente del servidor, sino también de cómo el navegador interpreta y ejecuta el contenido recibido.

Desde una perspectiva defensiva, XSS pone de manifiesto la importancia de aplicar principios de desarrollo seguro, validación estricta y políticas de ejecución controlada en aplicaciones web modernas.

6.4 Cross-Site Request Forgery (CSRF)

6.4.1 Definición

Cross-Site Request Forgery (CSRF) es una vulnerabilidad web que permite a un atacante forzar a un usuario autenticado a realizar acciones no deseadas dentro de una aplicación web en la que ya ha iniciado sesión.

Este ataque aprovecha la confianza que la aplicación deposita en el navegador del usuario autenticado, utilizando automáticamente sus cookies o tokens de sesión válidos para ejecutar solicitudes maliciosas.

El objetivo principal suele ser realizar acciones en nombre de la víctima sin su consentimiento, como modificar configuraciones, transferir información o ejecutar operaciones administrativas.

Desde el punto de vista de la seguridad, CSRF compromete principalmente la integridad y autenticidad de las acciones realizadas dentro de la aplicación.

6.4.2 Funcionamiento técnico

El funcionamiento de CSRF se basa en engañar al navegador de la víctima para que envíe solicitudes legítimas hacia una aplicación donde el usuario ya está autenticado.

El atacante crea una página o enlace malicioso que contiene una solicitud preparada hacia el servicio objetivo.

Cuando la víctima accede a dicho contenido mientras mantiene una sesión activa, el navegador envía automáticamente:

- Cookies de autenticación
- Tokens de sesión
- Cabeceras asociadas a la sesión

La aplicación interpreta la solicitud como legítima porque proviene de un usuario autenticado correctamente.

Las acciones ejecutadas pueden incluir:

- Cambios de contraseña
- Transferencias bancarias
- Modificación de configuraciones
- Creación de usuarios
- Eliminación de información

CSRF resulta especialmente peligroso en aplicaciones que no validan adecuadamente el origen de las solicitudes.

6.4.3 Tipos y características

Aunque CSRF suele considerarse una única categoría de ataque, puede presentarse de distintas formas según la técnica utilizada.

CSRF basado en formularios

La solicitud maliciosa se ejecuta mediante formularios HTML ocultos.

CSRF mediante enlaces

La víctima ejecuta la acción simplemente accediendo a una URL manipulada.

Login CSRF

El atacante fuerza a la víctima a autenticarse con credenciales controladas por el atacante.

Características principales

- Requiere una sesión autenticada activa
- Explota la confianza de la aplicación en el navegador
- No necesita robar credenciales directamente
- Alta dependencia de interacción del usuario
- Muy efectivo en aplicaciones sin protección CSRF

6.4.4 Ejemplos reales

Numerosas plataformas web han sufrido vulnerabilidades CSRF debido a la ausencia de mecanismos de validación adecuados.

En algunos casos, atacantes han conseguido modificar configuraciones de cuentas, realizar acciones administrativas o ejecutar transferencias financieras utilizando solicitudes CSRF.

También se han detectado vulnerabilidades CSRF en routers domésticos y paneles de administración internos donde bastaba visitar una página maliciosa para alterar configuraciones del dispositivo.

Aunque actualmente muchos frameworks incluyen protecciones automáticas, CSRF continúa apareciendo en aplicaciones desarrolladas sin controles adecuados.

6.4.5 Impacto y relevancia actual

El impacto de CSRF depende de las acciones que la aplicación permita realizar mediante solicitudes autenticadas.

Entre las principales consecuencias destacan:

- Modificación de cuentas de usuario
- Transferencias no autorizadas
- Alteración de configuraciones
- Creación o eliminación de usuarios
- Cambios de contraseña
- Compromiso parcial de servicios

Aunque las tecnologías modernas han reducido parte del riesgo, muchas aplicaciones siguen siendo vulnerables cuando no implementan correctamente mecanismos anti-CSRF.

6.4.6 Medidas de prevención y defensa

La protección frente a CSRF requiere validar que las solicitudes provienen realmente del usuario legítimo.

Entre las principales medidas defensivas destacan:

- Uso de tokens anti-CSRF
- Validación del encabezado Origin o Referer
- Cookies SameSite
- Reautenticación en acciones críticas
- Separación entre métodos GET y POST
- Expiración de sesiones
- Confirmación adicional de operaciones sensibles

Además, resulta importante evitar que acciones críticas puedan ejecutarse únicamente mediante solicitudes simples.

6.4.7 Defensa avanzada

Las soluciones avanzadas frente a CSRF incluyen mecanismos dinámicos de validación contextual y análisis de comportamiento.

Entre las tecnologías más utilizadas destacan:

- Tokens CSRF rotativos
- SameSite Strict
- Validación contextual de sesiones
- Detección de solicitudes anómalas
- WAF especializado
- Monitorización de comportamiento web

Asimismo, muchos frameworks modernos incorporan protección automática frente a CSRF de forma predeterminada.

6.4.8 Análisis crítico

CSRF demuestra que la seguridad web no depende únicamente de proteger credenciales o sesiones, sino también de validar correctamente la intención y legitimidad de las acciones realizadas.

Este tipo de vulnerabilidad evidencia cómo la confianza automática entre navegador y aplicación puede convertirse en un vector de ataque si no se aplican mecanismos de validación adecuados.

Además, CSRF pone de manifiesto la importancia de integrar controles de autenticidad en todas las operaciones críticas de una aplicación web.

Desde una perspectiva defensiva, este ataque resalta la necesidad de combinar protección de sesiones, validación de origen y control contextual de solicitudes para garantizar la seguridad de las aplicaciones modernas.

6.5 Path Traversal

6.5.1 Definición

Path Traversal, también conocido como Directory Traversal, es una vulnerabilidad web que permite a un atacante acceder a archivos o directorios fuera de las rutas previstas por la aplicación.

Este ataque aprovecha validaciones insuficientes en rutas de archivos manipuladas mediante secuencias especiales que permiten desplazarse por el sistema de directorios del servidor.

El objetivo principal suele ser acceder a información sensible almacenada en el sistema, como configuraciones, credenciales, archivos internos o datos de otros usuarios.

Desde el punto de vista de la seguridad, Path Traversal compromete principalmente la confidencialidad de la información y, en algunos casos, puede facilitar ataques más avanzados contra el sistema.

6.5.2 Funcionamiento técnico

El funcionamiento de Path Traversal se basa en manipular rutas de archivos utilizadas por la aplicación web.

Cuando una aplicación permite al usuario indicar nombres o ubicaciones de archivos sin validación adecuada, el atacante puede insertar secuencias especiales para salir del directorio previsto.

Las secuencias más habituales incluyen:

- ../
- ..\
- rutas absolutas
- codificaciones alternativas

De esta forma, el atacante puede intentar acceder a:

- Archivos del sistema operativo
- Configuraciones internas
- Credenciales
- Logs
- Claves privadas
- Archivos de aplicaciones

En sistemas Linux, por ejemplo, un atacante podría intentar acceder a archivos sensibles del sistema mediante rutas manipuladas.

La gravedad del ataque depende de los permisos disponibles para la aplicación vulnerable.

6.5.3 Tipos y características

Path Traversal puede presentarse de distintas formas dependiendo del entorno y técnica utilizada.

Relative Path Traversal

Se utilizan rutas relativas para desplazarse entre directorios.

Absolute Path Traversal

El atacante intenta acceder directamente mediante rutas completas del sistema.

Encoded Traversal

Las secuencias son codificadas para evadir filtros de seguridad.

Características principales

- Acceso no autorizado a archivos
- Aprovechamiento de validaciones insuficientes
- Riesgo elevado en servidores mal configurados
- Posibilidad de escalada hacia otros ataques
- Alta dependencia de permisos del sistema

6.5.4 Ejemplos reales

Numerosas aplicaciones web han sufrido vulnerabilidades Path Traversal permitiendo acceso a archivos críticos del servidor.

En algunos casos, los atacantes consiguieron obtener configuraciones internas, claves de acceso o credenciales almacenadas en archivos locales.

También se han detectado vulnerabilidades Path Traversal en dispositivos IoT, paneles administrativos y servidores empresariales.

Muchas veces, estas vulnerabilidades aparecen en funciones de descarga de archivos, visualización de imágenes o sistemas de carga dinámica de contenido.

6.5.5 Impacto y relevancia actual

El impacto de Path Traversal depende de los archivos accesibles desde la aplicación vulnerable.

Entre las principales consecuencias destacan:

- Exposición de información sensible
- Robo de credenciales
- Acceso a configuraciones internas
- Divulgación de código fuente
- Preparación para ataques posteriores
- Compromiso parcial del servidor

Aunque actualmente existen mejores prácticas de desarrollo seguro, este tipo de vulnerabilidad continúa apareciendo en aplicaciones mal diseñadas o con validaciones insuficientes.

6.5.6 Medidas de prevención y defensa

La protección frente a Path Traversal requiere restringir completamente el acceso a rutas no autorizadas.

Entre las principales medidas defensivas destacan:

- Validación estricta de rutas
- Uso de listas blancas de archivos permitidos
- Normalización de rutas
- Restricción de permisos del sistema
- Aislamiento de directorios accesibles
- Evitar acceso directo a rutas del sistema
- Hardening del servidor

Asimismo, resulta importante limitar los privilegios de las aplicaciones web sobre el sistema operativo.

6.5.7 Defensa avanzada

Las soluciones avanzadas frente a Path Traversal incluyen mecanismos capaces de detectar accesos anómalos a archivos y manipulación de rutas.

Entre las tecnologías más utilizadas destacan:

- WAF con protección anti-traversal
- Sandboxing de aplicaciones
- Monitorización de accesos a archivos
- Contenedorización
- Políticas SELinux o AppArmor
- IDS/IPS especializados

Además, los entornos modernos aplican aislamiento entre aplicaciones y sistemas de archivos para reducir el impacto potencial.

6.5.8 Análisis crítico

Path Traversal demuestra cómo errores aparentemente simples en la gestión de rutas pueden derivar en vulnerabilidades críticas capaces de exponer información sensible del sistema.

Aunque la técnica es conocida desde hace años, sigue apareciendo en aplicaciones modernas debido a validaciones insuficientes o diseños inseguros.

Además, este tipo de ataque evidencia la importancia de aplicar correctamente principios de aislamiento y mínimo privilegio dentro de aplicaciones web.

Desde una perspectiva defensiva, Path Traversal resalta la necesidad de controlar estrictamente cualquier interacción entre usuarios y el sistema de archivos del servidor.

6.6 File Inclusion

6.6.1 Definición

File Inclusion es una vulnerabilidad web que permite a un atacante forzar a una aplicación a cargar o ejecutar archivos no previstos originalmente por el sistema.

Este tipo de ataque suele producirse cuando la aplicación utiliza rutas o nombres de archivos proporcionados por el usuario sin validación adecuada.

Dependiendo de la configuración del servidor y del tipo de inclusión utilizada, el atacante puede acceder a archivos locales del sistema o incluso ejecutar archivos remotos controlados externamente.

Desde el punto de vista de la seguridad, File Inclusion puede comprometer la confidencialidad, integridad y disponibilidad del sistema, especialmente cuando permite ejecución de código arbitrario.

6.6.2 Funcionamiento técnico

El funcionamiento de File Inclusion se basa en manipular parámetros utilizados por la aplicación para cargar archivos dinámicamente.

Muchas aplicaciones web utilizan funciones que incluyen archivos dependiendo de parámetros enviados por el usuario.

Si no existen controles adecuados, el atacante puede modificar dichos parámetros para:

- Acceder a archivos sensibles
- Ejecutar código malicioso

- Cargar scripts externos
- Obtener información del sistema

Las vulnerabilidades suelen aparecer en:

- Sistemas PHP antiguos
- Motores de plantillas inseguros
- Módulos dinámicos
- Paneles administrativos
- Aplicaciones con carga dinámica de contenido

La gravedad depende de si la aplicación permite únicamente archivos locales o también recursos remotos.

6.6.3 Tipos y características

File Inclusion puede clasificarse principalmente en dos categorías.

Local File Inclusion (LFI)

Permite incluir archivos almacenados localmente en el servidor.

El atacante puede intentar acceder a:

- Archivos de configuración
- Logs
- Credenciales
- Código fuente
- Archivos internos del sistema

Remote File Inclusion (RFI)

Permite cargar archivos remotos desde servidores externos controlados por el atacante.

Este tipo de vulnerabilidad puede facilitar:

- Ejecución remota de código
- Instalación de malware
- Compromiso completo del servidor

Características principales

- Manipulación de carga dinámica de archivos
- Riesgo elevado de ejecución de código
- Dependencia de validación insuficiente
- Alta gravedad en servidores mal configurados
- Posible escalada hacia compromiso total

6.6.4 Ejemplos reales

Numerosas aplicaciones PHP antiguas han sufrido vulnerabilidades LFI y RFI debido al uso inseguro de funciones de inclusión dinámica.

En muchos casos, atacantes consiguieron acceder a archivos críticos del sistema o ejecutar shells remotas sobre el servidor comprometido.

También se han detectado vulnerabilidades File Inclusion en CMS, paneles administrativos y aplicaciones empresariales.

En combinación con otras vulnerabilidades, como subida insegura de archivos o Path Traversal, File Inclusion puede facilitar compromisos completos de servidores web.

6.6.5 Impacto y relevancia actual

El impacto de File Inclusion puede ser extremadamente grave dependiendo de las capacidades obtenidas por el atacante.

Entre las principales consecuencias destacan:

- Acceso a archivos sensibles
- Ejecución remota de código
- Instalación de backdoors
- Robo de información
- Compromiso del servidor
- Movimiento lateral dentro de la infraestructura

Aunque actualmente muchos frameworks modernos han reducido estos riesgos, siguen existiendo aplicaciones vulnerables desarrolladas con prácticas inseguras.

6.6.6 Medidas de prevención y defensa

La protección frente a File Inclusion requiere controlar estrictamente cualquier carga dinámica de archivos.

Entre las principales medidas defensivas destacan:

- Validación estricta de rutas y nombres
- Uso de listas blancas
- Desactivación de inclusión remota
- Restricción de permisos del sistema
- Hardening del servidor web
- Separación entre contenido y ejecución
- Actualización de frameworks y dependencias
- Monitorización de actividad sospechosa

Asimismo, resulta fundamental evitar utilizar directamente parámetros controlados por usuarios para cargar archivos del sistema.

6.6.7 Defensa avanzada

Las soluciones avanzadas frente a File Inclusion incluyen mecanismos capaces de detectar comportamientos anómalos relacionados con acceso o ejecución de archivos.

Entre las tecnologías más utilizadas destacan:

- WAF con protección LFI/RFI
- Sandboxing de aplicaciones
- Monitorización de integridad de archivos
- IDS/IPS especializados
- Contenedorización
- SELinux o AppArmor
- Runtime Application Self-Protection (RASP)

Además, muchas arquitecturas modernas utilizan aislamiento estricto entre aplicaciones y sistema operativo para minimizar el impacto potencial.

6.6.8 Análisis crítico

File Inclusion evidencia cómo errores en la gestión dinámica de archivos pueden convertirse en vulnerabilidades extremadamente críticas.

Aunque actualmente muchas tecnologías modernas incorporan mecanismos más seguros, numerosas aplicaciones heredadas continúan siendo vulnerables debido a configuraciones inseguras o validaciones insuficientes.

Además, este tipo de vulnerabilidad demuestra la importancia de limitar la interacción directa entre entradas de usuario y componentes internos del sistema.

Desde una perspectiva defensiva, File Inclusion resalta la necesidad de aplicar aislamiento, validación estricta y control de ejecución dentro de cualquier aplicación web.

6.7 Simulación Laboratorio

Este ataque no se ha realizado, sin embargo, explicaremos cómo se llevaría a cabo.

El ataque SQL Injection consiste en introducir instrucciones SQL maliciosas dentro de campos de entrada de una aplicación web vulnerable con el objetivo de manipular las consultas realizadas a la base de datos.

Este tipo de ataque puede permitir:

- Acceder sin autenticación,
- Visualizar información sensible,
- Modificar registros,
- Comprometer completamente la base de datos.

Se recreará un escenario básico de autenticación vulnerable donde el sistema no valida correctamente los datos introducidos por el usuario. Gracias a ello sería posible alterar la consulta SQL original y acceder al sistema sin disponer de credenciales legítimas.

6.8 Conclusiones del bloque de ataques a aplicaciones web

Los ataques a aplicaciones web representan una de las amenazas más relevantes dentro de la ciberseguridad actual debido a la enorme cantidad de servicios expuestos constantemente a Internet.

A lo largo de este bloque se ha podido observar cómo vulnerabilidades relacionadas con validación insuficiente, gestión insegura de sesiones o manipulación incorrecta de datos pueden comprometer aplicaciones completas y permitir accesos no autorizados a información sensible.

Asimismo, se ha analizado cómo técnicas como SQL Injection, XSS o File Inclusion continúan siendo utilizadas en la actualidad debido a errores de programación, configuraciones inseguras y ausencia de controles adecuados durante el desarrollo.

Uno de los aspectos más importantes es que muchas de estas vulnerabilidades pueden prevenirse aplicando buenas prácticas de desarrollo seguro, validación estricta de entradas y mecanismos adecuados de autenticación y control de acceso.

Desde una perspectiva defensiva, la protección de aplicaciones web requiere integrar la seguridad en todas las fases del desarrollo, mantenimiento y despliegue del sistema, utilizando metodologías modernas como DevSecOps y herramientas de monitorización continua.

Finalmente, este bloque demuestra que las aplicaciones web constituyen uno de los principales puntos de exposición de cualquier organización moderna, convirtiendo la seguridad web en un elemento crítico dentro de cualquier estrategia global de ciberseguridad.

7. Ataques de fuerza Bruta

7.1 Definición

Uno de los métodos que llevan más tiempo siendo utilizados y que a día de hoy siguen siendo los más vigentes para comprometer los sistemas son los **ataques de fuerza bruta**. Como su nombre lo dice consiste en probar múltiples veces usuarios, contraseñas o claves, intentando e intentando hasta lograr adivinar cual es. Suena muy simple y fácil de defender en sistemas pero aunque no lo parezca es muy efectivo debido a las contraseñas débiles o malas configuraciones de seguridad.
“si pruebas todas las combinaciones posibles, eventualmente encontrarás la correcta”

7.2 ¿Cómo funciona la teoría detrás del ataque?

El éxito y el tiempo necesario para un ataque de fuerza bruta clásico dependen matemáticamente del espacio de claves (keyspace), que es el número total de combinaciones posibles. Esto está determinado por dos factores:

- La longitud de la contraseña.
- El conjunto de caracteres utilizado (letras minúsculas, mayúsculas, números, símbolos).

Lo que quiere decir que si una contraseña tiene solo 4 caracteres numéricos (un número PIN), el espacio de claves es de 10,000 combinaciones (del 0000 al 9999). Un ordenador moderno puede probar estas combinaciones en milisegundos. Sin embargo, a medida que aumenta la longitud y la complejidad de la clave, el tiempo requerido crece excesivamente demasiado, haciendo que la fuerza bruta pura sea inviable para contraseñas largas y complejas.

7.3 ¿Qué tipos de ataques por fuerza bruta hay?

A medida que hemos estado investigando y avanzando en el proyecto nos hemos dado cuenta que realizar ataques de “fuerza bruta pura“ es demasiado ineficiente, ya que probar cada combinación tarda muchísimo. Así que existen diferentes métodos que con los años fue evolucionando hacia técnicas más inteligentes y optimizadas, estas son:

- **Ataque de Fuerza Bruta Tradicional/Exhaustivo:** Prueba todas las combinaciones posibles de caracteres sin ninguna lógica previa. Es el método más lento.

- **Ataque de Diccionario:** En lugar de probar combinaciones aleatorias, nuestra herramienta utiliza una lista predefinida de palabras comunes, frases, o contraseñas filtradas previamente (como "123456", "password", "admin"). Es mucho más rápido y efectivo contra usuarios que usan contraseñas débiles.

- **Credential Stuffing (Relleno de Credenciales):** Los atacantes toman listas masivas de usuarios y contraseñas que han sido robadas de la brecha de seguridad de una página web e intentan usar esas mismas credenciales en otros sitios (bancos, correos, redes sociales), aprovechando que la gente suele reciclar sus contraseñas.

- **Fuerza Bruta Inversa (Reverse Brute Force):** El atacante toma una contraseña muy común (por ejemplo, "123456") y prueba esa única contraseña contra millones de nombres de usuario diferentes. Esto ayuda a evitar los bloqueos de cuenta que se activan cuando se falla mucho en un solo usuario.

- **Ataque de Tablas Rainbow (Rainbow Tables):** Es un método avanzado donde se utilizan tablas precalculadas de hashes criptográficos para revertir contraseñas cifradas en bases de datos **robadas**, así mismo el tiempo de procesamiento es increíblemente inferior.

7.4 Tiempos de adivinanza

Realizamos una búsqueda y encontramos una tabla que nos hace entrar mas en razon sobre lo tardado que puede ser realizar todas las combinaciones, si observas la tablas podemos ver que los números es la forma más fácil de adivinar una contraseña, en cambio con una mezcla de todo y alfanuméricos el tiempo puede llegar incluso milenios.

Longitud	Caracteres	Combinaciones	Tiempo promedio
6	Solo numeros	1 millon	<1 segundo
8	Minúsculas	208 Mil millones	1 hora
10	Alfanumericos	830 Billones	5 Años
12	Complejos	150 Trillones	Entre 2000 y 3000 años

Ejemplos :

- **Solo Números** : “123456” —- 0,09 Segundos.
- **Minúsculas:** “password” —- 57 minutos
- **Alfanuméricos :** “ataque123#” —- 5 años
- **Alfanuméricos :** “c0ntr@s3ñ@C0mpL!c#d@” —- 2000 años

7.5 Objetivos.

Como cada ataque, este también tiene unos objetivos, cada uno de estos busca un resultado diferente no todos tienen que ser con el objetivo de delinquir, también sirve para comprometer la CID **(Confidencialidad, Integridad y Disponibilidad).** Decidimos clasificarlos en 2 puntos dependiendo de la finalidad estratégica con la que se desee usar. A continuación :

7.5.1 Objetivos Técnicos y estratégicos

Si nos ponemos en la piel de una persona que quiere realizar un ataque de fuerza bruta, ¿qué crees que debería de pensar? Primero concentrarnos en unos **objetivos estratégicos**, ¿Con qué finalidad realizamos este ataque?. Podemos filtrar información sensible, Escalar privilegios, Realizar saltos en la red y así ampliar el alcance de la intrusión, Desplegar virus, malware, etc. O simplemente reclutar dispositivos para usarlos como botnets y hacer ataques DDos o minería. Una vez el objetivo es claro podemos pasar al siguiente paso .

pensar en los **objetivos técnicos,** ya tenemos definido una meta a la cual atacar, solo faltaría pensar cómo atacar, ¿ Queremos adivinar un usuario / contraseña? ¿Deberíamos descifrar datos? ¿Creamos un diccionario? ¿ Realizamos una sesión hijacking? ¿ Qué métodos de defensa podría tener la víctima ? Etc. Realizando así un mapa mental de posibilidades para lograr estos ataques de fuerza bruta.

7.5.2 Objetivos en la seguridad defensiva

Anteriormente hablamos de lo que un atacante podría hacer a una víctima y sus objetivos, pero y si usamos este ataque para emplear fines legítimos y éticos? Podríamos nosotros mismos ponernos en la piel de un atacante y predecir cómo nos van a atacar, así saber cómo defendernos. en empresas importantes y con muchos datos personales usualmente hacen cada X tiempo un **Pentesting** que consiste en ello, atacarte a ti mismo y probar que tan eficaz son tus sistemas de detección, Que tan robustas son tus políticas de contraseñas.

También podríamos usarlo en casos de **informática forense**, Imagina que un dispositivo es incautado en una investigación criminal y no tenemos métodos para adivinar la contraseña, La fuerza bruta con pistas, datos e incluso información de las escenas del crimen, podrían permitir crear el uso de diccionarios para adivinar claves de acceso desconocidas.
Los objetivos en ataque de fuerza brutas son en su mayoría casos no éticos y maliciosos, pero no significa que pueda usarse como una herramienta beneficiosa mas para otras situaciones

7.6 Un poco de Historia

La fuerza bruta no viene con la informática, tiene como origen siglos y siglos atrás, consiste en el intento manual de probar combinaciones en candados o cifrados de sustitución.
En la segunda Guerra mundial existían máquinas como la **enigma alemana(1)** Que manualmente era imposible de adivinar con combinaciones superiores a 10^22, Esto llevó a una persona llamada Alan Turing a crear la máquina **“bombas”(2)** Lo que hacía era automatizar el descarte de combinaciones muchisimo mas rapido, asentando así las primeras bases de la fuerza bruta

Pasamos a los años 1977 Donde se desarrolla el estándar **DES (Data Encryption Standard)** en Estados Unidos, Tenía una clave de 56 bits, lo que traducimos a 2^56 (que serían 72 mil billones de combinaciones). en esos años era totalmente seguro ya que ningún ordenador del mundo era capaz de adivinar esto en un tiempo que fuese razonable.
Pero no todo seria estable y seguro ya que en 1998 salió a la luz el **EFF DES Cracker,** La compañía **Electronic Frontier Foundation** construyó una máquina específica por menos de 250.000 dólares, con el fin de demostrar que des ya no es seguro, con un resultado de éxito de solo 56 horas, La consecuencia de esto es que obligó a la comunidad tecnológica a buscar un nuevo estándar, conocido y usado a dia de hoy como AES (Advanced Encryption Standard) Hoy usamos claves de 128, 192 y 256 bits.

En la década de los 2000 los atacantes descubrieron que las **GPU** diseñadas para videojuegos son increíbles para realizar cálculos matemáticos repetitivos, por lo que cambió el juego, lo que un procesador tardaba meses, una tarjeta grafica lo hacia en dias, con esto nacieron aplicaciones famosas como john the ripper y hashcat, lo que permitió que cualquier persona con un ordenador medianamente potente pudiese realizar ataques de fuerza bruta a contraseñas.

7.7 Fiabilidad y uso hoy en dia

La fuerza bruta representa una paradoja: es simultáneamente el método de ataque más fácil de defender y uno de los que mayor tasa de éxito sigue reportando en brechas de seguridad a nivel global.

Aunque los algoritmos de cifrado son cada vez más robustos como leímos antes de casi 256 bits, la fiabilidad de la fuerza bruta no depende de la matemática del cifrado, sino de la **psicología del usuario**.

Según informes de seguridad recientes, más del **80% de las brechas de confirmación de datos** están relacionadas con contraseñas débiles o reutilizadas. Esto significa que la fuerza bruta "pura" (probar combinaciones al azar) ha perdido viabilidad, pero la fuerza bruta "inteligente" (ataques de diccionario y *credential stuffing*) es más fiable que nunca.

7.7.1 ¿Sigue siendo una amenaza real?

La fiabilidad de este ataque se divide en dos escenarios críticos:

1. **En Entornos Online (Baja fiabilidad):** Hoy en día, realizar fuerza bruta contra servicios web (como Instagram, Gmail o portales bancarios) es prácticamente **inútil**. Los sistemas de bloqueo tras 3 o 5 intentos, los desafíos CAPTCHA y la implementación masiva de la **Autenticación Multifactor (MFA)** han reducido la tasa de éxito de la fuerza bruta directa a niveles casi despreciables.
2. **En Entornos Offline (Alta fiabilidad):** Si un atacante logra robar una base de datos de hashes, la fuerza bruta es **extremadamente fiable**. Con la potencia

de las GPUs actuales (como las series NVIDIA RTX 5000), las contraseñas de menos de 10 caracteres que no utilicen algoritmos de "estiramiento de clave" pueden ser vulneradas en cuestión de minutos u horas.

7.7.2. El Desplazamiento hacia el "Credential Stuffing"

la fuerza bruta ha evolucionado. El uso de ataques "ciegos" ha sido sustituido por el **Relleno de Credenciales**.

- **El factor masivo:** Los atacantes ya no intentan adivinar una contraseña; utilizan bots que prueban millones de combinaciones de correos y claves ya filtradas en otros sitios. En 2025, se estima que estos ataques representan casi el **40% de todos los intentos de inicio de sesión** en sitios de e-commerce a nivel mundial.

7.8 Herramientas populares

Una vez hemos visto todo el marco teórico y como se piensa el ataque, realizaremos la investigación de cómo se ejecutan estos ataques, encontramos 4 herramientas muy usadas entre los atacantes y los defensores, nosotros no seremos la excepción y usaremos algunas de ellas para la parte práctica, Pero primero veremos cómo funcionan:

7.8.1 Hashcat: El motor de paralelización masiva

Hashcat es el estándar de la industria para ataques **offline**. Su superioridad técnica radica en cómo traslada la carga de trabajo de la CPU a la GPU.

- **Arquitectura de Procesamiento:** Utiliza el concepto de **SIMD (Single Instruction, Multiple Data)**. Mientras una CPU está optimizada para tareas complejas secuenciales, Hashcat aprovecha los miles de núcleos de una GPU para ejecutar la misma operación matemática (el cálculo del hash) sobre miles de variantes de datos simultáneamente.
- **Lógica de "Workload Tuning":** Permite al usuario ajustar la intensidad del ataque mediante parámetros como **-w** (workload profile). Técnicamente, esto define cuánto tiempo la GPU se dedica exclusivamente a Hashcat antes de devolver el control al sistema operativo, optimizando el rendimiento térmico y de procesamiento.
- **Modo de Ataque por Máscara (Mask Attack):** En lugar de probar combinaciones al azar, utiliza una sintaxis técnica (ej. ?u?l?l?l?d?d) que define una estructura lógica (Mayúscula + 3 minúsculas + 2 números). Esto reduce el **espacio de búsqueda** de forma drástica comparado con la fuerza bruta exhaustiva.

7.8.2. John the Ripper

A diferencia de Hashcat, JtR brilla por su capacidad de entender la "psicología" de las contraseñas mediante reglas de transformación.

- **Single Crack Mode (Lógica de Permutación):** Es su modo más técnico y rápido. Toma información del sistema (nombre de usuario, ID, carpeta personal) y aplica un motor de reglas interno para generar variaciones (ej. si el usuario es jsmith, prueba Jsmith123, smith_j, etc.). Se basa en la premisa estadística de que los usuarios tienden a basar sus claves en su identidad.
- **Incremental Mode (Cadenas de Markov):** JtR no prueba caracteres en orden alfabético (aaa, aab, aac). Utiliza tablas de frecuencia de caracteres. Técnicamente, emplea **modelos de Markov** para predecir qué letra es más probable que siga a otra (ej. en español, después de una "q" es casi seguro que va una "u"). Esto hace que encuentre contraseñas en una fracción del tiempo de un ataque puramente lineal.

7.8.3 THC-Hydra: El cracker de protocolos multihilo

Hydra es una herramienta para ataques **online**. Su complejidad técnica reside en la gestión de redes.

- **Arquitectura "Head & Brain":** Hydra utiliza un modelo multihilo. Las "cabezas" (*heads*) son los hilos individuales que intentan una conexión; el "cerebro" (*brain*) gestiona qué combinaciones han sido probadas para evitar duplicados y manejar errores de red (timeouts).
- **Manejo de Estados de Red:** A nivel técnico, Hydra implementa módulos específicos para cada protocolo (SSH, FTP, HTTP). No solo envía texto, sino que debe entender el "apretón de manos" (*handshake*) y las respuestas del servidor para distinguir entre un error de red, un usuario bloqueado o una contraseña correcta.

7.8.4 Medusa: Modularidad y Estabilidad

Es el competidor directo de Hydra, preferido en auditorías por su estabilidad en conexiones inestables.

- **Diseño Modular (.mod):** Cada servicio que Medusa ataca es un módulo independiente cargado dinámicamente. Esto permite que el núcleo del programa sea muy ligero y eficiente en el uso de memoria.
- **Paralelismo de Host/Usuario:** Técnicamente, permite configurar si quieres atacar múltiples usuarios en un host simultáneamente, o un mismo usuario en múltiples hosts. Esta granularidad es vital para evitar sistemas de detección de intrusos (IDS).

7.9 Simulación Laboratorio

La experimentación realizada en el laboratorio mediante ataques de fuerza bruta y diccionarios sobre el protocolo SMB permite extraer las siguientes conclusiones fundamentales:

- Vulnerabilidad de las configuraciones por defecto: El Escenario A demuestra que, sin directivas de seguridad activas, un ataque de fuerza bruta es extremadamente simple y rápido. El uso de herramientas de escaneo como Nmap y de explotación como NetExec (nxc) permite comprometer credenciales débiles en cuestión de segundos, subrayando que la comodidad del usuario es la mayor brecha de seguridad.
- Insuficiencia de las defensas reactivas simples: El Escenario B evidencia que medidas básicas, como el bloqueo temporal de cuentas, pueden ser evadidas mediante automatización y scripting. El atacante, ajustando los tiempos de respuesta (*sleep timers*) y personalizando diccionarios, puede mantener la persistencia del ataque sin activar las alertas de seguridad, lo que demuestra que la defensa debe ser más profunda que un simple contador de intentos.
- Eficacia de la Complejidad y el Factor Tiempo: En el Escenario C, se confirma que la combinación de contraseñas robustas (alfanuméricas con símbolos) y directivas de bloqueo estrictas desplaza el tiempo de compromiso de minutos a milenios. Esto vuelve el ataque de fuerza bruta puro en una técnica inviable, forzando al atacante a migrar hacia vectores más complejos como la Ingeniería Social o el Malware.
- Balance entre Seguridad y Usabilidad: El estudio revela que la seguridad absoluta en protocolos de compartición de archivos como SMB requiere un equilibrio. Mientras más estricta es la directiva (menos intentos, más tiempo de bloqueo), mayor es el riesgo de denegación de servicio accidental para el usuario legítimo, lo que justifica la implementación de capas adicionales como el Doble Factor de Autenticación (MFA).

Para una visualización detallada de la ejecución del ataque y los resultados obtenidos en el laboratorio, consulte el apartado 6 en la memoria práctica.

7.10 Conclusiones

Sinceramente, la fuerza bruta sigue siendo un pilar, no porque sea sofisticada, sino porque es **escalable y económica**. Mientras los usuarios prefieran la comodidad (contraseñas fáciles de recordar) sobre la seguridad, la fuerza bruta seguirá siendo una herramienta fiable.

> **Reflexión técnica:** La fuerza bruta es el "termómetro" de la seguridad de un sistema. Si un ataque de fuerza bruta tiene éxito en 2026, no es un mérito del atacante, sino una negligencia crítica en el diseño de las defensas del sistema (falta de MFA, políticas de contraseñas nulas o falta de límites de tasa).

8. Ataques a Configuración o Infraestructura

8.1.1 Definición de ataques a configuración e infraestructura

Los ataques a configuración o infraestructura son aquellos que aprovechan errores de configuración, vulnerabilidades del sistema o debilidades en la arquitectura tecnológica para comprometer la seguridad de un entorno informático.

A diferencia de otros ataques centrados en el usuario o en aplicaciones concretas, este tipo de amenazas se dirige directamente a los sistemas operativos, servicios, entornos virtualizados, APIs o mecanismos de administración de la infraestructura.

En muchos casos, estos ataques no requieren vulnerabilidades extremadamente complejas, sino simplemente configuraciones inseguras, servicios expuestos innecesariamente o permisos mal gestionados.

Dentro de este bloque se analizarán técnicas como la explotación de vulnerabilidades, la escalada de privilegios, los ataques dirigidos a APIs y las amenazas relacionadas con contenedores y virtualización.

8.1.2 Evolución de los ataques a infraestructura

La evolución de estos ataques ha estado estrechamente ligada al crecimiento de las infraestructuras digitales y la complejidad de los sistemas modernos.

Inicialmente, muchos ataques se centraban en vulnerabilidades conocidas de sistemas operativos o servicios mal configurados. Sin embargo, con el avance de tecnologías cloud, virtualización y contenedores, la superficie de ataque ha aumentado considerablemente.

Actualmente, los atacantes buscan explotar errores de configuración, servicios expuestos, APIs inseguras y fallos de aislamiento entre entornos virtualizados. Además, la automatización y el uso masivo de infraestructuras distribuidas han permitido que este tipo de ataques se realicen a gran escala.

En la actualidad, los ataques a infraestructura representan una de las principales amenazas para organizaciones y servicios críticos debido al impacto que pueden generar sobre sistemas completos.

8.1.3 Objetivos de los ataques a infraestructura

El principal objetivo de estos ataques es obtener acceso no autorizado, aumentar privilegios dentro del sistema o comprometer la disponibilidad y seguridad de la infraestructura tecnológica.

Entre los objetivos más habituales destacan:

- Obtener acceso administrativo
- Escalar privilegios
- Ejecutar código arbitrario
- Comprometer servicios internos
- Acceder a información sensible
- Mantener persistencia en el sistema
- Afectar a la disponibilidad de servicios

En muchos casos, estos ataques constituyen una fase intermedia dentro de campañas más complejas orientadas al espionaje, robo de datos o sabotaje.

8.1.4 Vectores de ataque

Los vectores de ataque más frecuentes relacionados con infraestructura suelen estar asociados a errores humanos, malas configuraciones o vulnerabilidades sin corregir.

Entre los más habituales destacan:

- Servicios expuestos innecesariamente
- Sistemas desactualizados
- Permisos mal configurados
- APIs sin autenticación adecuada
- Contenedores inseguros
- Credenciales débiles o reutilizadas
- Configuraciones por defecto
- Falta de segmentación de red

La complejidad creciente de las infraestructuras modernas incrementa la probabilidad de errores de configuración que puedan ser aprovechados por atacantes.

8.1.5 Público objetivo

Este tipo de ataques suele dirigirse principalmente contra organizaciones, empresas y entornos cloud donde la infraestructura tecnológica tiene un papel crítico.

Los objetivos más frecuentes incluyen:

- Empresas con servicios expuestos a Internet
- Infraestructuras cloud
- Centros de datos
- Plataformas virtualizadas
- APIs públicas
- Sistemas corporativos internos
- Infraestructuras críticas

Sin embargo, también pueden afectar a usuarios domésticos que utilicen sistemas mal configurados o servicios expuestos incorrectamente.

8.2 Explotación de vulnerabilidades (Exploits)

8.2.1 Definición

La explotación de vulnerabilidades consiste en aprovechar fallos o debilidades presentes en sistemas, aplicaciones o servicios para ejecutar acciones no autorizadas dentro de un entorno informático.

Estas vulnerabilidades pueden deberse a errores de programación, configuraciones inseguras, fallos lógicos o software desactualizado. Cuando un atacante consigue aprovechar una de estas debilidades mediante código o técnicas específicas, se habla de un exploit.

Desde el punto de vista de la ciberseguridad, la explotación de vulnerabilidades representa una de las amenazas más importantes, ya que puede permitir acceso no autorizado, ejecución remota de código, escalada de privilegios o compromiso completo de sistemas.

8.2.2 Funcionamiento técnico

El funcionamiento de un exploit depende de la vulnerabilidad que se pretende aprovechar. Generalmente, el atacante identifica primero un fallo existente en el sistema objetivo y posteriormente utiliza una técnica específica para explotarlo.

El proceso suele seguir varias fases:

- Identificación de la vulnerabilidad
- Análisis técnico del fallo
- Desarrollo o utilización de un exploit existente
- Ejecución del exploit sobre el objetivo
- Obtención de acceso o control del sistema

Las vulnerabilidades explotadas pueden encontrarse en:

- Sistemas operativos
- Servicios de red
- Aplicaciones web
- Software de terceros
- APIs
- Controladores o drivers
- Entornos virtualizados

Dependiendo del tipo de vulnerabilidad, el exploit puede permitir desde una simple fuga de información hasta la ejecución completa de código remoto (Remote Code Execution – RCE).

En muchos casos, los atacantes automatizan este proceso utilizando frameworks especializados capaces de identificar y explotar vulnerabilidades conocidas.

8.2.3 Tipos y características

Las vulnerabilidades explotables pueden clasificarse según su naturaleza y el impacto generado.

Remote Code Execution (RCE)

Permite ejecutar código arbitrario de forma remota sobre el sistema vulnerable.

Buffer Overflow

Se produce cuando una aplicación escribe más datos de los que un espacio de memoria puede almacenar, permitiendo modificar el comportamiento del programa.

Local Privilege Escalation

Permite aumentar privilegios dentro de un sistema una vez obtenido acceso inicial.

Zero-Day Exploits

Aprovechan vulnerabilidades desconocidas públicamente o sin parche disponible.

Vulnerabilidades de configuración

Relacionadas con permisos inseguros, servicios expuestos o configuraciones incorrectas.

Características principales

- Aprovechamiento de debilidades técnicas
- Posibilidad de automatización
- Alto impacto potencial
- Dependencia del estado de actualización del sistema
- Riesgo crítico en sistemas expuestos a Internet

8.2.4 Ejemplos reales

Uno de los casos más relevantes fue la vulnerabilidad EternalBlue, utilizada en 2017 por el ransomware WannaCry. Este exploit afectaba al protocolo SMB de Windows y permitió la propagación masiva del malware a nivel mundial.

Otro ejemplo importante es Log4Shell, descubierta en 2021 en la librería Log4j de Java. Esta vulnerabilidad permitía ejecución remota de código y afectó a miles de servicios y aplicaciones en todo el mundo.

También destacan vulnerabilidades críticas en software empresarial, servidores VPN y sistemas cloud que han permitido accesos no autorizados a infraestructuras corporativas.

En muchos casos, los exploits públicos aparecen poco tiempo después de publicarse una vulnerabilidad, aumentando rápidamente el riesgo para sistemas no actualizados.

8.2.5 Impacto y relevancia actual

La explotación de vulnerabilidades continúa siendo una de las principales vías de acceso utilizadas por atacantes y grupos organizados.

Su impacto puede ser extremadamente elevado debido a que una única vulnerabilidad crítica puede comprometer completamente un sistema o infraestructura.

Entre las principales consecuencias destacan:

- Ejecución remota de código

- Robo de información
- Despliegue de ransomware
- Escalada de privilegios
- Movimiento lateral dentro de redes corporativas
- Interrupción de servicios

Además, la velocidad con la que aparecen nuevos exploits y la complejidad de las infraestructuras modernas dificultan enormemente la protección total de los sistemas.

La automatización de ataques y la existencia de repositorios públicos de exploits también han reducido significativamente la barrera técnica necesaria para explotar vulnerabilidades conocidas.

8.2.6 Medidas de prevención y defensa

La protección frente a exploits requiere una estrategia basada en prevención, actualización y monitorización continua.

Entre las principales medidas defensivas destacan:

- Actualización periódica de sistemas y software
- Gestión de parches de seguridad
- Hardening de sistemas
- Eliminación de servicios innecesarios
- Segmentación de red
- Principio de mínimo privilegio
- Uso de firewalls y sistemas IDS/IPS
- Escaneo periódico de vulnerabilidades
- Auditorías de seguridad

Asimismo, resulta fundamental disponer de inventarios actualizados de activos y sistemas expuestos.

8.2.7 Defensa avanzada

Las soluciones avanzadas frente a exploits incluyen mecanismos capaces de detectar comportamientos anómalos incluso cuando la vulnerabilidad todavía no es conocida.

Entre las tecnologías más utilizadas destacan:

- EDR (Endpoint Detection and Response)
- XDR (Extended Detection and Response)
- Sandboxing
- Protección basada en comportamiento
- Inteligencia de amenazas
- Machine Learning aplicado a detección de exploits
- ASLR y DEP en sistemas operativos
- WAF en aplicaciones web

Además, los entornos Zero Trust y la microsegmentación permiten limitar el impacto potencial de una explotación exitosa.

Las organizaciones modernas también implementan programas de gestión de vulnerabilidades y análisis continuo de exposición.

8.2.8 Análisis crítico

La explotación de vulnerabilidades demuestra que ningún sistema puede considerarse completamente seguro. Incluso infraestructuras ampliamente utilizadas y auditadas pueden contener fallos críticos explotables.

Uno de los mayores problemas actuales es la diferencia temporal entre la aparición de una vulnerabilidad y la aplicación efectiva de los parches de seguridad. Durante este periodo, los sistemas permanecen expuestos a ataques potenciales.

Además, la creciente complejidad del software moderno incrementa significativamente la superficie de ataque y dificulta la detección temprana de errores de seguridad.

Desde una perspectiva defensiva, la explotación de vulnerabilidades pone de manifiesto la necesidad de adoptar una estrategia proactiva basada en actualización constante, monitorización continua y reducción de superficie de exposición.

8.3 Escalada de privilegios (Privilege Escalation)

8.3.1 Definición

La escalada de privilegios es una técnica utilizada por atacantes para obtener permisos o privilegios superiores dentro de un sistema informático después de haber conseguido un acceso inicial limitado.

El objetivo principal de este tipo de ataque es aumentar el nivel de control sobre el sistema comprometido, permitiendo acceder a recursos restringidos, ejecutar acciones administrativas o mantener persistencia dentro de la infraestructura.

La escalada de privilegios puede afectar tanto a sistemas Linux como Windows y constituye una de las fases más importantes dentro de muchos ataques avanzados, especialmente en entornos corporativos.

Desde el punto de vista de la seguridad, este tipo de ataque compromete principalmente la integridad y el control del sistema, ya que el atacante consigue capacidades superiores a las originalmente permitidas.

8.3.2 Funcionamiento técnico

El funcionamiento de la escalada de privilegios se basa en aprovechar vulnerabilidades, errores de configuración o permisos inseguros presentes en el sistema objetivo.

Normalmente, el atacante obtiene primero un acceso inicial con privilegios limitados mediante:

- Credenciales comprometidas
- Explotación de vulnerabilidades
- Malware
- Ingeniería social
- Servicios expuestos

Una vez dentro del sistema, el atacante busca mecanismos que le permitan aumentar privilegios.

Entre las técnicas más habituales destacan:

- Explotación de vulnerabilidades locales
- Abuso de permisos incorrectos
- Binarios SUID inseguros en Linux
- Servicios mal configurados
- DLL Hijacking en Windows

- Robo de tokens o credenciales
- Configuraciones inseguras de sudo

El objetivo final suele ser obtener privilegios de administrador o root para controlar completamente el sistema.

8.3.3 Tipos y características

La escalada de privilegios puede clasificarse según el tipo de acceso obtenido.

Escalada vertical

Consiste en aumentar privilegios dentro del mismo sistema, pasando de un usuario estándar a administrador o root.

Escalada horizontal

El atacante obtiene acceso a cuentas o recursos de otros usuarios con el mismo nivel de privilegios.

Escalada mediante vulnerabilidades locales

Se aprovechan fallos del sistema operativo o software instalado para ejecutar código con permisos elevados.

Escalada mediante configuraciones inseguras

Se basa en errores administrativos relacionados con permisos, servicios o políticas de seguridad.

Características principales

- Requiere acceso inicial previo
- Alto impacto sobre el sistema
- Frecuente en ataques avanzados
- Aprovecha errores de configuración y vulnerabilidades
- Permite persistencia y movimiento lateral

8.3.4 Ejemplos reales

Uno de los ejemplos más conocidos fue la vulnerabilidad Dirty COW en Linux, descubierta en 2016, que permitía a usuarios locales obtener privilegios root mediante una condición de carrera en la gestión de memoria.

En sistemas Windows, numerosas vulnerabilidades relacionadas con servicios o controladores han permitido escaladas de privilegios hacia cuentas SYSTEM.

También son frecuentes los casos derivados de configuraciones inseguras, como permisos incorrectos en archivos críticos, scripts ejecutables por otros usuarios o configuraciones sudo mal implementadas.

En entornos corporativos, la escalada de privilegios suele formar parte de ataques más amplios orientados al movimiento lateral y control de dominios completos.

8.3.5 Impacto y relevancia actual

La escalada de privilegios representa una amenaza crítica porque transforma un acceso limitado en un compromiso total del sistema.

Su impacto puede incluir:

- Control completo del sistema
- Instalación de malware persistente
- Robo de credenciales
- Movimiento lateral en redes corporativas
- Desactivación de medidas de seguridad
- Acceso a información sensible

Actualmente, este tipo de ataques sigue siendo extremadamente relevante debido a la gran cantidad de sistemas mal configurados y vulnerabilidades locales descubiertas constantemente.

Además, en muchas ocasiones la escalada de privilegios es la fase que convierte una intrusión menor en un incidente de seguridad grave.

8.3.6 Medidas de prevención y defensa

La protección frente a escalada de privilegios requiere aplicar principios estrictos de seguridad y administración del sistema.

Entre las principales medidas defensivas destacan:

- Aplicación periódica de parches de seguridad
- Principio de mínimo privilegio
- Restricción de permisos innecesarios
- Configuración segura de sudo y grupos administrativos
- Hardening del sistema operativo
- Monitorización de actividad privilegiada
- Segmentación de usuarios y servicios
- Uso de MFA en accesos administrativos
- Auditorías de permisos y configuraciones

Además, resulta fundamental limitar la exposición de cuentas con privilegios elevados.

8.3.7 Defensa avanzada

Las soluciones avanzadas frente a escalada de privilegios incluyen tecnologías capaces de detectar comportamientos anómalos relacionados con el uso de permisos elevados.

Entre las estrategias más utilizadas destacan:

- PAM (Privileged Access Management)
- EDR con análisis de comportamiento
- Detección de exploits locales
- Control de aplicaciones
- Políticas SELinux o AppArmor
- Credential Guard en Windows
- Sandboxing y aislamiento de procesos
- Zero Trust Architecture

Asimismo, la monitorización continua de eventos del sistema permite identificar intentos de abuso de privilegios antes de que el atacante consiga control total.

Las organizaciones modernas también utilizan registros centralizados y SIEM para correlacionar actividades sospechosas relacionadas con cuentas privilegiadas.

8.3.8 Análisis crítico

La escalada de privilegios evidencia que la seguridad de un sistema no depende únicamente de impedir accesos iniciales, sino también de limitar las capacidades disponibles una vez comprometido el entorno.

Muchos incidentes graves no comienzan con privilegios administrativos, sino con accesos limitados que posteriormente son ampliados mediante vulnerabilidades o configuraciones inseguras.

Además, este tipo de ataques demuestra la importancia crítica de la correcta gestión de permisos y la reducción de privilegios innecesarios dentro de cualquier infraestructura.

Desde una perspectiva defensiva, la escalada de privilegios pone de manifiesto la necesidad de adoptar modelos de seguridad basados en segmentación, control estricto de accesos y monitorización continua de actividades privilegiadas.

8.4 Ataques a APIs

8.4.1 Definición

Los ataques a APIs consisten en explotar vulnerabilidades o configuraciones inseguras presentes en interfaces de programación de aplicaciones (APIs) con el objetivo de acceder a información, manipular servicios o comprometer sistemas conectados.

Las APIs permiten la comunicación entre diferentes aplicaciones, servicios y dispositivos, convirtiéndose en un componente fundamental dentro de arquitecturas modernas, entornos cloud y aplicaciones web.

Debido a que muchas APIs manejan autenticación, datos sensibles y funciones críticas, representan un objetivo muy atractivo para los atacantes.

Desde el punto de vista de la seguridad, los ataques a APIs pueden comprometer la confidencialidad, integridad y disponibilidad de los sistemas, especialmente cuando existen errores de autenticación, validación o control de acceso.

8.4.2 Funcionamiento técnico

El funcionamiento de los ataques a APIs se basa en identificar debilidades en la forma en la que la API procesa solicitudes, válida usuarios o expone información.

Los atacantes suelen analizar:

- Endpoints disponibles
- Métodos HTTP utilizados

- Parámetros enviados
- Tokens de autenticación
- Respuestas del servidor
- Control de permisos

Una vez identificadas posibles debilidades, pueden ejecutarse diferentes técnicas de explotación.

Entre las más habituales destacan:

- Acceso no autorizado a endpoints
- Manipulación de parámetros
- Bypass de autenticación
- Exposición excesiva de datos
- Inyección de código
- Enumeración de recursos
- Abuso de tokens o sesiones

En muchos casos, los atacantes automatizan las solicitudes utilizando scripts o herramientas especializadas para analizar grandes cantidades de endpoints de forma rápida.

8.4.3 Tipos y características

Los ataques a APIs pueden clasificarse según el tipo de vulnerabilidad explotada.

Broken Authentication

Se produce cuando la autenticación está mal implementada y permite accesos no autorizados.

Broken Object Level Authorization (BOLA)

El atacante accede a recursos de otros usuarios manipulando identificadores o parámetros.

Excessive Data Exposure

La API devuelve más información de la necesaria, exponiendo datos sensibles.

Rate Limiting Bypass

Se eluden restricciones de peticiones para automatizar ataques o provocar saturación.

Injection Attacks

Incluyen inyecciones SQL, comandos o código mediante parámetros manipulados.

Características principales

- Explotación de lógica de negocio
- Acceso directo a datos y servicios
- Automatización sencilla
- Impacto elevado en entornos cloud
- Dependencia de autenticación y validación correctas

8.4.4 Ejemplos reales

Uno de los problemas más frecuentes en APIs modernas son las vulnerabilidades BOLA, donde usuarios autenticados pueden acceder a información de otras cuentas simplemente modificando identificadores en las solicitudes.

También se han detectado numerosas APIs que exponían datos sensibles debido a configuraciones incorrectas o falta de filtrado en las respuestas.

En aplicaciones móviles, muchos ataques se dirigen directamente a las APIs backend, ya que suelen contener la lógica principal de autenticación y acceso a datos.

Asimismo, servicios cloud y plataformas SaaS han sufrido incidentes relacionados con APIs inseguras que permitieron accesos no autorizados o filtraciones masivas de información.

8.4.5 Impacto y relevancia actual

Los ataques a APIs representan una de las amenazas con mayor crecimiento en la actualidad debido a la enorme dependencia de estas tecnologías en aplicaciones modernas.

El impacto puede incluir:

- Robo de datos sensibles
- Acceso no autorizado a cuentas
- Manipulación de información
- Exposición de infraestructura interna
- Interrupción de servicios
- Compromiso de aplicaciones móviles y cloud

Además, muchas APIs están expuestas directamente a Internet, aumentando significativamente la superficie de ataque.

La adopción masiva de microservicios, aplicaciones móviles y arquitecturas cloud ha convertido las APIs en un objetivo prioritario para atacantes y grupos organizados.

8.4.6 Medidas de prevención y defensa

La protección frente a ataques a APIs requiere asegurar tanto la autenticación como la validación de todas las solicitudes.

Entre las principales medidas defensivas destacan:

- Autenticación robusta mediante OAuth2 o JWT
- Validación estricta de permisos
- Implementación de rate limiting
- Filtrado y validación de parámetros
- Uso de HTTPS obligatorio
- Registro y monitorización de actividad
- Principio de mínimo privilegio
- Segmentación de servicios internos
- Gestión segura de tokens y claves API

Además, resulta fundamental realizar auditorías periódicas y pruebas de seguridad sobre las APIs expuestas.

8.4.7 Defensa avanzada

Las soluciones avanzadas frente a ataques a APIs incluyen mecanismos especializados capaces de detectar comportamientos anómalos y abusos de endpoints.

Entre las tecnologías más utilizadas destacan:

- API Gateway con políticas de seguridad
- WAF especializado en APIs
- Detección de anomalías mediante IA
- Monitorización continua de tráfico API
- Sistemas Zero Trust
- Gestión centralizada de identidades
- Detección automática de abuso de tokens

Asimismo, las organizaciones modernas utilizan herramientas de API Security Posture Management (ASPM) para identificar configuraciones inseguras y endpoints expuestos.

El uso de segmentación y autenticación contextual también reduce significativamente el riesgo de explotación.

8.4.8 Análisis crítico

Los ataques a APIs reflejan cómo la evolución hacia arquitecturas modernas y servicios conectados ha transformado la superficie de ataque tradicional.

Aunque las APIs ofrecen enormes ventajas en interoperabilidad y escalabilidad, también introducen nuevos riesgos relacionados con autenticación, control de acceso y exposición de datos.

Muchos incidentes actuales no se producen por vulnerabilidades complejas, sino por errores lógicos, permisos mal implementados o exposición excesiva de información.

Desde una perspectiva defensiva, los ataques a APIs demuestran la necesidad de considerar la seguridad como un elemento integrado desde el diseño inicial del servicio, aplicando validaciones estrictas, monitorización continua y modelos Zero Trust en todas las comunicaciones.

8.5 Ataques a contenedores y virtualización

8.5.1 Definición

Los ataques a contenedores y entornos de virtualización son aquellos dirigidos contra tecnologías utilizadas para aislar, desplegar y gestionar sistemas y aplicaciones virtualizadas.

Actualmente, tecnologías como Docker, Kubernetes, VMware o Hyper-V son ampliamente utilizadas tanto en infraestructuras empresariales como en entornos cloud. Aunque estos sistemas ofrecen flexibilidad y escalabilidad, también introducen nuevas superficies de ataque y riesgos de seguridad.

El objetivo de estos ataques suele ser comprometer contenedores, escapar del entorno aislado, acceder al host principal o afectar a múltiples sistemas virtualizados desde un único punto vulnerable.

Desde el punto de vista de la seguridad, estos ataques pueden comprometer la confidencialidad, integridad y disponibilidad de infraestructuras completas debido al alto nivel de centralización existente en entornos virtualizados.

8.5.2 Funcionamiento técnico

El funcionamiento de estos ataques depende de la tecnología objetivo y de las vulnerabilidades presentes en la infraestructura virtualizada.

En entornos de contenedores, los atacantes suelen buscar:

- Contenedores mal configurados
- Imágenes vulnerables
- Secretos expuestos
- APIs inseguras
- Permisos excesivos
- Escapes de contenedor

En sistemas de virtualización tradicional, los ataques pueden dirigirse contra:

- Hipervisores vulnerables
- Máquinas virtuales mal aisladas
- Servicios de administración expuestos
- Compartición insegura de recursos

Uno de los escenarios más críticos es el container escape o VM escape, donde el atacante consigue salir del entorno aislado y acceder al sistema host o a otras máquinas virtuales.

Además, muchas veces los atacantes aprovechan errores de configuración relacionados con privilegios elevados, volúmenes compartidos o acceso inseguro a sockets de administración.

8.5.3 Tipos y características

Los ataques a contenedores y virtualización pueden clasificarse según el objetivo comprometido.

Container Escape

El atacante logra salir del contenedor y acceder al sistema host.

Ataques a imágenes vulnerables

Se utilizan imágenes con software desactualizado o malware integrado.

Compromiso de orquestadores

Ataques dirigidos contra Kubernetes o sistemas de gestión de contenedores.

VM Escape

Permite acceder desde una máquina virtual al hipervisor o a otras máquinas virtuales.

Exposición de servicios de administración

Se comprometen paneles, APIs o interfaces de gestión mal protegidas.

Características principales

- Alto impacto potencial
- Riesgo de compromiso masivo
- Dependencia de configuraciones seguras
- Explotación de aislamiento insuficiente
- Objetivos frecuentes en entornos cloud

8.5.4 Ejemplos reales

Uno de los casos más conocidos relacionados con contenedores fue la explotación de APIs Docker expuestas públicamente sin autenticación, permitiendo a atacantes desplegar contenedores maliciosos para minería de criptomonedas.

También se han detectado múltiples incidentes relacionados con imágenes Docker comprometidas publicadas en repositorios públicos.

En entornos Kubernetes, numerosas configuraciones inseguras han permitido accesos administrativos completos a clústeres empresariales.

Por otro lado, vulnerabilidades de VM escape en plataformas de virtualización han demostrado que un fallo crítico puede comprometer múltiples sistemas virtualizados simultáneamente.

La creciente adopción de infraestructuras cloud ha incrementado significativamente la relevancia de este tipo de amenazas.

8.5.5 Impacto y relevancia actual

Los ataques a contenedores y virtualización tienen un impacto especialmente crítico debido al papel central que estas tecnologías desempeñan en infraestructuras modernas.

Las consecuencias pueden incluir:

- Compromiso del sistema host
- Acceso a múltiples servicios simultáneamente
- Robo de secretos y credenciales
- Movimiento lateral dentro de la infraestructura
- Despliegue de malware

- Interrupción de servicios cloud

Además, muchos entornos virtualizados gestionan aplicaciones críticas, bases de datos y servicios empresariales esenciales.

La automatización y escalabilidad de los entornos cloud hacen que un único error de configuración pueda afectar a una gran cantidad de sistemas.

8.5.6 Medidas de prevención y defensa

La protección frente a ataques en entornos virtualizados requiere aplicar medidas estrictas de aislamiento y control de acceso.

Entre las principales medidas defensivas destacan:

- Uso de imágenes verificadas y actualizadas
- Hardening de contenedores y hosts
- Restricción de privilegios innecesarios
- Segmentación de redes internas
- Control de acceso a APIs y paneles administrativos
- Escaneo de vulnerabilidades en imágenes
- Gestión segura de secretos
- Aplicación de parches de seguridad
- Monitorización continua de actividad

Asimismo, resulta fundamental limitar el uso de contenedores privilegiados y evitar configuraciones por defecto inseguras.

8.5.7 Defensa avanzada

Las soluciones avanzadas frente a amenazas en contenedores y virtualización incluyen herramientas especializadas de monitorización y protección de entornos cloud.

Entre las tecnologías más utilizadas destacan:

- Runtime Security para contenedores
- Kubernetes Security Policies
- Zero Trust aplicado a microservicios
- Sandboxing avanzado
- Monitorización de comportamiento
- Escaneo automático de imágenes
- Microsegmentación
- SIEM y EDR integrados

Además, muchas organizaciones implementan herramientas CNAPP (Cloud-Native Application Protection Platform) para proteger infraestructuras cloud modernas.

La automatización de políticas de seguridad y compliance también resulta clave en entornos altamente dinámicos.

8.5.8 Análisis crítico

Los ataques a contenedores y virtualización reflejan cómo la evolución tecnológica ha transformado tanto las capacidades defensivas como las superficies de ataque.

Aunque estas tecnologías ofrecen grandes ventajas en flexibilidad, despliegue y escalabilidad, también introducen riesgos derivados de configuraciones complejas y modelos de administración distribuidos.

En muchos casos, los incidentes de seguridad no se producen por vulnerabilidades extremadamente sofisticadas, sino por errores humanos, exposición de servicios o configuraciones inseguras.

Desde una perspectiva defensiva, este tipo de ataques pone de manifiesto la necesidad de integrar la seguridad desde el diseño inicial de las infraestructuras cloud y virtualizadas, aplicando principios de mínimo privilegio, aislamiento estricto y monitorización continua.

8.6 Simulación Laboratorio

En esta práctica se desarrolló un laboratorio orientado al estudio de vulnerabilidades en APIs, centrándose específicamente en el ataque Broken Object Level Authorization (BOLA), una de las vulnerabilidades más comunes dentro del pentesting de APIs modernas.

Para ello se creó una API vulnerable utilizando Flask, Docker y autenticación mediante JSON Web Tokens (JWT). Durante la simulación se demostró cómo un atacante podía acceder a información de otros usuarios simplemente modificando identificadores dentro de las peticiones HTTP, evidenciando un fallo de autorización en la aplicación.

A lo largo de la práctica se desplegó el entorno mediante contenedores Docker, permitiendo simular de forma aislada y controlada el funcionamiento de la API vulnerable. Posteriormente se implementaron mecanismos de autenticación y control de acceso basados en JWT para mitigar la vulnerabilidad y restringir el acceso únicamente a los recursos pertenecientes al usuario autenticado.

La investigación permitió comprender la diferencia entre autenticación y autorización, así como la importancia de validar correctamente los permisos de acceso en APIs modernas para evitar fugas de información y accesos no autorizados.

Para una visualización detallada de la ejecución del ataque y los resultados obtenidos en el laboratorio, consulte el apartado 5 en la memoria práctica.

8.7 Conclusiones del bloque de ataques a configuración e infraestructura

Los ataques dirigidos a la configuración y a la infraestructura representan una de las amenazas más críticas dentro de la ciberseguridad moderna debido al impacto que pueden generar sobre sistemas completos y servicios esenciales.

A lo largo de este bloque se ha podido observar cómo vulnerabilidades, errores de configuración y malas prácticas administrativas pueden convertirse en puntos de entrada para atacantes, permitiendo desde accesos no autorizados hasta el control total de infraestructuras complejas.

Asimismo, se ha analizado cómo tecnologías actuales como APIs, contenedores y entornos virtualizados han ampliado significativamente la superficie de ataque, introduciendo nuevos riesgos asociados a la automatización, la exposición de servicios y la gestión distribuida de sistemas.

Uno de los aspectos más relevantes es que muchos de estos ataques no requieren técnicas extremadamente avanzadas, sino el aprovechamiento de configuraciones inseguras, sistemas desactualizados o permisos mal gestionados. Esto demuestra la importancia crítica de la administración segura de sistemas y del hardening de infraestructuras.

Desde una perspectiva defensiva, la protección frente a este tipo de amenazas requiere un enfoque integral basado en actualización continua, segmentación, control de privilegios, monitorización constante y modelos de seguridad modernos como Zero Trust.

Finalmente, este bloque evidencia que la seguridad de una infraestructura no depende únicamente de herramientas de protección, sino también de una correcta configuración, mantenimiento y supervisión de todos los componentes del entorno tecnológico.

9. Insider Threats (Amenazas Internas)

9.1 Definición de Insider Threat

Los ataques conocidos como ***insider threats***,son provocados por personas que ya tienen acceso legítimo a los sistemas, redes o información de una empresa, ya sea de manera intencionada o accidental, terminan comprometiendo la seguridad. A diferencia de otros ataques externos, en este caso el peligro proviene desde dentro de la propia organización. El atacante no necesita vulnerar sistemas para entrar, ya que dispone previamente de permisos, credenciales o acceso físico autorizado.Este tipo de amenazas pueden originarse por empleados, ex empleados, administradores, proveedores externos o cualquier persona con acceso interno a recursos corporativos. A muchas empresas estas amenazas internas se consideran uno de los riesgos más difíciles de detectar, debido a que las acciones realizadas suelen parecer legítimas a simple vista.

9.2 Evolución de las Amenazas Internas

Las amenazas internas han existido prácticamente desde el inicio de la informática, pero con el crecimiento de la digitalización y el aumento de información almacenada en los sistemas se aumentó enormemente su impacto.

En los tiempos de antes muchos incidentes internos estaban relacionados con sabotajes físicos o robo de documentación. Hoy en día, un único empleado puede acceder a enormes cantidades de información utilizando únicamente un ordenador conectado a la red corporativa. Además con el teletrabajo y el uso de servicios en la nube han complicado todavía más el control de accesos y la supervisión de actividades sospechosas.

En los últimos años, numerosas empresas han sufrido filtraciones provocadas por trabajadores descontentos, errores humanos o empleados manipulados mediante ingeniería social. La evolución tecnológica también ha aumentado el riesgo.

9.3 Tipos de Insider Threats

Las amenazas internas pueden clasificarse en categorías dependiendo de las intenciones y comportamiento del usuario.

9.3.1 Insider Malicioso

El insider malicioso actúa de forma intencionada con el objetivo de perjudicar a la organización o beneficiarse personalmente.

Estos usuarios pueden:

- robar información confidencial,
- filtrar datos,
- sabotear sistemas,
- vender credenciales,
- o colaborar con grupos externos.

En muchos casos, los motivos suelen estar relacionados con problemas económicos, conflictos laborales o venganza contra la empresa.

9.3.2 Insider Negligente

No todas las amenazas internas son intencionadas. Muchos incidentes ocurren debido a errores humanos o malas prácticas de seguridad.

Algunos ejemplos comunes son:

- utilizar contraseñas débiles,
- compartir credenciales,
- abrir archivos maliciosos,
- perder dispositivos corporativos,
- o enviar información sensible por error.

Aunque no exista mala intención, este tipo de comportamientos puede generar consecuencias muy graves para la organización. Como en anteriores temas, numerosas brechas de seguridad importantes comenzaron simplemente por un descuido humano.

9.3.3 Insider Comprometido

En este caso, el usuario legítimo no actúa voluntariamente, sino que su cuenta o dispositivo ha sido comprometido por un atacante externo.

Esto suele ocurrir mediante:

- phishing,
- malware,
- robo de credenciales,
- o ingeniería social.

El atacante aprovecha los permisos del usuario comprometido para acceder a sistemas internos sin levantar demasiadas sospechas. Esta amenaza es especialmente peligrosa porque muchas veces las actividades parecen realizadas por un usuario autorizado normal.

9.4 Objetivos de una Amenaza Interna

Las amenazas internas pueden tener diferentes objetivos dependiendo de la situación y del atacante involucrado. Uno de los más habituales es el robo de información sensible, como bases de datos, documentos internos, información financiera o propiedad intelectual.

También existen casos relacionados con espionaje empresarial, donde empleados o colaboradores filtran información estratégica a empresas competidoras. Y en otros escenarios, el objetivo puede ser el sabotaje interno, provocando interrupciones de servicio, eliminación de datos o daños en infraestructuras críticas. Por otro lado, algunos incidentes simplemente se producen por negligencia o desconocimiento técnico, sin que exista intención directa de causar daño.

9.5 Factores que Favorecen las Amenazas Internas

Existen varios factores que aumentan considerablemente el riesgo de sufrir incidentes internos dentro de una organización.

Uno de los principales problemas es el exceso de privilegios. En muchas empresas, algunos empleados disponen de más acceso de lo realmente necesario para realizar su trabajo.

La falta de supervisión también representa un riesgo importante. Cuando no existen controles adecuados sobre accesos y actividades, resulta más difícil detectar comportamientos sospechosos.

Otro factor habitual es la escasa formación en ciberseguridad. Muchos usuarios desconocen los riesgos asociados al manejo incorrecto de información sensible o al uso inseguro de dispositivos corporativos.

Además, situaciones como conflictos laborales, estrés o descontento profesional pueden aumentar las probabilidades de comportamientos maliciosos internos.

9.6 Casos Reales de Insider Threats

9.6.1 Caso Edward Snowden

Uno de los casos más conocidos relacionados con amenazas internas fue el protagonizado por **Edward Snowden** en 2013. Snowden trabajaba como contratista para la Agencia de Seguridad Nacional de Estados Unidos (NSA) y tenía acceso autorizado a información altamente confidencial. Aprovechando sus privilegios internos, filtró miles de documentos clasificados relacionados con programas de vigilancia gubernamental. Este incidente generó un enorme impacto político y demostró cómo un único usuario interno puede comprometer información extremadamente sensible.

9.6.2 Caso Tesla

La empresa Tesla sufrió incidentes relacionados con empleados internos que filtraron información confidencial y modificaron sistemas internos de manera no autorizada.Algunos trabajadores compartieron datos corporativos sensibles y código interno, generando problemas de seguridad y reputación para la compañía.

9.6.3 Caso Coca-Cola

En otro caso conocido, un ex empleado de The Coca-Cola Company intentó vender información confidencial relacionada con productos y estrategias empresariales.Aunque finalmente el intento fue detectado. Los antiguos empleados también pueden representar riesgos importantes para una organización.

9.7 Impacto de las Amenazas Internas

Las consecuencias de una amenaza interna pueden llegar a ser extremadamente graves. Las organizaciones pueden sufrir:

- pérdidas económicas
- filtraciones de datos
- daños reputacionales
- interrupciones operativas
- y problemas legales relacionados con privacidad y protección de información.

Además, detectar este tipo de incidentes suele resultar complicado porque el usuario ya posee acceso legítimo al sistema. En muchos casos, las actividades sospechosas pueden confundirse fácilmente con tareas normales realizadas por empleados autorizados.

9.8 Medidas de Prevención y Defensa

La prevención frente a amenazas internas requiere combinar tecnología, supervisión y formación continua.Una de las medidas más importantes es aplicar el principio de mínimo privilegio, es decir, conceder únicamente los accesos estrictamente necesarios para cada usuario.También resulta fundamental monitorizar actividades sospechosas mediante sistemas de detección y análisis de comportamiento.

La autenticación multifactor ayuda a reducir riesgos relacionados con cuentas comprometidas o robo de credenciales.Además, muchas empresas realizan auditorías internas, controles de acceso y programas de concienciación en ciberseguridad para disminuir errores humanos.En algunos entornos especialmente sensibles, también se utilizan sistemas DLP (*Data Loss Prevention*) capaces de detectar intentos de extracción de información confidencial.

9.9 Relevancia Actual

Actualmente, las amenazas internas representan una preocupación creciente para empresas y organismos públicos. El aumento del teletrabajo, el uso de servicios cloud y la enorme cantidad de información digital almacenada han incrementado considerablemente el riesgo.

Además, muchos ataques modernos combinan amenazas externas con usuarios internos comprometidos mediante phishing o malware.

Por este motivo, las organizaciones ya no solo deben protegerse frente a atacantes externos, sino también controlar adecuadamente los riesgos asociados a usuarios con acceso legítimo.

9.10 Simulación Laboratorio

En esta práctica se simuló un escenario de Insider Threat, representando el caso de un empleado descontento que, aprovechando los permisos legítimos otorgados por la empresa, accede y extrae información confidencial de la organización.

Durante la simulación se mostró cómo un usuario interno podía copiar documentos sensibles relacionados con finanzas, datos de empleados y archivos corporativos hacia un dispositivo USB externo sin necesidad de utilizar técnicas avanzadas de hacking. El objetivo principal fue demostrar que muchas amenazas de seguridad no provienen únicamente de atacantes externos, sino también de usuarios autorizados que abusan de sus privilegios de acceso.

La práctica permitió observar cómo una fuga de información interna puede generar consecuencias graves para una empresa, incluyendo pérdidas económicas, daños reputacionales y exposición de datos sensibles. Además, se analizó la importancia de aplicar medidas defensivas como controles de

acceso, restricciones sobre dispositivos USB, monitorización de actividad y sistemas de detección de comportamiento sospechoso para reducir el riesgo frente a amenazas internas.

Para una visualización detallada de la ejecución del ataque y los resultados obtenidos en el laboratorio, consulte el apartado 7 en la memoria práctica.

9.11 Conclusión

Las amenazas internas demuestran que la seguridad informática no depende únicamente de protegerse frente a atacantes externos. En muchas ocasiones, el mayor riesgo puede encontrarse dentro de la propia organización. Ya sea por negligencia, errores humanos o acciones maliciosas, un usuario con acceso legítimo puede llegar a comprometer sistemas críticos e información sensible con relativa facilidad.La formación, la supervisión continua y una correcta gestión de privilegios se han convertido en elementos esenciales para reducir este tipo de riesgos.

10 Conclusiones Finales

10.1 Conclusiones generales del proyecto

Como conclusiones para nuestro proyecto de investigación, Hemos podido analizar una visión de la ciberseguridad que no teníamos, y es que no importa que tan excelente técnico seas o cuantas defensas pongas en tu sistema, si al final cualquier usuario de tu organización o empresa comete un error, decide vengarse, o es simplemente atacado, todas tus defensas caerán y perjudicarán todo tu trabajo.

Además de conocer métodos de ataque y defensas para los sistemas, conocer más sobre redes e infraestructuras a nivel de profesional. Creemos que tenemos un nivel un poco más avanzado a la hora de explicar a un usuario los peligros de la ciberseguridad, somos más conscientes de lo peligroso que puede ser dar tanta libertad a un usuario sin conocimientos en informática.

En el desenvolvimiento del proyecto también se creó un interés mayor del que ya teníamos, Teniendo una idea de lo que puede ser estudiar o trabajar en un entorno de ciberseguridad, Habiendo ciertos temas más interesantes que otros.

Ataques, Defensas más complicados de realizar, Concluimos el proyecto sabiendo que la seguridad en la informática será un pilar fundamental siempre.

10.2 Consecución de objetivos

Los objetivos planteados al inicio del proyecto han sido mayoritariamente alcanzados de forma satisfactoria, tanto a nivel técnico como formativo. A lo largo del desarrollo del trabajo se ha conseguido investigar, comprender y simular diferentes tipos de ataques utilizados actualmente dentro del ámbito de la ciberseguridad, siempre en entornos controlados y seguros. Sin embargo, en el apartado de defensas, no se han realizado tantas como estaban previstas.

Uno de los principales objetivos era recrear escenarios realistas de ataque y defensa que permitieran demostrar de forma práctica el funcionamiento de amenazas reales. Este objetivo se ha cumplido mediante la creación de laboratorios virtualizados utilizando máquinas Ubuntu, Kali Linux y Windows, conectadas en redes aisladas mediante VirtualBox. Gracias a esta infraestructura se han podido realizar simulaciones de ingeniería social, malware, ataques de red y técnicas relacionadas con la explotación de sistemas y servicios. El único ataque no se ha llevado a la práctica debido a su complejidad y falta de tiempo es el de web.

Además de la parte ofensiva, también se han aplicado medidas defensivas y mecanismos de detección con el objetivo de comprender cómo prevenir o mitigar este tipo de amenazas. Esto ha permitido no solo ejecutar ataques controlados, sino también analizar sus consecuencias y estudiar posibles soluciones de protección.

10.3 Valoración de la metodología y planificación

Al empezar el proyecto teníamos una aplicación web, llamada **clickup**, que ya hemos mostrado anteriormente, con esta organizamos el proyecto y dividimos los ataques en semanas y fechas límites, Pero habian semanas o ataques que se iban complicando más de lo esperado. Por lo que tuvimos que hacer recortes de ciertos temas e incluso dejando de lado ciertos ataques interesantes por la complejidad de estos y el tiempo que estos nos robarían, decidimos desarrollar mejor algunos ataques mas interesantes y profundizarlos mas para asi llegar a la fecha límite y entregar exitosamente el proyecto.

Somos conscientes de que no realizamos una buena optimización del tiempo por lo que hubieron puntos y un ataque que no pudimos desarrollar mejor, por ello valoramos que no fuimos lo suficientemente realistas de lo complejo que pueden ser los temas desconocidos y el tiempo que estos nos consumieron.

10.4 Visión de futuro

Como posible ampliación futura del proyecto, se plantea el desarrollo de una plataforma web educativa orientada al aprendizaje práctico de ciberseguridad en entornos controlados y gamificados.

La idea principal consistiría en crear un sistema interactivo donde los usuarios puedan aprender las técnicas estudiadas durante este proyecto mediante documentación, simulaciones y desafíos prácticos organizados por categorías de ataque y defensa.

La plataforma estaría basada en un modelo competitivo tipo “1 vs 1”, donde dos participantes asumirán roles diferentes:

- Equipo atacante.
- Equipo defensor.

Antes de cada partida, ambos equipos dispondrán de un tiempo limitado de preparación durante el cual podrían consultar documentación técnica, estudiar vulnerabilidades y configurar sus estrategias ofensivas o defensivas utilizando los conocimientos aprendidos previamente.

Una vez finalizado el tiempo de preparación, comenzaría la simulación práctica dentro de un entorno aislado y controlado. El objetivo del atacante sería comprometer el sistema vulnerable mediante distintas técnicas de explotación, mientras que el defensor tendría que detectar, bloquear o mitigar el ataque utilizando configuraciones de seguridad, monitorización y medidas defensivas.

Las partidas podrían organizarse mediante un sistema al mejor de tres rondas, permitiendo alternar escenarios, vulnerabilidades y técnicas utilizadas en cada enfrentamiento. Además, la plataforma podría incluir:

- Sistema de puntuación.
- Rankings de jugadores.
- Historial de partidas.
- Registro de eventos y ataques.
- Laboratorios virtuales automatizados.
- Retos progresivos según dificultad.

11. Webgrafia

11.1 Ingeniería Social

Instituto Nacional de Ciberseguridad (INCIBE). (2025). *¿Qué es el phishing?* Recuperado el 15 de enero de 2026, de https://www.incibe.es/aprendeciberseguridad/phishing

Panda Security. (2025). *Qué es el phishing y cómo protegerse*. Recuperado el 15 de enero de 2026, de https://www.pandasecurity.com/es/mediacenter/phishing/

Computer Hoy. (2023). *¿Sabes qué es INGENIERÍA SOCIAL?* [Vídeo]. YouTube. Recuperado el 16 de enero de 2026, de https://www.youtube.com/watch?v=PN0j35dbG-8

Kaspersky. (2025). *Qué es el spear phishing*. Recuperado el 21 de enero de 2026, de https://www.kaspersky.es/resource-center/definitions/spear-phishing

IBM. (2025). *Spear phishing*. Recuperado el 17 de enero de 2026, de https://www.ibm.com/es-es/topics/spear-phishing

Píldoras Informáticas. (2024).*¿Qué es… SPEAR PHISHING?* [Vídeo]. YouTube. Recuperado el 17 de enero de 2026, de https://www.youtube.com/watch?v=EBZNIeO9BuI

Kaspersky. (2025). *Qué es la ingeniería social*. Recuperado el 17 de enero de 2026, de https://www.kaspersky.es/resource-center/definitions/what-is-social-engineering

INCIBE.(2026). *Pretexting* https://www.incibe.es/aprendeciberseguridad/pretexting

Kaspersky. (2025). *Qué es el baiting*. Recuperado el 20 de enero de 2026, de https://www.kaspersky.es/resource-center/definitions/what-is-baiting

Red Seguridad. (2026). Baiting: *qué es y cómo evitar este ataque informático de ingeniería social,* Recuperado el 25 de enero de 2026, dehttps://www.redseguridad.com/actualidad/cibercrimen/baiting-que-es-y-como-evitar-este-ataque-informatico-de-ingenieria-social_20210628.html

Delta protect. (2025). *Tailgating: ¿qué es y cómo puedes evitar que afecte a tu empresa?,* Recuperado el 25 de enero de 2026, de https://www.deltaprotect.com/blog/tailgating-que-es

11.2 Malware

Explicado en minutos (2024), *Todos los Virus Informáticos en 8 Minutos,* Recuperado el 30 de enero de 2026, de https://www.youtube.com/watch?v=WjBrRvqpw1Y

Computer Hoy (2021), *¿Qué es malware? Los principales tipos de ataques informáticos y cómo protegernos ante ellos.* Recuperado el 30 de enero de 2026, de https://www.youtube.com/watch?v=HuasitV4lcw

Microsoft. (s.f.). *¿Qué es el malware?* Recuperado el 2 de febrero de 2026, de https://www.microsoft.com/es-es/security/business/security-101/what-is-malware

Kaspersky. (s.f.). *¿Qué es un troyano?* Recuperado el 3 de febrero de 2026, de https://www.kaspersky.es/resource-center/threats/trojans

Microsoft. (s.f.). *¿Qué es el ransomware?* Recuperado el 3 de febrero de 2026, de https://www.microsoft.com/es-es/security/business/security-101/what-is-ransomware

Fortinet (2025), *¿Qué es un virus de gusano informático?* Recuperado el 3 de febrero de 2026, de https://www.fortinet.com/lat/resources/cyberglossary/worm-virus

Fortinet (2025),*¿Qué es el spyware?* Recuperado el 10 de febrero de 2026, de https://www.fortinet.com/lat/resources/cyberglossary/spyware

Fortinet (2025), *¿Qué es un rootkit?* Recuperado el 15 de febrero de 2026, de

https://www.fortinet.com/lat/resources/cyberglossary/rootkit

11.3 Ataques de red

Fortinet (2025), *¿Qué es un ataque DDoS y cómo evitarlo?* Recuperado el 18 de febrero de 2026, https://www.arsys.es/blog/que-es-un-ataque-ddos-y-como-evitarlo?itc=LSOP2E4O-AFTQ4U-QCDI73Z&gclsrc=aw.ds&&acp=23163238293&avl=|||&utm_campaign=SGE-ES-ECO-ECOX-PMX-----Arsys&utm_source=google&utm_medium=cpc&gad_source=1&gad_campaignid=23163281274&gclid=Cj0KCQjwzqXQBhD2ARIsAKrIeU_1mBM8PWptUus_WxL3wYIo8Xg1X73k89utDx0wg_y50n_rNtx4DfoaAuwXEALw_wcB

Wondershare (2026) *¿Qué es un DDoS? Qué hacer en caso de un ataque* Recuperado el 18 de febrero de 2026 https://recoverit.wondershare.es/windows-computer-tips/what-is-ddos.html?gad_source=1&gad_campaignid=23635106601&gclid=Cj0KCQjwzqXQBhD2ARIsAKrIeU-BWPtBrV1mrOvpNoSZ20faCIT6CvHdZZ8yTrhNjP2oZT1xcDk92rwaAkRsEALw_wcB

Red Seguridad (2023), *Sniffing: ¿qué es y cómo podemos evitarlo?* Recuperado el 20 de febrero de 2026 *https://www.redseguridad.com/actualidad/sniffing-que-es-como-evitar_20230713.html*

Arsys (2024), *¿Qué es el spoofing y cómo prevenir un ataque?* Recuperado el 20 de febrero de 2026 *https://www.arsys.es/blog/que-es-el-spoofing-y-como-prevenir-un-ataque?itc=LSOP2E4O-FAC1E9-5RLVJF2&gclsrc=aw.ds&&acp=23552274436&avl=|||&utm_campaign=SGE-ES-MYW-MYWX-PMX-----Arsys&utm_source=google&utm_medium=cpc&gad_source=1&gad_campaignid=23552277313&gclid=Cj0KCQjwzqXQBhD2ARIsAKrIeU-L0aMnKG7GNOM4-DXwUlnHS29FZtTB7xvGjHVRUbvdRHkxAOSph-IaAnDyEALw_wcB*

INCIBE(2020), *El ataque del "Man in the middle" en la empresa, riesgos y formas de evitarlo,* Recuperado el 23 de febrero de 2026 https://www.incibe.es/empresas/blog/el-ataque-del-man-middle-empresa-riesgos-y-formas-evitarlo

IBM (2025) *¿Qué es un ataque de intermediario (MITM)?* Recuperado el 4 de marzo de 2026 https://www.ibm.com/es-es/think/topics/man-in-the-middle

OptimaWeb (2026), *Hijacking: qué es y cómo prevenir estos ataques informáticos*, Recuperado el 6 de marzo de 2026 https://www.optimaweb.es/hijacking-que-es-y-como-prevenir-ataques/

11.4 Ataques de Aplicaciones web

Akamai (2026) *¿Qué es un ataque a aplicaciones web?* Recuperado el 9 de marzo de 2026 https://www.akamai.com/es/glossary/what-is-a-web-application-attack

Varela Alexis (2022) *ATAQUES A APLICACIONES WEB Y SUS TIPOS - ALEXIS VARELA* [video], Recuperado el de 9 marzo de 2026 https://www.youtube.com/watch?v=_Tdc5GTUW8A

Fortinet (2026) *¿Qué es la inyección SQL?* Recuperado el de 11 marzo de 2026 https://www.fortinet.com/lat/resources/cyberglossary/sql-injection

Fortinet (2026) *¿Qué es el scripting entre sitios (XSS)?* Recuperado el de 13 marzo de 2026 https://www.fortinet.com/lat/resources/cyberglossary/cross-site-scripting

Fortinet (2025),*¿Qué es CSRF (Falsificación de solicitudes entre sitios)?* Recuperado el de 4 marzo de 2026 https://www.fortinet.com/lat/resources/cyberglossary/csrf

Rinku (2025), *Path Traversal: Vulnerabilidad, ejemplos reales y explotación* Recuperado el de 16 marzo de 2026 https://www.youtube.com/watch?v=eY_gfuZHsQU

11.5 Ataques de fuerza bruta

El Pingüino de Mario (2024), *TODO lo que Tienes que Saber sobre los ATAQUES de FUERZA BRUTA - En MENOS de 15 MINUTOS* [video] Recuperado el de 23 marzo de 2026 , https://www.youtube.com/watch?v=6jM00NlRI9c

La Guía Informática (2025) *¿Qué es Ataque de Fuerza bruta?* | *Tipos de Fuera bruta* | *Como funciona la fuerza bruta* [video] Recuperado el de 26 marzo de 2026 https://www.youtube.com/watch?v=IuYHbwV0qPI

Campus de Ciberseguridad (2026) *¿Cómo mitigar ataques de fuerza bruta?* [video] Recuperado el 30 de marzo de 2026 https://www.youtube.com/watch?v=UwAHlPyTZGs

11.6 Ataques a Configuración o infraestructura

Ergo, Hackers (2026). *IaC ¿La clave para una infraestructura más segura o una puerta abierta a los ataques?* [video] Recuperado el 3 abril de 2026 https://www.youtube.com/watch?v=b1K2V9b1RFA

Bitdefender (2026) *¿Que es un Exploit? Prevención de Exploits,* Recuperado el 6 abril de 2026 https://www.bitdefender.es/consumer/support/answer/22884/

IBM (2025) *¿Qué es la escalada de privilegios?* Recuperado el 6 abril de 2026

https://www.bitdefender.es/consumer/support/answer/22884/

Akamai (2025), *¿Qué son los ataques a las API?* Recuperado el 10 abril de 2026 https://www.akamai.com/es/glossary/what-are-api-attacks

Press Any Key (2021) *Máquinas Virtuales y Contenedores - ¿Qué son? ¿A qué huelen? ¿Para qué sirven?* Recuperado el 15 abril de 2026
https://www.youtube.com/watch?v=mzo2OjcSxag

11.7 Insider Threats

INCIBE (2023) *Insiders: cómo atacan desde dentro,* Recuperado el 20 abril de 2026 https://www.incibe.es/empresas/blog/insiders-como-atacan-desde-dentro

Red Seguridad (2024) *Qué es el Ciberataque del Enemigo Interno o 'Insider Threat',* Recuperado el 23 abril de 2026 https://www.redseguridad.com/actualidad/que-es-el-ciberataque-del-infiltrado-o-insider-threat_20230213.html

Microsoft(2023) *¿Qué es la amenaza interna?* Recuperado el 29 abril de 2026 https://www.microsoft.com/es-es/security/business/security-101/what-is-insider-threat