CAIGG

CONSEJO DE AUDITORÍA INTERNA GENERAL DE GOBIERNO

Ministerio Secretaría General de la Presidencia

"Guía GASIC N° 13" - Noviembre 2025

Guía de Auditoría, Seguridad de la Información y Ciberseguridad N°13

BUENAS PRÁCTICAS PARA LA AUDITORÍA A LA CIBERSEGURIDAD

ÍNDICE

Nota

PRESENTACIÓN

En cumplimiento con las instrucciones del Presidente de la República, Gabriel Boric Font, sobre fortalecimiento de la Política de Auditoría Interna de Gobierno; el Consejo de Auditoría Interna General de Gobierno, entidad asesora en materias de auditoría interna, control interno, probidad, gestión de riesgos y gobernanza del Supremo Gobierno, presenta a la Red de Auditoría Gubernamental, la GASIC N°13: Buenas Prácticas para la Auditoría a la Ciberseguridad.

Esta guía forma parte de una iniciativa del CAIGG orientada a fortalecer las competencias de la función de auditoría interna del sector público en materia de Seguridad de la Información y Ciberseguridad. Su propósito es proporcionar a los Auditores Internos y a los Servicios Públicos herramientas e instrumentos técnicos que les permitan analizar y evaluar los sistemas de información conforme a las mejores prácticas internacionales y a la legislación vigente.

Santiago, noviembre 2025.

Daniela Caldana Fulss
Auditora General de Gobierno

Nota

INTRODUCCIÓN

El objetivo del presente handbook es ejemplificar qué hacer y cómo reportar hallazgos de auditoría en el ámbito de la ciberseguridad, entregando lineamientos prácticos que orienten la planificación, ejecución y documentación de auditorías técnicas, sin constituir un manual exhaustivo para el diseño de pruebas de auditoría. Las pruebas que se presentan a lo largo del documento tienen un carácter ilustrativo y didáctico, y su finalidad es apoyar la comprensión del proceso de auditoría, más que servir como procedimientos prescriptivos.

Esta guía busca además promover una visión integral de la auditoría a la ciberseguridad, entendida no solo como la verificación de controles técnicos, sino también como un ejercicio de evaluación de madurez, gestión de riesgos y gobernanza tecnológica. En este sentido, se articula con marcos de referencia como ISO/IEC 27001:2022, NIST Cybersecurity Framework (CSF), CIS Controls v8, y la Ley N° 21.459 sobre Ciberseguridad de las Infraestructuras Críticas de la Información, entre otros instrumentos relevantes para el contexto nacional.

La estructura de la guía se organiza en cuatro capítulos que recorren las principales etapas del proceso de auditoría:

1. Preparación y Alcance – Define los elementos formales y técnicos necesarios para iniciar la auditoría, estableciendo los límites, recursos y objetivos del ejercicio.
2. Diseño de Pruebas – Describe la metodología y técnicas empleadas para planificar las pruebas, los criterios de éxito y los métodos de obtención de evidencia.
3. Ejecución de Pruebas – Explica las fases operativas del ejercicio de auditoría, siguiendo un enfoque secuencial inspirado en la Cyber Kill Chain.
4. Confección del Reporte – Entrega directrices para la documentación de hallazgos, la trazabilidad de evidencias y la redacción del informe final.

De esta forma, la GASIC N°13 constituye una herramienta de apoyo para los auditores internos del sector público, contribuyendo a fortalecer las competencias en ciberseguridad, la gestión proactiva de riesgos tecnológicos y la mejora continua del control interno gubernamental.

Capítulo 1

PREPARACIÓN Y ALCANCE

Capítulo 1

1.1 PREPARACIÓN

El profesional que se esté preparando para realizar el ejercicio de auditoría a la ciberseguridad debe considerar dos tipos de pre-requisitos: los asociados a la **capacidad técnica** necesaria para ejecutar las pruebas, y los que guardan relación con la **formalidad del ejercicio** de auditoría interna.

Los requisitos formales del ejercicio de la auditoría podrían variar dependiendo de los requisitos de cumplimiento y estándares adoptados por cada organización, pero como mínimo consideran:

- **A) Seleccionar el equipo auditor y el auditor líder.**
- **B) Diseñar el programa de auditoría con intervalos de tiempo suficiente para analizar los problemas encontrados y diseñar las acciones correctivas.**
- **C) Diseñar el plan de auditoría .**
- **D) Establecer los recursos humanos y tecnológicos, así como los documentos que resultan necesarios para el ejercicio de la auditoría.**
- **E) Definir el alcance de la auditoría (punto siguiente).**
- **F) Establecer los mecanismos de comunicación con la dirección.**

Preparación de la auditoría interna: Detalles

Los requisitos descritos en los incisos A) al E) del punto 1.1dan cumplimiento a los requisitos de la norma ISO 27001:2023 "Sistema de Gestión de Seguridad de la información" punto 9.2, sin embargo, cada organización puede requerir de ciertos elementos específicos.

Aunque no es propósito de esta guía desarrollar los requisitos formales de una auditoría, a continuación, se listan algunos documentos útiles que permiten al lector obtener mayores detalles en estas áreas:

-Documento Técnico Nº90:

Modelo Integral de Auditoría Interna de Gobierno.

-Documento Técnico Nº118:

Plan Anual del Trabajo en Auditoría Interna.

Los prerrequisitos de capacidad técnica se tratan de los instrumentos y artefactos que serán requeridos para llevar a cabo las pruebas.

Los elementos de software más utilizados son: de descubrimiento y mapeo de red, escáneres de vulnerabilidades, frameworks para la explotación controlada (como Metasploit o Cobalt), captura y análisis de tráfico, Instrumentos para el análisis forense, de revisión de código y seguridad aplicativa, de análisis para la nube, y para el análisis de parámetros criptográficos.

Los elementos de hardware y físicos tienen un espectro amplio, desde estaciones portátiles especializadas para el desarrollo de una auditoría o prueba de penetración, pasando por sondas, taps para las figuras, adaptadores USB con carga útil, analizadores de protocolos industriales, bloqueadores de señal, etc.

Los requisitos técnicos se asocian directamente con el alcance del ejercicio, pues cada componente de la infraestructura tecnológica requerirá de un set de herramientas específicas. Esto no significa que no sea posible realizar un buen ejercicio utilizando herramientas open-source.

Para poder demostrar ejemplos prácticos, esta guía utilizará los siguientes instrumentos técnicos:

VMWare (de Broadcom) como Hipervisor, es un freeware para uso personal. Otras alternativas son Proxmox VE o VirtualBox

Kali Linux como suite en sistema operativo. Aunque usted puede utilizar otra distribución del este SO, o incluso trabajar en entornos Windows, Kali Linux es conocido por ser amigable al usuario y una buena puerta de entrada. Viene precargado con herramientas para todo tipo de pruebas. .

1.2 ALCANCE DE LA AUDITORÍA A LA CIBERSEGURIDAD

1.2.1 Auditoría a la Ciberseguridad, Auditoría a la Seguridad de la Información y Pentesting:

Aunque estos términos son similares, existen sutiles diferencias que delimitan sus objetivos. Mientras que la Auditoría a la Ciberseguridad tiene un foco en los controles que protegen la infraestructura digital (o ciberespacio, de allí "ciber" seguridad); la Auditoría a la Seguridad de la Información contempla todo el Sistema de Gestión de Seguridad de la Información, incluidos controles que van más allá del ciberespacio, como políticas, personas y procesos. Por otro lado, el pentesting tiene como principal foco identificar los vectores de ataque reales y demostrar su impacto en los activos críticos de la organización.

Característica	*Auditoría de Ciberseguridad*	*Auditoría de Seguridad de la Información*	*Pentesting*
Enfoque Principal	Controles técnicos que protegen la infraestructura digital: (Redes, endpoints, nube, OT/IoT).	Todo el Sistema de Gestión de Seguridad de la Información: políticas, procesos, personass tecnología y entorno físico.	Simulación controlada de ataques para descubrir y explotar vulnerabilidades técnicas.
Objetivo	Verificar que los mecanismos de defensa cibernética funcionan y están alineados con el riesgo (ej. CIS Controls, NIST CSF).	Determinar el grado de conformidad con marcos de gestión (ISO 27001/27002, normas sectoriales) y la eficacia global del SGSI.	Identificar vectores de ataque reales y demostrar su impacto potencial en los activos priorizados.
Entregables	Informe técnico con hallazgos clasificados, plan de prioridades y métricas de madurez ciber.	Informe de conformidad, no conformidades, acciones correctivas y grado de eficacia del SGSI.	Informe de vulnerabilidades explotadas, rutas de ataque, pruebas de concepto y recomendaciones de mitigación.

Considerando lo anterior, este handbook tiene como enfoque principal ejemplificar el proceso a través del cual un auditor puede evaluar el grado en que los controles técnicos protegen los activos digitales al responder a los riesgos detectados y los incidentes previos y reportados por la comunidad.

Al reportar la principal diferencia entre la auditoría a la ciberseguridad2 y otros métodos de auditoría es el volumen de información, el nivel de especificidad técnica (que incluyen no solo detalles de las pruebas si no también correlaciones con métodos de análisis adicionales a cada prueba) y la variabilidad de la aplicación de los controles, que puede tener una rotación diaria (por ejemplo, las reglas de las listas de control de acceso -ACL- podrían actualizarse de forma diaria).

Algunos de estos datos específicos pueden ser:

- **Puntaje de la vulnerabilidad (si la hay) asociada al hallazgo en formato CVE / CVSS .**
- **Mapeo a índices de controles.**
- **Referencias a marcos de ataques (ATT&CK).**
- **Especificación del parámetro de configuración del control.**
- **Etc.**

1.2.2 Delimitando el alcance de la auditoría:

El alcance define "que" revisar, se expresa en límites concretos que buscan minimizar el riesgo de interrupción de la operación, el daño a los activos o compromisos legales. En ocasiones (especialmente si se trata de auditorías externas) se complementa con un documento denominado Reglas de Compromiso (ROE, por sus siglas en inglés) que define el "como" llevar a cabo la revisión.

La definición del **alcance** se hace estableciendo límites temporales (por ejemplo, revisar logs desde que a que fecha), de negocio (por ejemplo, sobre que unidad organizativa o sobre que segmento de subred), o físico (por ejemplo, los activos presentes en un determinado edificio). Es aceptable también un alcance basado en la clasificación de los activos (por ejemplo, sobre los activos críticos y los expuestos a riesgos de mayor severidad, rango de IPs o listas de dominos y subdominios).

Las **reglas de compromiso** por su parte establecen los principios de legitimidad del ejercicio, lista las acciones permitidas y prohibidas, las ventajas de ejecución de las pruebas técnicas, los puntos de contacto para notificar incidentes, criterios para detener las pruebas (por ejemplo, caía de un servicio crítico o consumo extremo de recursos) , la gestión de evidencias, criterios de disposición de datos (retención y destrucción tras la auditoría) y el plan de contingencia (como rollbacks, contactos de soporte y SLA internos).

Las ROE no están presentes en toda auditoría interna. Son, principalmente, un instrumento para controlar a las entidades externas que entregan servicios de auditorías o pentesting. Sin embargo, dado el nivel de especificada de la información y su potencial para minimizar los riesgos del ejercicio, pueden ser utilizadas como un instrumento de sincronización interna.

Ejemplo de Alcance: Auditoría al SII	
Ámbito Organizacional	Dirección de Tecnología, Subdirección de Operaciones y unidades regionales que alojan servicios críticos.
Sistemas y Activos	- Portales sii.cl y Oficina Virtual (Todos los subdominios encontrables) - Plataforma de Declaracion de Renta - Servicios expuestos en nube publica (AWS) - Red corporativa y data center central (Santiago) - Bases de datos Oracle tributarias - Estaciones de trabajo del personal de fiscalizacion.
Controles y Marcos de Referencia	-Ley 21459, -Decreto 83/2021 (Seguridad de la Informacion del Estado), -ISO 27001:2022 -Controles A.5 a A.18, -CIS Controls v8, -NIST CSF.
Periodo Revisado	Configuraciones, registros y evidencias comprendidos entre 01 - 01 - 2025 y 31 - 03 - 2025
Profundidad de Pruebas	- Revisión documental del SGSI - Escaneo de vulnerabilidades autenticado - Pruebas de intrusion controladas sobre red interna y portal publico - Revision de configuraciones de AWS (IAM, VPC, S3) - Muestreo de politicas de acceso fisico en data center.
Exclusiones	Equipos personales de funcionarios, servicios legacy en desuso, redes OT de climatizacion.

Capítulo 2

DISEÑO DE PRUEBAS

2 DISEÑO DE PRUEBAS

2.1 Introducción

Las pruebas de auditoría se pueden agrupar en "secciones" que siguen metodologías probadas, como el CKC (Cyberkillchain). Esto es especialmente útil por que algunas pruebas sólo tienen sentido si es posible ejecutar con éxito la prueba anterior (y sólo si arroja un hallazgo). Por ejemplo, antes de revisar si una casa tiene cámaras al interior del comedor, es necesario primero comprobar que la puerta no se abre sin la llave. Si la puerta se abre, entonces recién puedo observar si hay o no hay una cámara en el living.

Entonces, para poder revisar la cámara al interior del comedor necesito cumplir con al menos una de dos condiciones: A) Ser capaz de identificar que existe esa cámara, atravesando la puerta o B) Tener la información desde antes de que existe tal cámara, y, además, poseer una llave para entrar por la puerta si es que no puedo atravesarla por mis propios medios.

El primer método, donde tengo que abrir yo mismo la puerta y buscar la cámara, se conoce como "caja negra" y es útil para poder ponerse en los zapatos de un adversario y evaluar que tan difícil sería penetrar el sistema, mientras que el segundo método, cuando tengo toda la información, se conoce como "caja blanca" y permite evaluar cada componente de la arquitectura de seguridad con independencia de otros controles.Por lo tanto, organizar las pruebas en "fases" como propone CKC es tanto útil para categorizar las pruebas en sí, como para planificar la ejecución de pruebas cuando se usa el método de caja negra.

Note que, en el informe final, puede presentar las pruebas y sus hallazgos en cualquier orden. Siempre se recomienda presentar primero las pruebas que arrojan hallazgos de mayor severidad, para aprovechar al máximo el tiempo de atención del receptor del informe.

Solo a modo de ilustración, a continuación, se muestran los 7 pasos de la CKC que sirven como referencia para agrupar las pruebas. En realidad, no todos los pasos son replicados, y suelen utilizarse categorías mas genéricas como: Reconocimiento (pasos del 1 al 2), Explotación (pasos del 3 al 5), Post Explotación que incluye el escalamiento de privilegios en los sistemas (pasos del 5 al 7).

No toda prueba debe terminar necesariamente en una explotación. Siempre debe evaluar el daño potencial de explotar una vulnerabilidad y verificar si está dentro del alcance y las ROE

Paso de Cyber Kill Chain		*Ejemplos de Técnicas*
1. Reconnaissance (Reconocimiento)	Recopilación de información sobre la organización, usuarios y sistemas antes del ataque.	Búsqueda OSINT sobre dominios y subdominios corporativos.
		Enumeración pasiva de DNS y huella de puertos externos.
		Revisión de redes sociales y fugas de metadatos en documentos públicos.
2. Weaponization (Desarrollo de Armas)	Preparación del exploit y carga maliciosa que aprovechará la información reunida.	Análisis de muestras sospechosas en sandbox (Cuckoo, Any.Run).
		Revisión documental de procesos de generación de macros y binarios internos.
		Validación de reglas de detección de firmas en EDR/antimalware.
3. Delivery (Despliegue)	Transmisión del arma al objetivo mediante correo, web, USB u otro canal.	Pruebas de filtrado de adjuntos en gateway de correo (simular envío de archivo malicioso).
		Evaluación de listas blancas y reglas de proxy para descargas de ejecutables.
		Test de bloqueo de dispositivos extraíbles con política de grupo.
4. Exploitation (Explotación)	Ejecución del código malicioso en el sistema víctima para aprovechar una vulnerabilidad.	Escaneo autenticado de parches faltantes (Nessus, OpenVAS).
		Intento controlado de explotación de CVE recientes en entorno de pruebas.
		Verificación de mitigaciones (ASLR, DEP, EMET, Windows Exploit Guard).
5. Installation (Persistencia)	Persistencia mediante instalación de backdoors, servicios o modificaciones de registro.	Enumeración de servicios y tareas programadas desconocidas con herramientas forenses (Autoruns).
		Pruebas de escritura en rutas críticas (%SYSTEM32%) para usuarios no privilegiados.
		Revisión de integridad de software con firmas y checksums.
6. Command & Control (C2) (Comando y Control)	Establecimiento de canal remoto para controlar el sistema comprometido.	Simulación de beaconing con Cobalt Strike o Caldera en red aislada.
		Inspección de reglas de firewall y egress filtering para conexiones salientes no estándar.
		Análisis de logs de DNS y proxy en busca de dominios generados algorítmicamente.
7. Actions on Objectives (Acciones Maliciosas)	Ejecución de las metas finales: exfiltración, sabotaje, robo de credenciales.	Pruebas de transferencia controlada de grandes volúmenes de datos y monitoreo DLP.
		Simulación de escalamiento de privilegios con herramientas como BloodHound y mimikatz.
		Validación de políticas de segregación de datos sensibles y alertas de acceso anómalo.

2.2 ANALIZAR EL ACTIVO Y VECTOR DE ATAQUE

Como primer paso, el auditor debe comprender con precisión cuál es el objeto que auditará y cómo este podría ser comprometido. Esta información suele encontrarse en el inventario de activos de información o puede ser levantada de forma manual. Idealmente, se debería conocer:

- **Tipo de Activo: Servidor, Aplicación Web, API, Base de Datos, etc. `**
- **La Criticidad del activo: Definida por la organización en virtud de los procesos en la que es utilizada y su requisito de Confidencialidad, Integridad y Disponibilidad.**
- **La Versiones y las Configuraciones del activo3: Usualmente, registrados en la CMDB o en el inventario de activos.**
- **Vulnerabilidades conocidas a las que el activo es vulnerable: Revise las bases CVE o CIS Benchmark para determinar si la versión actual tiene una vulnerabilidad conocida nueva.**
- **El/Los Vector de ataque que pueden ser usados para llegar al activo: Defina el o los vectores que utilizará para realizar la prueba (interno, in-band, red inalámbrica, etc.)**

Vectores de Ataque:

El vector de ataque es la ruta o mecanismo específico que un actor de amenaza puede aprovechar para interactuar con un activo e intentar comprometer su confidencialidad, integridad o disponibilidad. Describe desde dónde y a través de qué medio se origina la acción hostil (red, aplicación, dispositivo, interfaz física, etc.), e incluye las condiciones técnicas que la hacen posible (protocolos expuestos, credenciales débiles, puertos abiertos, accesos físicos, configuraciones incorrectas).

Categoría	*Descripción Operativa*	*Descripción Operativa*
Externo	El intento de intrusión parte fuera del perímetro corporativo (Internet, redes de terceros).	Exploits sobre puertos publicados (HTTP, VPN, SSH).
		Fuerza bruta contra portales remotos.
		Phishing contra usuarios con acceso VPN.
Interno	La amenaza se origina dentro de la red o de una cuenta con credenciales válidas.	Escalada lateral desde una estación comprometida.
		Uso indebido de privilegios por parte de un empleado.
		Conexiones no autorizadas entre VLAN internas.
Inband	El ataque viaja por el mismo canal de comunicación diseñado para la funcionalidad legítima.	SQL Injection en formularios web.
		Comandos OSInjection en API REST.
		Manipulación de cabeceras HTTP (Host, XForwardedFor).
Outofband	Se desencadena a través de un canal distinto al principal, normalmente para eludir controles o recuperar datos a ciegas.	XXE "blind" que fuerza al servidor a leer un archivo y enviarlo a un DNS externo.
		SSRF que causa peticiones HTTP hacia servicios internos.
Red Inalámbrica	Se explota la capa radio (WiFi, Bluetooth) para acceder o espiar la red.	Rogue AP imitando SSID corporativo.
		Captura de handshakes WPA2 y cracking offline.
		BlueBorne sobre dispositivos Bluetooth cercanos.
Acceso Físico	El adversario interactúa directamente con el hardware o el entorno físico del activo.	Extracción de discos o memorias RAM.
		Uso de puertos de consola y JTAG para routers o IoT.
		Conexión de dispositivos USB maliciosos.

2.3 CLASIFICAR LOS TIPOS DE PRUEBAS

Seleccione el tipo de prueba basándose en el objetivo de la evidencia. Si busca determinar si el control existe, basta chequear la información documental versus la configuración real. Si busco determinar si el control es efectivo para su función, entonces una prueba técnica o física es más útil. Las categorías de pruebas pueden ser:

- **1- Documental: Revisión de políticas, registros, evidencias de cumplimiento.**
- **2- Configuración: Validación de parámetros contra guías de hardening.**
- **3- Técnica: Escaneo y explotación controlada (red, aplicación, nube).**
- **4- Física: Acceso a racks, lectores biométricos, WLC, PLC industriales.**

2.4 ESCOGER TÉCNICAS Y HERRAMIENTAS

Las técnicas y herramientas dependen de cada activo, para asegurar la replicabilidad de la prueba, es importante mantener un registro con la versión de la herramienta, los parámetros exactos utilizados y el hash del binario. Este registro puede ser realizado al momento de realizar la prueba, sobre todo si se planea ejecutar la prueba mucho después de su diseño (ver explicación en el ejemplo).

Activo: Controlador de Dominio

Prueba: Técnica, validar política de contraseñas.

Técnica	*Herramienta*	*Parámetro*	*Versión*
SMB Signing	PowerShell Get-AD*	PS C:\> Get-ADUser -Filter * -SearchBase "OU=Finance,OU=UserAccounts,DC=FABRIKAM,DC=COM"	5.1.26

2.5 DEFINIR CRITERIOS DE ÉXITO Y UMBRALES

En este paso, se debe convertir el requisito o criterio en una regla cuantificable. Por ejemplo, frente al requisito "Desahbilitar TLS 1.0" el criterio seria tal que:

PASS si sslscan >= TLS 1.2, de otro modo FAIL

También se debe asignar un valor cuantificable de severidad, siempre cuidando la justificación técnica y regulatoria para cada umbral y valor. Se pueden utilizar métodos abiertos como DREAD o CVSS Score, o bien la escala de riesgos interna de la organización. Por ejemplo:

Sslscan fail > Severidad alta > CVSS Score homologado a 7.5 (Muy alto)

Sslscan fail > Severidad alta > Escala de Riesgo Interna homologado a 16 "Muy Alto"

2.6 ESTABLECER UN MÉTODO DE OBTENCIÓN Y CUSTODIA DE EVIDENCIA

Por último, defina el formato (CVS, txt, captura de pantalla, JSON, pcap, etc.) **asegurando que sea suficiente**, agregue la fecha, zona horaria y host de origen. Observe la política de criptografía de su organización, si existe, para seleccionar el tipo de cifrado y hash que aplicará.

Ubique la evidencia en un repositorio cifrado, con privilegios de lectura limitados al personal autorizado del equipo auditor, previamente definido en el programa de auditoría.

Como buena práctica, identifique cada pieza de evidencia con una numeración única y registre cada acceso, copia o traslado.

2.7 EJEMPLO DE PLAN DE PRUEBAS:

Ejemplo: Diseño de Prueba de Verificación de Contraseña de Dominio en AD	
Activo	Controlador de Dominio Windows Server 2022
Vector de Ataque	Vector 1: Acceso RDP Remoto Vector 2: Red Interna Corporativa VLAN-SEC
Controles y Requisitos de Referencia	NIST PR.AC-01 ISO 27002:2022 5.18 Política Interna de IAM
Tipo de Prueba	Prueba técnica autenticada de configuración
Método	Consulta directa a GPO (Políticas Globales) y verificación automatizada. Sin muestreo, se revisan todas las GPO asociadas al dominio.
Técnicas y Herramientas	Powershell: Para extraer política efectiva (con PowerShell (GetADDefaultDomainPasswordPolicy). Herramienta: Powershell 7.4
	CrackMapEec: Para enumerar las políticas en el DC remoto. Herramienta: CrackMapExec 6.1
	Script Python: Para generar un informe JSON que evalúe la complejidad de las contraseñas.
Criterios de Éxito y Umbrales	Longitud mínima >= 14 caracteres Bloqueo: Máximo 3 intentos Caducidad <= 60 días Estos requisitos se encuentran definidos en la Política Interna de IAM de la organización. Si alguno de los criterios falla, se le asignará una severidad media (homologado a CVSS 6.0) o alta (homologado a CVSS 7.5).
Método de Obtención y Custodia	Guardar salida de Powershell y CME en archivos .txt y captura de pantalla. Calcular hash (mínimo SHA 256) de la evidencia. Registrar fecha, hora y usuario que realizó la extracción.

Notas:

1. **Es posible pre-diseñar y ejecutar los pasos de una prueba estableciendo el parámetro exacto o procedimiento detallado. Sin embargo, pequeños cambios en el entorno de ejecución de la herramienta, de la conexión con el objeto auditado (en el ejemplo anterior, el DC), de la herramienta misma, o de configuraciones que afecten el proceso podría requerir la modificación de la ejecución (por ejemplo, agregando un switch más o menos, o modificado un parámetro como el nombre del usuario). Por ello, la ejecución en detalle se registra a posterori. Si usted lo desea, o se exige el diseño a priori del paso por paso de la prueba, deberá incluir los cambios en los comandos y herramientas utilizadas (si los hay) como parte del reporte de la auditoría.**

Capítulo 3

EJECUCIÓN DE PRUEBAS

3 EJECUCIÓN DE PRUEBAS

Consideraciones iniciales:

> Entorno de pruebas, red, accesos, credenciales si es caja blanca, excepciones en las reglas de fw de acuerdo con el nivel de ruido definido

> Virtualización: configuración de la red, asilamiento de malware si es que se considera como parte del ejercicio, recursos de laboratorio,

Para una lectura más sencilla, se propone una metodología que sigue la lógica similar al de "caja negra". Es decir, las pruebas se ordenan según las fases que debería ejecutar un adversario para comprometer un activo o sistema.

Es normal que las auditorías internas que son parte de un programa de auditoría continua no requieran el enfoque de caja negra, pues podría estar enfocado en un activo o control particular.

ATENCIÓN:

En todo momento recuerde el objetivo de esta guía: ejemplificar el proceso de desarrollo de una auditoría en el contexto de ciberseguridad, las pruebas que se describen en este documento tienen un propósito ilustrativo.

Recuerde: El primer paso es seleccionar las herramientas idóneas, considerando el activo, el entorno, los límites del alcance y ROE y la disponibilidad los instrumentos.

3.1 PRUEBAS DE RECONOCIMIENTO Y ENUMERACIÓN

Las pruebas de reconocimiento y pruebas de enumeración son las etapas donde ocurre la recolección de información de la infraestructura de TI de una organización. Estas técnicas resultan especialmente útiles en la ejecución de auditorías internas de infraestructuras críticas y entornos del sector público, ya que permiten al auditor identificar la superficie de ataque y posibles debilidades antes de que lo haga un adversario.

3.1.1 Pruebas de Reconocimiento

Las pruebas de reconocimiento (también llamadas recon, reconnaissance o recolección de información) implican recopilar datos generales del objetivo (sistema, red u organización) antes de intentar explotarlo. El objetivo de la etapa de reconocimiento es obtener un mapa del entorno: identificar activos, direcciones IP, topología de la red, puertos abiertos, servicios visibles, etc.

Durante esta fase inicial, el auditor puede usar:

- **Técnicas de reconocimiento pasivo: Sin interactuar directamente, Ej: buscando información pública o metadatos.**
- **Técnicas de reconocimiento activo: Interactuando con los sistemas objetivo, Ej: mediante escaneos de red.**

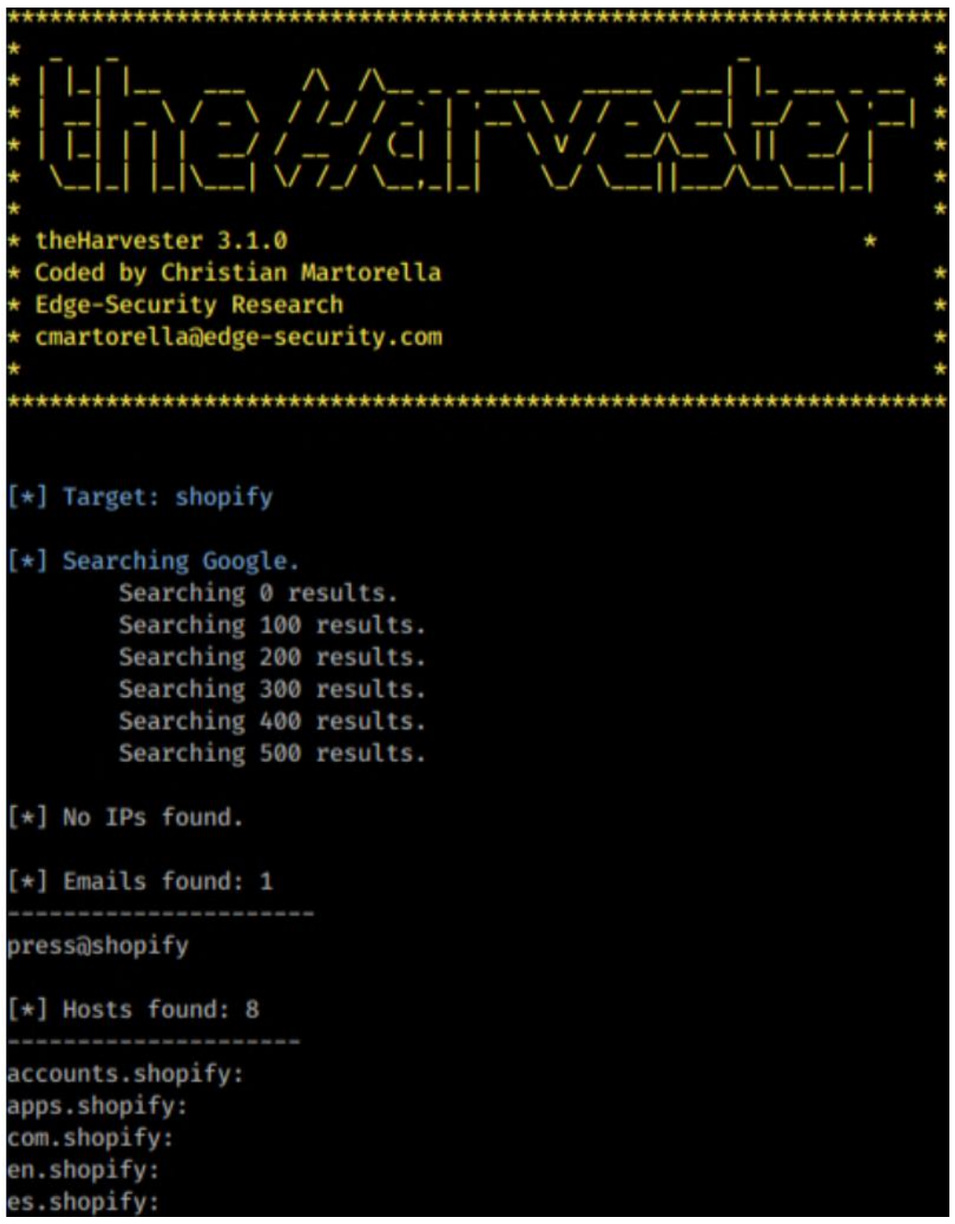
ATENCIÓN:

En todo momento debemos hacernos la siguiente pregunta ¿cuál es el objetivo que estoy siguiendo? Al realizar una prueba de reconocimiento no tenemos la intención de recopilar datos de la infraestructura TI para nosotros (¡Sería más fácil pedir la información al encargado de TI!), si no, emular la técnica del adversario y responder:

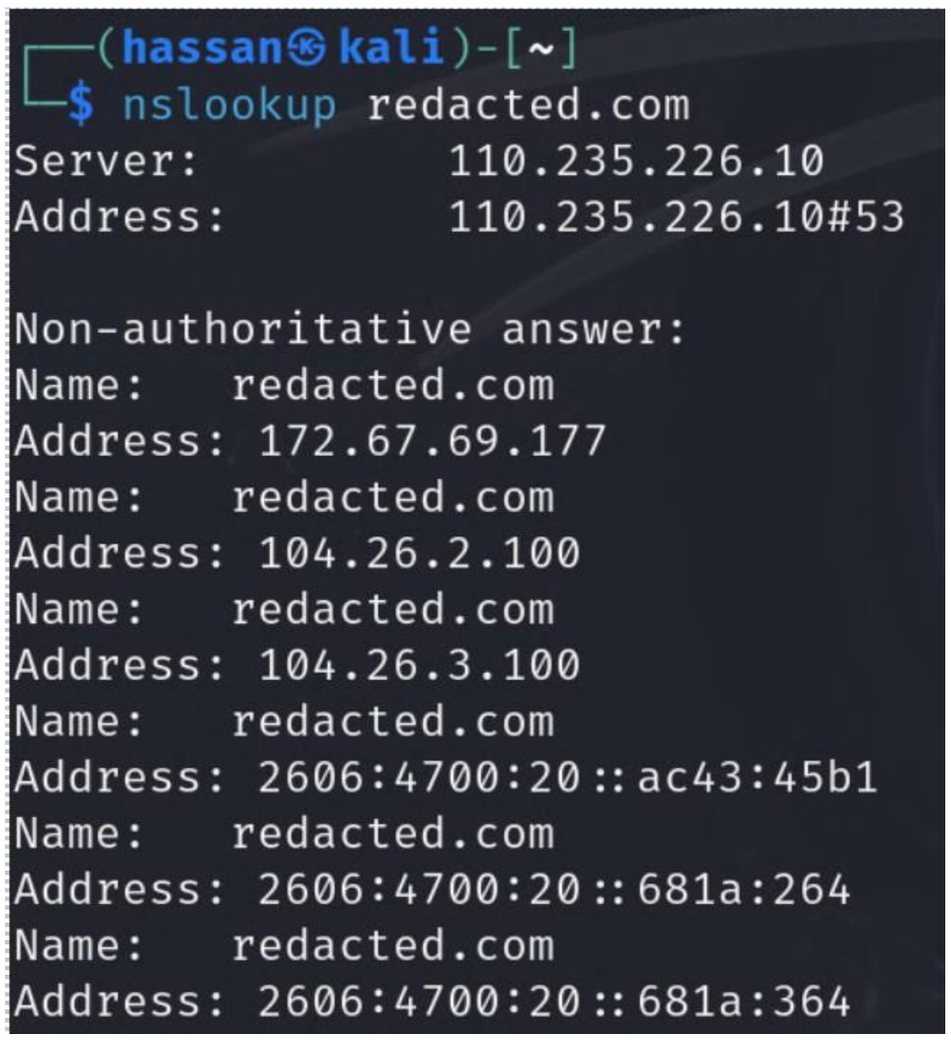
¿Podría un adversario obtener información sobre los activos de mi organización?

¿Hemos configurado correctamente los dispositivos, para evitar este tipo de intrusión?

¿Están funcionando bien los controles de seguridad?

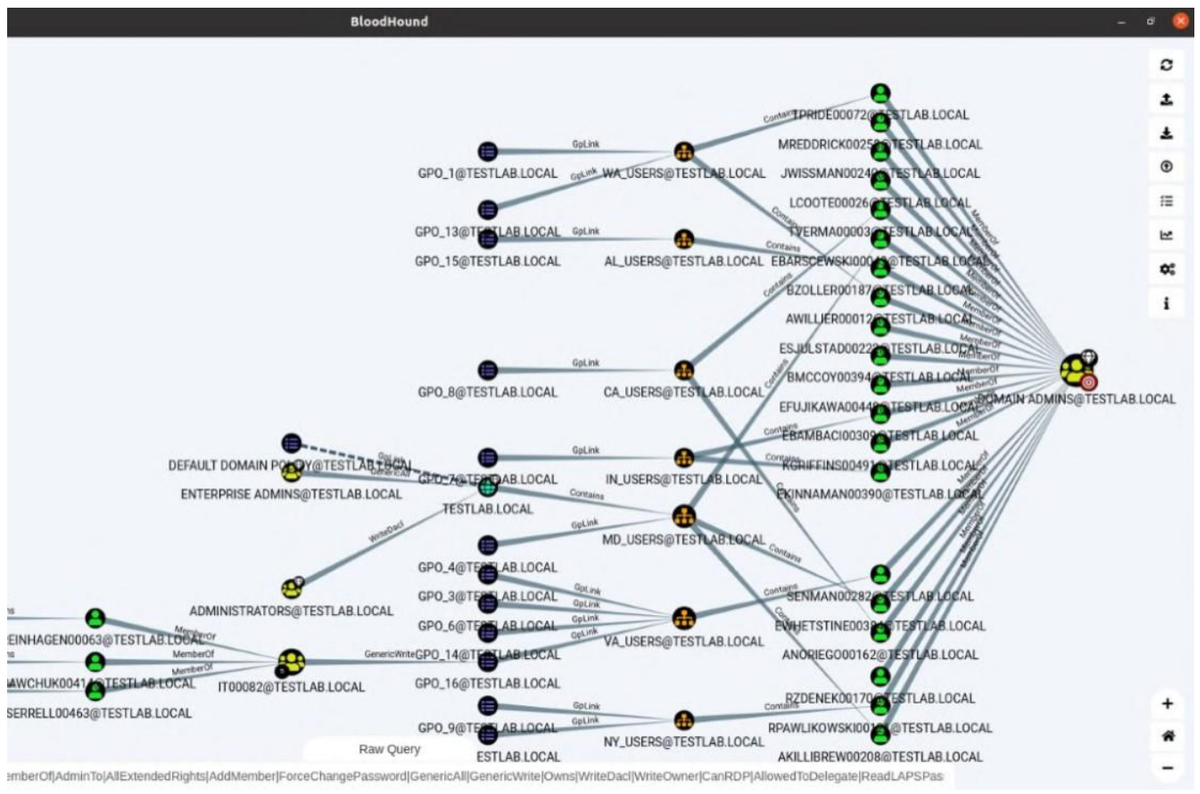


Herramienta de Reconocimiento Pasivo – The Harvester.



Herramienta de Reconocimiento Pasivo – Nslookup.

Herramienta de Reconocimiento Pasivo - Sigurlfind3r.



Herramienta de Reconocimiento y descubrimiento de hosts - BloodHound.

Herramienta de Reconocimiento y descubrimiento de hosts - BloodHound.

En una auditoría interna de red, las pruebas de reconocimiento suelen incluir:

- **Descubrimiento de hosts y redes:**
  Identificar qué equipos están conectados y en qué segmentos de la red (direcciones IP activas, rangos de IP relevantes).

- **Escaneo de puertos:**
  Enviar paquetes para determinar qué puertos están abiertos o filtrados en cada host y, por tanto, qué servicios pueden estar escuchando. Esto revela "puertas de entrada" al sistema.

- **Detección de sistema operativo y hardware:**
  Deducir qué sistema operativo o dispositivo corre en un host (Ej: Windows vs. Linux, modelo de un PLC o router) a partir de huellas en las respuestas de red.

- **Mapeo de la red:**
  Entender la arquitectura, posibles firewalls, rutas y subredes existentes en la infraestructura (Ej: si hay segmentación entre redes).

La etapa de reconocimiento se puede realizar entregando o no credenciales a los sistemas. Por supuesto, si se trata de un ejercicio de caja negra, no se entrega ningún tipo de credencial. Cuando se trata de un ejercicio de caja blanca o gris, se pueden entregar todas o algunas credenciales. Es importante que el nivel de información que se entregue al auditor para realizar las pruebas coincida con el tipo de ejercicio que se está realizando.

Objetivos del Auditor para la fase de reconocimiento:

1. **Identificar activos críticos y puntos de entrada a la red.**
2. **Identificar los hosts accesibles y puertos abiertos en la red.**
3. **Identificar información de acceso público sobre la organización que podría generar un compromiso.**

3.1.2 Pruebas de Enumeración

Las pruebas de enumeración ocurren típicamente tras el reconocimiento. Una vez identificados los sistemas y servicios abiertos, la enumeración consiste en conectar activamente con esos servicios para extraer información detallada: usuarios, nombres de equipo, versiones de software, recursos compartidos, etc. A diferencia del reconocimiento (que puede ser pasivo), la enumeración requiere interacción directa con los sistemas objetivo, enviando consultas o peticiones y analizando las respuestas

En esencia, la enumeración profundiza en lo descubierto durante el reconocimiento. Si el reconocimiento "mapea" la red, la enumeración "boqueja el estado de seguridad" detallado del objetivo. Por eso, suele considerarse una extensión activa de la fase de recon: "durante el reconocimiento se recaba información amplia; la enumeración permite una exploración más específica y profunda de servicios, configuraciones y cuentas".

En esta fase, el auditor profundiza en cada activo, puerto o servicio descubierto. **Los objetivos de la enumeración son**, entre otros:

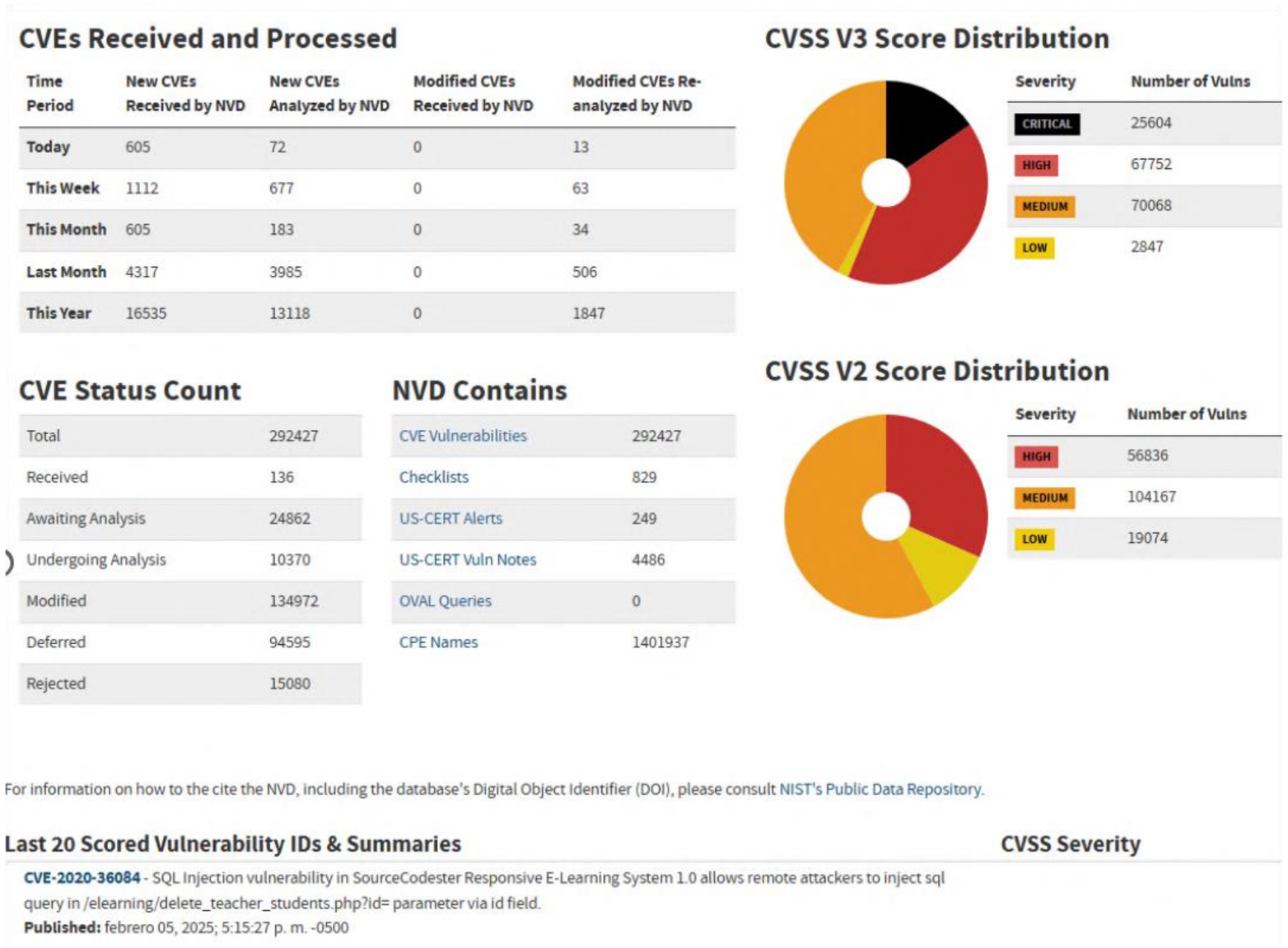
A) Identificar servicios y versiones:

Por cada puerto abierto detectado, determinar qué servicio específico corre allí y qué versión (por ejemplo: SSH 7.4, Apache 2.4.48, MSSQL 2019, etc.). Con técnicas de fingerprinting se obtiene la "huella" del servicio para conocer su software exacto. Saber la versión permite luego verificar si tiene vulnerabilidades conocidas.

Es fundamental recordar que todo este análisis se realiza sin comprometer el activo analizado.

Para verificar si el servicio o componente tiene vulnerabilidades en la versión detectada, el auditor puede consultar a una base de datos de vulnerabilidades conocidas, como la CVE:

https://nvd.nist.gov/general/nvd-dashboard



CVEs Received and Processed

Time Period	New CVEs Received by NVD	New CVEs Analyzed by NVD	Modified CVEs Received by NVD	Modified CVEs Re-analyzed by NVD
Today	605	72	0	13
This Week	1112	677	0	63
This Month	605	183	0	34
Last Month	4317	3985	0	506
This Year	16535	13118	0	1847

CVE Status Count

Total	292427
Received	136
Awaiting Analysis	24862
Undergoing Analysis	10370
Modified	134972
Deferred	94595
Rejected	15080

NVD Contains

CVE Vulnerabilities	292427
Checklists	829
US-CERT Alerts	249
US-CERT Vuln Notes	4486
OVAL Queries	0
CPE Names	1401937

CVSS V3 Score Distribution

Severity	Number of Vulns
CRITICAL	25604
HIGH	67752
MEDIUM	70068
LOW	2847

CVSS V2 Score Distribution

Severity	Number of Vulns
HIGH	56836
MEDIUM	104167
LOW	19074

For information on how to the cite the NVD, including the database's Digital Object Identifier (DOI), please consult NIST's Public Data Repository.

Last 20 Scored Vulnerability IDs & Summaries **CVSS Severity**

CVE-2020-36084 - SQL Injection vulnerability in SourceCodester Responsive E-Learning System 1.0 allows remote attackers to inject sql query in /elearning/delete_teacher_students.php?id= parameter via id field.
Published: febrero 05, 2025; 5:15:27 p. m. -0500

¿Sabías qué?:

Aunque parezca obvio, no todas las vulnerabilidades son conocidas. Por lo tanto, no es posible asegurar con un cien por cien de probabilidades que una versión de una aplicación o servicio es segura. Una nueva versión siempre trae, de forma inherente, nuevos riesgos no conocidos. Cuando un atacante explota una vulnerabilidad no conocida se denomina "explotación de vulnerabilidad de día cero"; son más riesgosas y la mejor forma de protegernos en a través de una seguridad en profundidad bien diseñada, correctos controles de acceso y para el peor de los casos, buenos mecanismos de recuperación ante incidentes.

B) Enumerar usuarios y cuentas:

Obtener listas de nombres de usuario o cuentas presentes en un sistema o dominio. Por ejemplo, enumeración de usuarios de Windows a través de una sesión SMB anónima, o usuarios de un directorio LDAP. Esta información ayuda a detectar cuentas por defecto o credenciales débiles.

Un ejemplo podría ser el siguiente: "*mediante un escaneo a un controlador de dominio Windows, un auditor podría establecer una sesión nula SMB (anónima) que le devuelva la lista de usuarios del dominio si la configuración lo permite*".

Existen herramientas específicas para llevar a cabo este tipo de análisis, como la lista que se expone a continuación. Recuerde que es importante considerar que herramientas están vigentes y cuáles son adecuadas para el activo y el entorno donde se realiza la actividad.

- **01. KrbGuess:**
  Herramienta Java independiente para enumerar nombres de usuario de dominios Kerberos.

- **02. Ldapdomainnom:**
  Herramienta que utiliza LDAP para la enumeración de cuentas, especialmente útil para descubrir nombres de usuario en un dominio.

- **03. Nmap:**
  Un escáner de red versátil que permite identificar puertos y servicios abiertos, proporcionando información sobre recursos de red y posibles vulnerabilidades, incluyendo cuentas de usuario.

- **04. NBTEnum:**
  Utilidad de línea de comandos que utiliza sesiones nulas para recuperar listas de usuarios, listas de máquinas y otra información de una red.

- **05. User2sid y Sid2user:**
  Estas utilidades ayudan a asignar nombres de cuentas de usuario a identificadores de seguridad (SID) y viceversa, proporcionando información sobre la información de las cuentas de usuario.

- **06. Zed Attack Proxy (ZAP):**
  Escáner de seguridad de aplicaciones web que permite detectar vulnerabilidades de enumeración de cuentas en aplicaciones web.

- **07. Etc:**
  Hay, literalmente, cientos de aplicaciones diferentes.

Nunca está de más revisar si existen usuarios, casillas de correo o contraseñas filtradas en internet. Existen muchos foros de internet y recopilaciones de bases de datos de credenciales que pueden ser utilizadas, con ellas se puede probar si una contraseña ha sido reutilizada o como un insumo para robustecer los ataques de diccionario y fuerza bruta.

```
msf auxiliary(kerberos_enumusers) > run

[*] Validating options...
[*] Using domain: MYDOMAIN...
[*] 192.168.5.10:88 - Testing User: "bob"...
[*] 192.168.5.10:88 - KDC_ERR_PREAUTH_REQUIRED - Additional pre-authentication required
[+] 192.168.5.10:88 - User: "bob" is present
[*] 192.168.5.10:88 - Testing User: "alice"...
[*] 192.168.5.10:88 - KDC_ERR_PREAUTH_REQUIRED - Additional pre-authentication required
[+] 192.168.5.10:88 - User: "alice" is present
[*] 192.168.5.10:88 - Testing User: "matt"...
[*] 192.168.5.10:88 - KDC_ERR_PREAUTH_REQUIRED - Additional pre-authentication required
[+] 192.168.5.10:88 - User: "matt" is present
[*] 192.168.5.10:88 - Testing User: "guest"...
[*] 192.168.5.10:88 - KDC_ERR_CLIENT_REVOKED - Clients credentials have been revoked
[-] 192.168.5.10:88 - User: "guest" account disabled or locked out
[*] 192.168.5.10:88 - Testing User: "admin2"...
[*] 192.168.5.10:88 - KDC_ERR_C_PRINCIPAL_UNKNOWN - Client not found in Kerberos database
[*] 192.168.5.10:88 - User: "admin2" does not exist
[*] 192.168.5.10:88 - Testing User: "admin"...
[*] 192.168.5.10:88 - KDC_ERR_C_PRINCIPAL_UNKNOWN - Client not found in Kerberos database
[*] 192.168.5.10:88 - User: "admin" does not exist
[*] 192.168.5.10:88 - Testing User: "administrator"...
[*] 192.168.5.10:88 - KDC_ERR_C_PRINCIPAL_UNKNOWN - Client not found in Kerberos database
[*] 192.168.5.10:88 - User: "administrator" does not exist
[*] Auxiliary module execution completed
msf auxiliary(kerberos_enumusers) >
```

KRBGUEST:
Herramienta utilizada para enumerar los nombres de usuarios en kerberos.

```
┌──(kali㉿kali)-[~/marvel.local]
└─$ sudo ldapdomaindump ldaps://192.168.138.136 -u 'MARVEL\fcastle' -p Password1
[sudo] password for kali:
[*] Connecting to host...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
[+] Domain dump finished

┌──(kali㉿kali)-[~/marvel.local]
└─$ ls
domain_computers_by_os.html  domain_groups.json  domain_trusts.json
domain_computers.grep        domain_policy.grep  domain_users_by_group.html
domain_computers.html        domain_policy.html  domain_users.grep
domain_computers.json        domain_policy.json  domain_users.html
domain_groups.grep           domain_trusts.grep  domain_users.json
domain_groups.html           domain_trusts.html
```

LDAPDOMAINDUMP:
Herramienta para obtener cuentas en un dominio.

C) Enumerar recursos compartidos y servicios internos:

or ejemplo, listar las carpetas compartidas en un servidor de archivos, enumerar los servicios RPC expuestos, buzones en un servidor de correo, bases de datos abiertas, etc. Cualquier recurso accesible sin credenciales fuertes es un hallazgo. Un sistema mal configurado puede revelar sus compartidos administrativos o información de dominio a consultas anónimas.

Si un auditor encuentra evidencia de que existen carpetas compartidas, inmediatamente puede presumir lo siguiente:

> Un tercero externo y sin autorización podría descargar y obtener la información en las carpetas compartidas. Si el servicio de archivos compartidos no tenía la expresa finalidad de compartir información al público (por ejemplo, un repositorio de circulares públicas) entonces es un hallazgo importante que pone en peligro la privacidad de la información y probablemente un sin fin de criterios relevantes para la organización.

> Un tercero externo y sin autorización podría incluir archivos maliciosos en la carpeta compartida; y luego, un usuario autorizado pero descuidado lo ejecuta en su entorno local. Esta es una excelente forma de comprometer una red.

A demás, es poco probable que la información se almacene en servicios FTP o carpetas de servidor ya que existen mejores alternativas y más amigables por lo que es una buena decisión informar cada recurso compartido inseguro detectado.

```
rpcclient $> enumdomgroups
group:[Enterprise Read-only Domain Controllers] rid:[0x1f2]
group:[Domain Admins] rid:[0x200]
group:[Domain Users] rid:[0x201]
group:[Domain Guests] rid:[0x202]
group:[Domain Computers] rid:[0x203]
group:[Domain Controllers] rid:[0x204]
group:[Schema Admins] rid:[0x206]
group:[Enterprise Admins] rid:[0x207]
group:[Group Policy Creator Owners] rid:[0x208]
group:[Read-only Domain Controllers] rid:[0x209]
group:[DnsUpdateProxy] rid:[0x44e]
rpcclient $>
```

RPCCLIENT:
Herramienta para enumerar servicios RPC expuestos.

D) Recopilar banners y configuraciones:

muchos servicios envían un banner de bienvenida al conectarse, que incluye datos como el nombre del software, versión y a veces configuraciones habilitadas. Banner grabbing es precisamente la técnica de capturar esos mensajes iniciales de los servicios. Por ejemplo, al conectar por Telnet o Netcat a un puerto SMTP o FTP, suele aparecer un banner con la versión del servidor. Es un complemento muy similar al punto A de esta lista.

En esencia, la enumeración profundiza en lo descubierto durante el reconocimiento. Si el reconocimiento "mapea" la red, la enumeración "bosqueja el estado de seguridad" detallado del objetivo. Por eso, suele considerarse una extensión activa de la fase de recon: "durante el reconocimiento se recaba información amplia; la enumeración permite una exploración más específica y profunda de servicios, configuraciones y cuentas".

Caso de Ejemplo:

En una auditoría interna, tras detectar un puerto 445/TCP (SMB) abierto en un servidor, el auditor realiza una enumeración SMB. Utilizando herramientas adecuadas, descubre que el servidor permite consultas anónimas y así obtiene una lista de usuarios y compartidos disponibles. Esto reveló, por ejemplo, un recurso compartido de respaldo con información sensible sin proteger: un hallazgo importante.

Al encontrar el puerto 161/UDP (SNMP) abierto en routers o switches de una infraestructura, el auditor intenta una enumeración SNMP. Si los dispositivos usan la comunidad SNMP por defecto ("public"), la herramienta devolverá una gran cantidad de información de configuración: tablas de rutas, listas de interfaces, versiones de firmware, e incluso credenciales en texto claro en algunos casos. De hecho, se sabe que "la comunidad predeterminada 'public' permite a un atacante leer la configuración completa de un dispositivo por consultas SNMP, y la clave 'private' (si no cambiada) incluso modificar esa configuración". Encontrar SNMP abierto con cadenas por defecto sería un resultado crítico de enumeración.

3.1.3 Hallazgos de Auditoría e Información Relevante

Durante el proceso de enumeración es probable que se obtenga mucha información, listas y listas de activos, versiones, parches, servicios, puertos y sus estados, etc. No toda la información constituye un hallazgo relevante. La información debe reflejar un estado de incumplimiento o riesgo de lo analizado. Como guía general, la siguiente información es interesante de reportar:

- **Inventario y Visibilidad de Activos:**
  El reconocimiento proporciona un inventario actualizado de los hosts y servicios activos en la red. Esto es muy valioso ya que en entornos dinámicos (en especial infraestructura crítica) constantemente aparecen nuevos sistemas o aplicaciones. Un escaneo de red ayuda al auditor a verificar que todos esos sistemas nuevos están debidamente controlados y que no están expuestos inadvertidamente a ataques. Asegura que la organización sabe exactamente qué tiene en la red en un momento dado. Muchas brechas de seguridad ocurren por dispositivos olvidados o no gestionados; el auditor interno, mediante reconocimiento, puede detectar esos activos ocultos.

  Tip:
  Siempre que esté dentro del alcance y en las reglas pactadas puedes contrastar el inventario de activos TI contra el resultado del escaneo de red, quizá el inventario está incompleto, desactualizado o registra activos que no se encuentran disponibles en la red.

- **Detección de configuraciones débiles o incumplimientos de política:**
  Las pruebas de enumeración permiten al auditor comprobar si los servicios descubiertos cumplen con las políticas de seguridad internas. Por ejemplo, si la política dice que "no debe haber comparticiones abiertas sin autenticación" o "SNMP v1 no debe usarse", el auditor lo puede verificar intentando enumerar esos recursos. Si logra enumerar usuarios vía SMB anónimo o leer SNMP con "public", demuestra un incumplimiento. Esto brinda evidencia concreta para recomendar mejoras. Asimismo, la enumeración de versiones de software ayuda a ver si hay software obsoleto no parcheado ejecutándose (lo que sería un hallazgo que sugiere falta de gestión de parches en sistemas críticos).

- **Evaluación de la segmentación y accesos no autorizados:**
  En infraestructuras críticas, suele haber segmentación de redes (OT separada de TI, por ejemplo). El auditor puede usar el reconocimiento desde distintos puntos de la red interna para verificar qué es accesible y qué no. Un hallazgo típico podría ser descubrir que desde la red corporativa se puede escanear y llegar a controladores industriales, cuando en teoría debería estar aislada. Mediante escaneos y enumeración, el auditor comprueba la efectividad de las medidas de segmentación o firewall internas. Si encuentra que servicios críticos están accesibles desde redes no autorizadas, evidenciará la necesidad de reforzar el perímetro interno. Del mismo modo, la técnica ayuda a identificar dispositivos no autorizados conectados: "un dispositivo no confiable es cualquier equipo en la red no aprobado; puede ser un punto de acceso inalámbrico o un portátil personal. Hay muchos riesgos asociados a dispositivos no autorizados, por lo que deben bloquearse hasta ser verificados por TI".

- **Puertos abiertos innecesarios o inseguros:**
  Un hallazgo frecuente es descubrir puertos abiertos que no deberían estarlo en determinados sistemas, o servicios legados utilizando puertos inseguros. Por ejemplo, en una planta de energía podría hallarse que ciertos controladores PLC tienen abierto el puerto Telnet (23/TCP) o HTTP no cifrado, a pesar de no requerirse para la operación normal. Cada puerto abierto de más amplía la superficie de ataque. Como se mencionó, los puertos abiertos son potenciales "puntos de ataque para hackers" si no están justificadosionos. La acción correctiva suele ser cerrar o filtrar esos puertos no necesarios, reduciendo posibles vectores de intrusión.

- **Servicios sin autenticación o con credenciales por defecto:**
  Este es uno de los hallazgos más peligrosos. Gracias a la enumeración, el auditor puede detectar servicios que no exigen credenciales para acceder o que usan contraseñas predeterminadas de fábrica. Ejemplos concretos:

  > **SNMP con comunidad "public"**: Si la enumeración SNMP tuvo éxito, significa que cualquiera con acceso a la red puede leer información de los dispositivos. Un atacante interno podría recopilar configuraciones de routers y switches (mapeando la red completa, con contraseñas cifradas que podrían crackearse, etc.), o incluso cambiar configuraciones si la comunidad "private" también está activa.
  > **Carpetas Compartidas SMB abiertos/anónimos**: Si herramientas como enum4linux revelan que existen shares accesibles sin credencial (sesión nula o usuario invitado), cualquier usuario en la red podría leer/escribir en esos recursos.
  > **Credenciales por defecto en dispositivos críticos:** No cambiar las contraseñas por defecto de dispositivos es un error desafortunadamente común. Al enumerar, podría intentar accesos con credenciales conocidas (por ejemplo, admin/admin en cámaras de seguridad, root/calvin en iDRAC de servidores, etc.)

- **Dispositivos o redes no autorizados:**
  Como ya se mencionó, el reconocimiento puede descubrir equipos ajenos conectados a la red interna. Un hallazgo típico en auditorías es identificar, por ejemplo, una red WiFi oculta operando en la instalación o un servidor conectado indebidamente. Estos dispositivos "sombra" pueden ser puerta de entrada para atacantes si no son gestionados.

- **Información sensible accesible y otras configuraciones débiles:**
  La enumeración a veces revela **datos o configuraciones que, si bien no son vulnerabilidades explotables directamente, suponen riesgos.** Por ejemplo, listar todos los usuarios de Active Directory no es en sí "explotar" algo, pero ese listado de usuarios es información sensible (un atacante podría utilizarlo para ataques de password spraying o ingeniería social dirigida). Otro ejemplo es descubrir mediante DNS interno (zona de dominio) entradas que desvelan la estructura interna (nombres de servidores y funciones). O encontrar en un servidor web un directorio de backups accesible sin autenticación.

3.1.4 Herramientas de Código Abierto para Reconocimiento y Enumeración:

Existen numerosas herramientas open-source que un auditor puede emplear para realizar estas pruebas de forma eficaz. Algunas de las más utilizadas (mencionadas entre paréntesis en la pregunta) son:

Herramienta	*Descripción*
Nmap	Es la herramienta por excelencia para reconocimiento y escaneo de puertos. Nmap (Network Mapper) permite descubrir hosts activos, puertos abiertos, servicios y hasta sistemas operativos mediante técnicas de fingerprinting. Es gratuita y de código abierto, ampliamente utilizada en auditorías de seguridad. Con Nmap se pueden hacer escaneos intensivos de red, identificar qué puertos y servicios están disponibles en cada máquina, y usar sus scripts NSE para tareas de enumeración más avanzadas (por ejemplo, enumerar versiones de servicios, usuarios SMB, consultas LDAP, etc.)
Netcat	Conocido como la "navaja suiza" de las redes, Netcat (o su alternativa ncat) es una pequeña utilidad para leer y escribir datos a través de conexiones de red. En pruebas de reconocimiento sirve para hacer banner grabbing manual – por ejemplo, conectar a un puerto TCP específico y ver la respuesta. Netcat también puede utilizarse para escanear puertos sencillamente, enviar paquetes personalizados o incluso transferir archivos.

Herramienta	*Descripción*
Enum4linux	Es una herramienta especializada en enumeración de sistemas Windows/Samba. Permite obtener información de servidores Windows o Samba usando peticiones SMB/RPC, incluso a veces sin autenticación. Enum4linux extrae datos como usuarios y grupos del sistema, recursos compartidos, políticas de contraseñas e información de dominio. Por ejemplo, en una auditoría de Active Directory es común usar enum4linux (u otras similares como smbclient o rpcclient) para listar usuarios y equipos del dominio, comprobar qué compartidos existen en los controladores de dominio, etc.
Otros	Dependiendo del contexto, existen muchas otras herramientas open-source útiles, por ejemplo: **Nikto**: Escáner web que realiza reconocimiento de servidores HTTP, buscando archivos o directorios comunes y configuraciones débiles.
	Dirbuster/Dirsearch: Enumera directorios y ficheros ocultos en aplicaciones web críticas del sector público.
	Snmpwalk o Onesixtyone: Para enumerar información vía SNMP en dispositivos de red.
	Nbtscan o el comando nbtstat de Windows: para descubrir nombres NetBIOS en redes Windows.
	LDAPSearch: Consulta de directorios LDAP/AD.
	Frameworks como Metasploit: Que incluyen módulos de scanner/enumeración (por ejemplo, para descubrir servicios de FTP anónimos, impresoras de red, etc.)
	PowerView (Parte de PowerShell Empire): Herramienta de código abierto orientada a enumeración de dominios Active Directory, que puede inventariar usuarios, grupos y equipos AD con comandos desde dentro de la red Windows.
	Cada una de estas herramientas ayuda a automatizar la recolección de datos en su ámbito específico. Por ejemplo, PowerView o PingCastle pueden auditar configuraciones de Active Directory, mientras que OpenVAS (versión libre de un escáner de vulnerabilidades) podría complementar la enumeración identificando servicios conocidos por tener fallos. No obstante, las tres primeras (Nmap, Netcat, Enum4linux) suelen cubrir gran parte de las necesidades de reconocimiento activo y enumeración en una auditoría típica.

Objetivos del Auditor para la fase enumeración:

1. **Identificar las características de los dispositivos en la red y el estado de los elementos de configuración.**
2. **Identificar vulnerabilidades conocidas para las versiones y estados de configuración.**
3. **Investigar por información sensible que pueda ser obtenida sin la necesidad de explotar.**
4. **Utilizar las credenciales obtenidas para iterar en este proceso hasta extraer toda la información posible**

3.2 PRUEBAS ASOCIADAS A LA EXPLOTACIÓN DE VULNERABILIDADES

En ciberseguridad, la explotación de vulnerabilidades se refiere al acto de tomar una debilidad o fallo identificado en un sistema y utilizarlo de forma activa para violar la seguridad de ese sistema. Dicho de otro modo, es la fase en la que un atacante "convierte" una vulnerabilidad teórica en un impacto real, por ejemplo, obteniendo acceso no autorizado, ejecutando comandos en el sistema víctima o robando información. La explotación es una etapa típica de las pruebas de penetración y ocurre después de haber detectado vulnerabilidades; en esta fase se lanzan **ataques controlados** aprovechando esas debilidades, de manera similar a cómo lo haría un atacante real.

El objetivo es **validar** la vulnerabilidad y demostrar concretamente qué tan comprometedor puede ser el fallo, obteniendo evidencia de hasta dónde se puede llegar al explotarlo.

La palabra explotación se puede rastrear al anglicismo exploitation, que se puede entender como "aprovecharse de" o "utilizar"

¿Qué implica?

Realizar pruebas de explotación significa usar código malicioso, herramientas o comandos diseñados para aprovechar una vulnerabilidad específica y así penetrar en el sistema afectado. A diferencia de solo detectar o reportar la existencia de un fallo, aquí se intenta ejecutar efectivamente el exploit en un entorno controlado. Por ejemplo, si se descubre un desbordamiento de búfer en una aplicación, la prueba de explotación consistiría en introducir datos especialmente diseñados que provoquen ese desbordamiento y den acceso al auditor (tal como lo haría un atacante para obtener control)

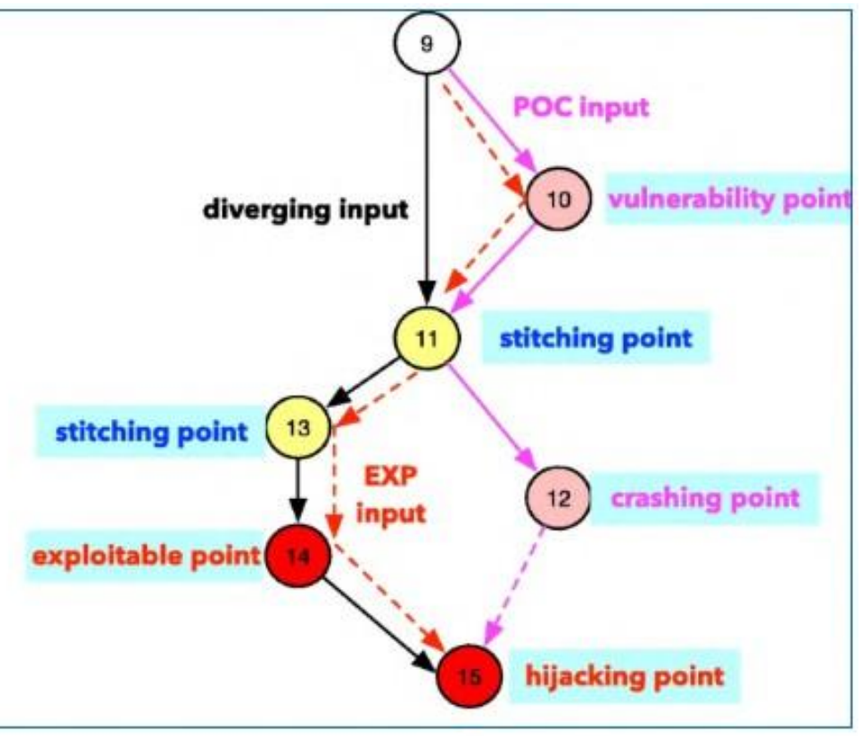
Este proceso puede causar efectos reales en el sistema (como abrir una shell remota o escalar privilegios), por lo que debe realizarse con precaución. La explotación de vulnerabilidades es pasar de la teoría a la práctica ofensiva.

3.2.1 Pruebas de Concepto:

Una prueba de concepto (PoC) en seguridad es una implementación simplificada o código de ejemplo diseñado únicamente para demostrar que una vulnerabilidad es real y explotable. La PoC sirve como evidencia de la existencia del fallo, mostrando en pequeña escala el comportamiento anómalo o la brecha de seguridad sin necesariamente llevar a cabo un ataque completo. Comúnmente, una PoC toma la forma de un script o programa que activa la vulnerabilidad para probar que el sistema es vulnerable

Por ejemplo, si se encuentra una inyección SQL en una aplicación web, un auditor podría escribir una consulta SQL sencilla que, al enviarse como PoC, devuelve información del esquema de la base de datos (demostrando la inyección) pero sin extraer datos sensibles en grandes cantidades. Una PoC de explotación suele involucrar código mínimo y controlado que explota la vulnerabilidad lo suficiente para ilustrar el problema y potencialmente persuadir a los responsables del sistema a corregirlo. Es importante destacar que no busca causar daño; a diferencia de un exploit completo, la PoC normalmente limita sus acciones al mínimo indispensable para comprobar la falla de seguridad.

```
1.  struct Type1 { char[8] data;                                    };
2.  struct Type2 { int status;    int* ptr;  void init(){…};       };
3.  int (*handler)(const int*) = …;
4.  struct{Type1* obj1; Type* obj2;} gvar = {};
5.  int foo(){
6.    gvar.obj1 = new Type1;
7.    gvar.obj2 = new Type2;
8.    gvar.obj2->init();  // resulting different statuses
9.    if(vul)
10.     scanf("%s", &gvar.obj1->data);     // vulnerability point
11.   if(gvar.obj2->status)                // stitching point
12.     res = *gvar.obj2->ptr;             // crashing point
13.   else                                 // stitching point
14.     *gvar.obj2->ptr = read_int();      // exploitable point
15.   handler(gvar.obj2->ptr);             // hijacking point
16.   return res;
17. }
```



PoC a Explotación: Wang, Y., Wu, W., Zhang, C. et al. From proof-of-concept to exploitable. Cybersecur 2, 12 (2019). https://doi.org/10.1186/s42400-019-0028-9

Cuidados Éticos:

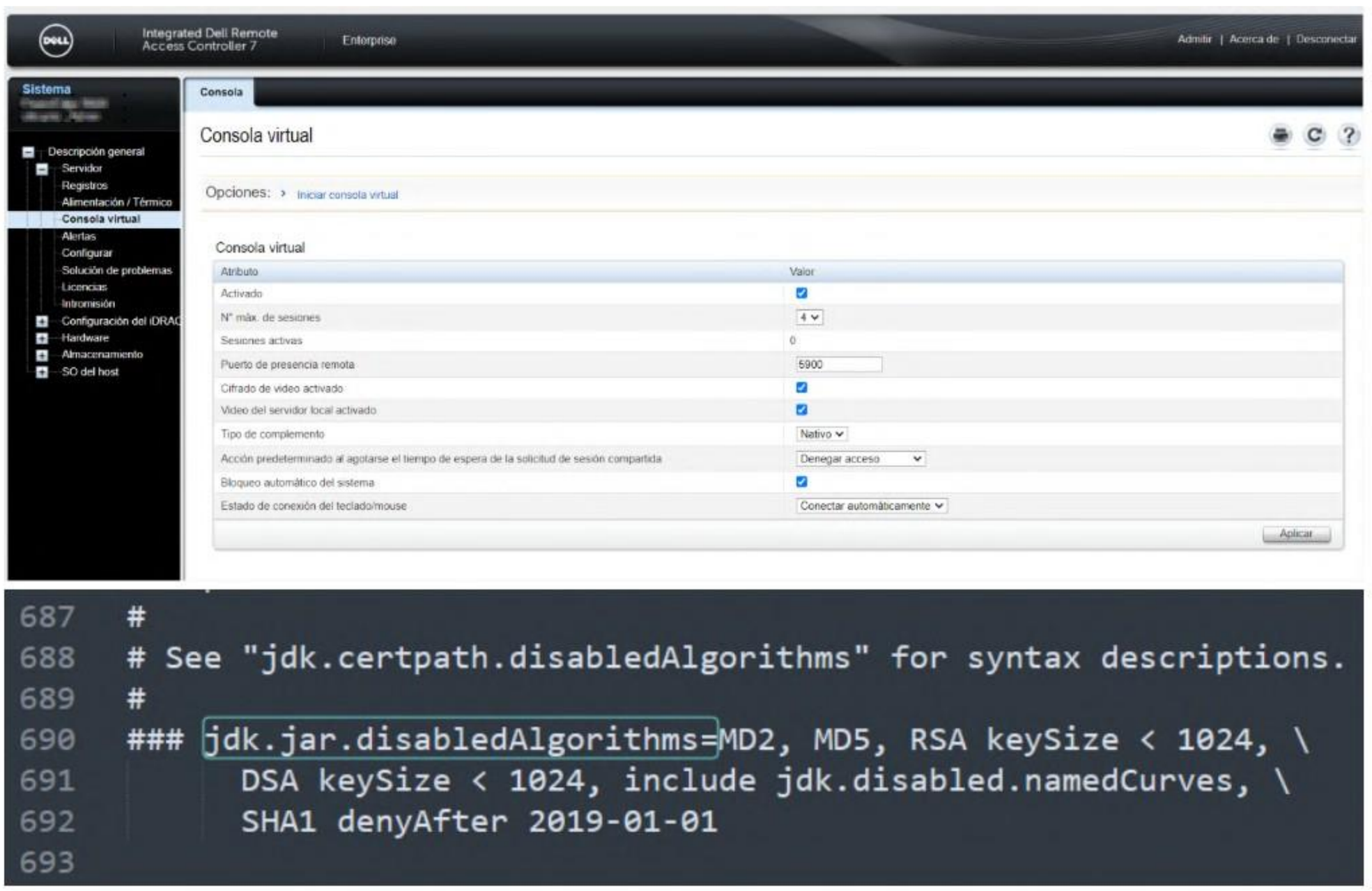
En la comunidad de ciberseguridad, es común que tras descubrirse una vulnerabilidad se desarrolle y comparta una PoC. Esta práctica puede ayudar a presionar a los proveedores de software para que emitan parches más rápido y alertar a la comunidad sobre el riesgo Sin embargo, también conlleva riesgos: una PoC hecha pública puede ser tomada por atacantes maliciosos y convertida en un exploit completamente funcional en muy poco tiempo. Por esta razón, los profesionales deben manejar las PoC de manera responsable y ética.

Hallazgos de Auditoría e Información Relevante:

Cuando se llevan a cabo pruebas de explotación de vulnerabilidades y PoCs en una auditoría, los resultados obtenidos pueden traducirse directamente en hallazgos de seguridad. Estos hallazgos describen qué tipo de compromiso o brecha se pudo lograr, sirviendo para calificar la gravedad del riesgo. Existen muchos tipos de hallazgos diferentes, una buena idea es utilizar una metodología o taxonomía para entender y modelar las técnicas, tácticas y procedimientos que los atacantes utilizan, por ejemplo, ATT&CK de MITRE. Una lista no extensiva, con algunos de los hallazgos más comunes que se pueden encontrar en esta fase es:

- **Posibilidad de ejecutar comandos de forma remota (RCE):**

Es uno de los resultados más críticos. Implica que mediante la explotación de una vulnerabilidad el auditor logró ejecutar comandos arbitrarios en el sistema objetivo, típicamente con los privilegios del servicio vulnerable o incluso con privilegios elevados. Por ejemplo, a través de una vulnerabilidad no parcheada, se pudo obtener una shell remota y ejecutar comandos del sistema operativo en un servidor crítico.

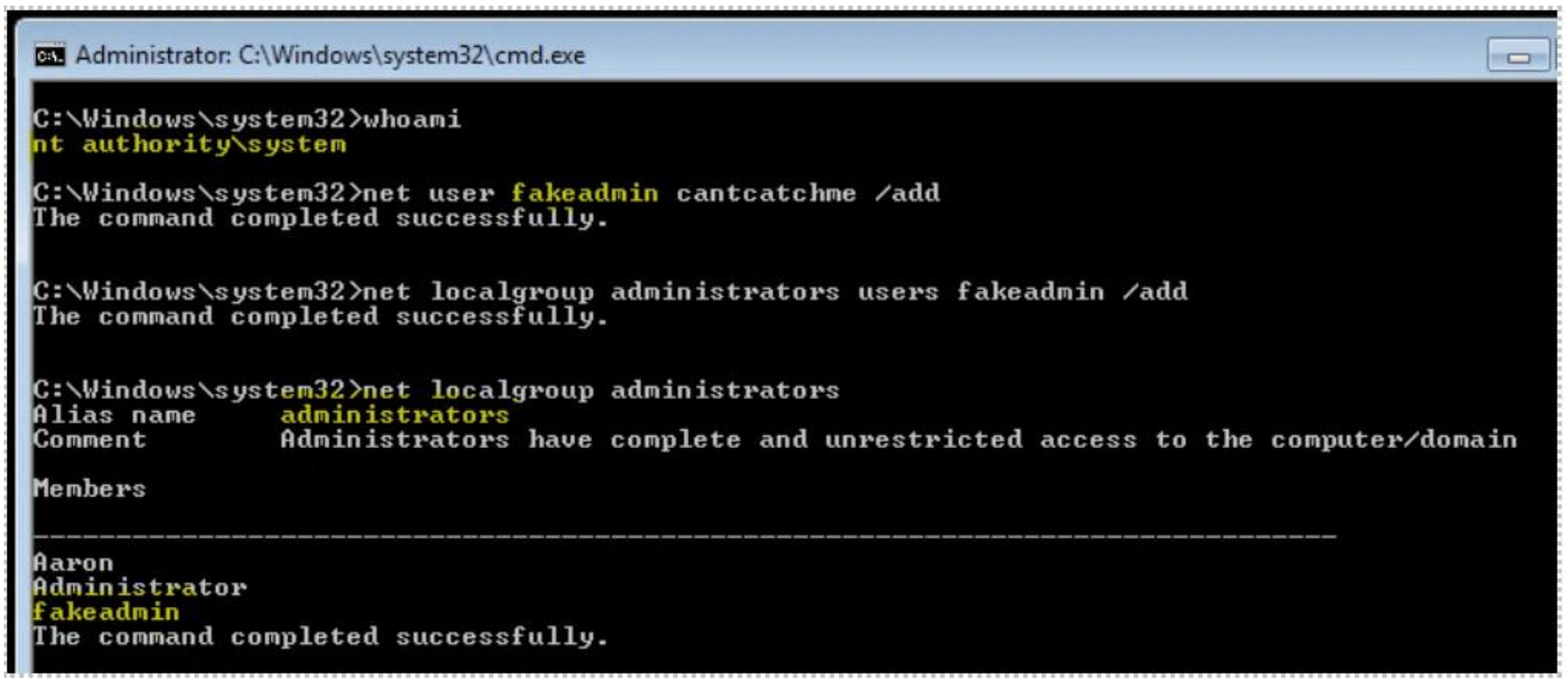


Fuente: https://ricardojoserf.github.io/Exploiting-iDRACs/

Un caso real publicado fue la vulnerabilidad CVE-2018-1207 en controladores Dell iDRAC, que permitía a un atacante remoto ejecutar comandos con privilegios de root. Si un auditor reproduce algo similar como PoC (en un entorno de prueba o con la debida autorización en producción), demostraría que un adversario podría tomar control total de ese sistema.

- **Posibilidad de realizar Escalamiento de Privilegios:**

  Este resultado ocurre cuando el auditor inicialmente consigue acceso limitado (por ejemplo, usuario no privilegiado en un servidor o acceso a una cuenta con pocos derechos) y, mediante explotación de otra vulnerabilidad o configuración débil, logra elevar sus privilegios a niveles administrativos.



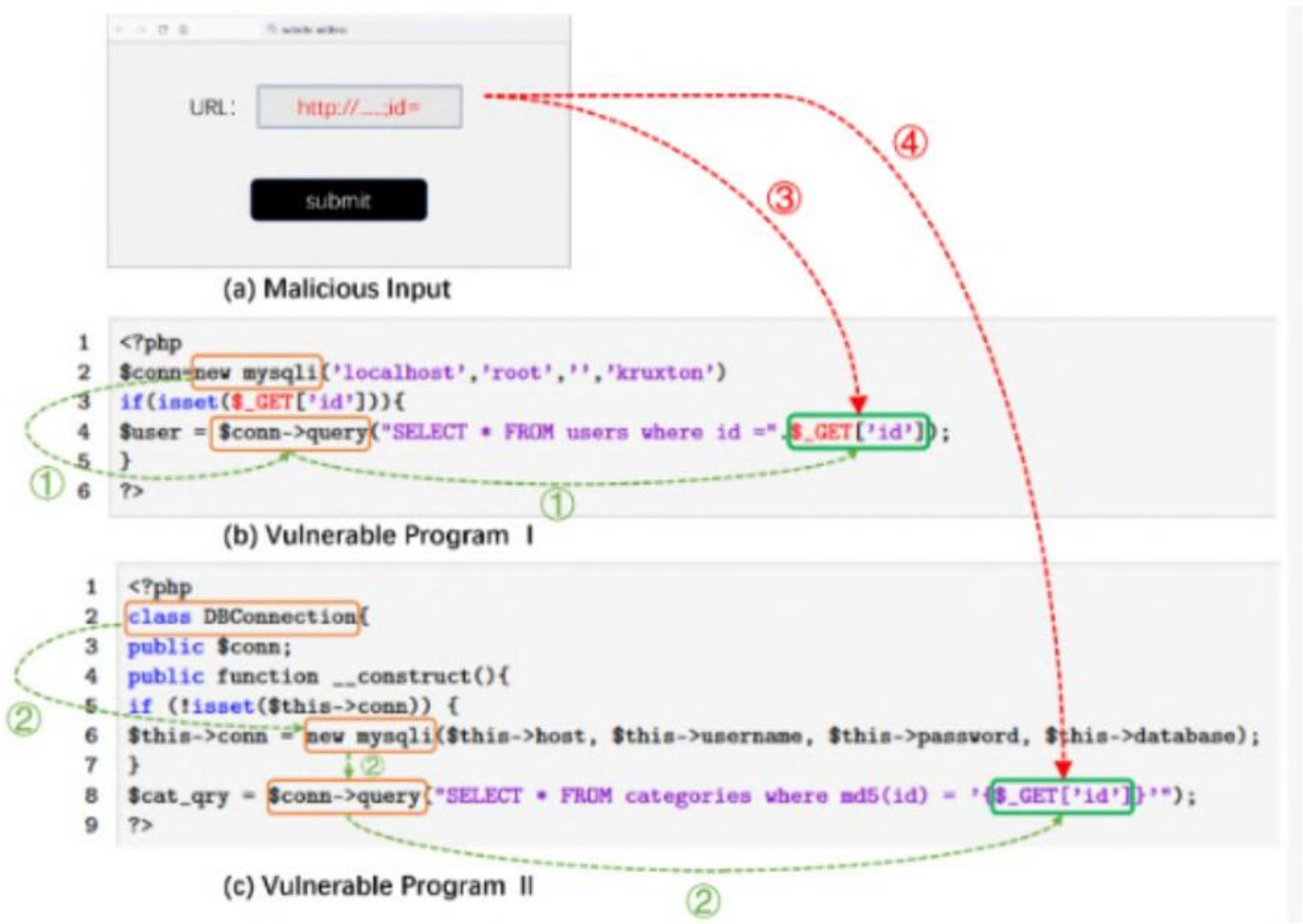
Resultado de un ejercicio de PrivEsc, vía: https://purplesec.us/learn/privilege-escalation-attacks/

Un ejemplo típico es explotar una vulnerabilidad local del kernel de Windows o Linux para pasar de usuario estándar a NT Authority/System o root. En auditorías, la escalada de privilegios evidencia falta de parches o configuraciones inseguras y suele destacarse con prioridad alta para remediación.

Accesos No Autorizados a Datos o Sistemas:

Bajo este paraguas entran hallazgos donde el auditor consiguió acceder a información o recursos que no debería. Puede presentarse de varias formas según la vulnerabilidad explotada:

- **Bypass de Autenticación:** Por ejemplo, a través de una falla en la lógica de autenticación o fuerza bruta de contraseñas, el auditor entra a un sistema/aplicación sin credenciales válidas. Esto podría significar acceder a una cuenta de usuario sin conocer la contraseña (Ej: mediante una SQL injection en la pantalla de login que permite loguearse sin credenciales) o eludir la autenticación de doble factor.
- **Exposición de Información Sensible:** Mediante una explotación (quizá una inyección SQL, un LFI o un acceso directo a S3, etc.), el auditor pudo extraer datos confidenciales: listas de usuarios, datos personales de ciudadanos, registros financieros, planos de infraestructuras, etc.
- **Acceso a Sistemas Internos Adicionales:** En una prueba de explotación exitosa, a veces el auditor compromete un pivote que le abre puertas a otros sistemas (esto es conocido también como movimiento lateral). Un hallazgo podría documentar que desde un servidor expuesto se obtuvo acceso a la intranet interna, violando segmentaciones de red.



Yuan, Y., Lu, Y., Zhu, K., Huang, H., Yu, L., & Zhao, J. (2023). A Static Detection Method for SQL Injection Vulnerability Based on Program Transformation. Applied Sciences, 13(21), 11763. https://doi.org/10.3390/app132111763

Ejemplo en SQL Injection.

- **Posibilidad de llevar a cabo una Denegación de Servicio o Inestabilidad Inducida:**

  Aunque en auditorías internas raramente se busca provocar interrupciones, a veces la explotación de una vulnerabilidad puede provocar caídas de servicio o degradación como efecto secundario.

  > **Por ejemplo:** "Al enviar X peticiones especialmente formateadas el servicio deja de responder, lo cual indica posibilidad de denegación de servicio por atacante remoto". Este tipo de hallazgo alerta sobre vulnerabilidades que un adversario podría usar no para robar datos, sino para interrumpir operaciones, algo muy relevante en infraestructuras crítica.

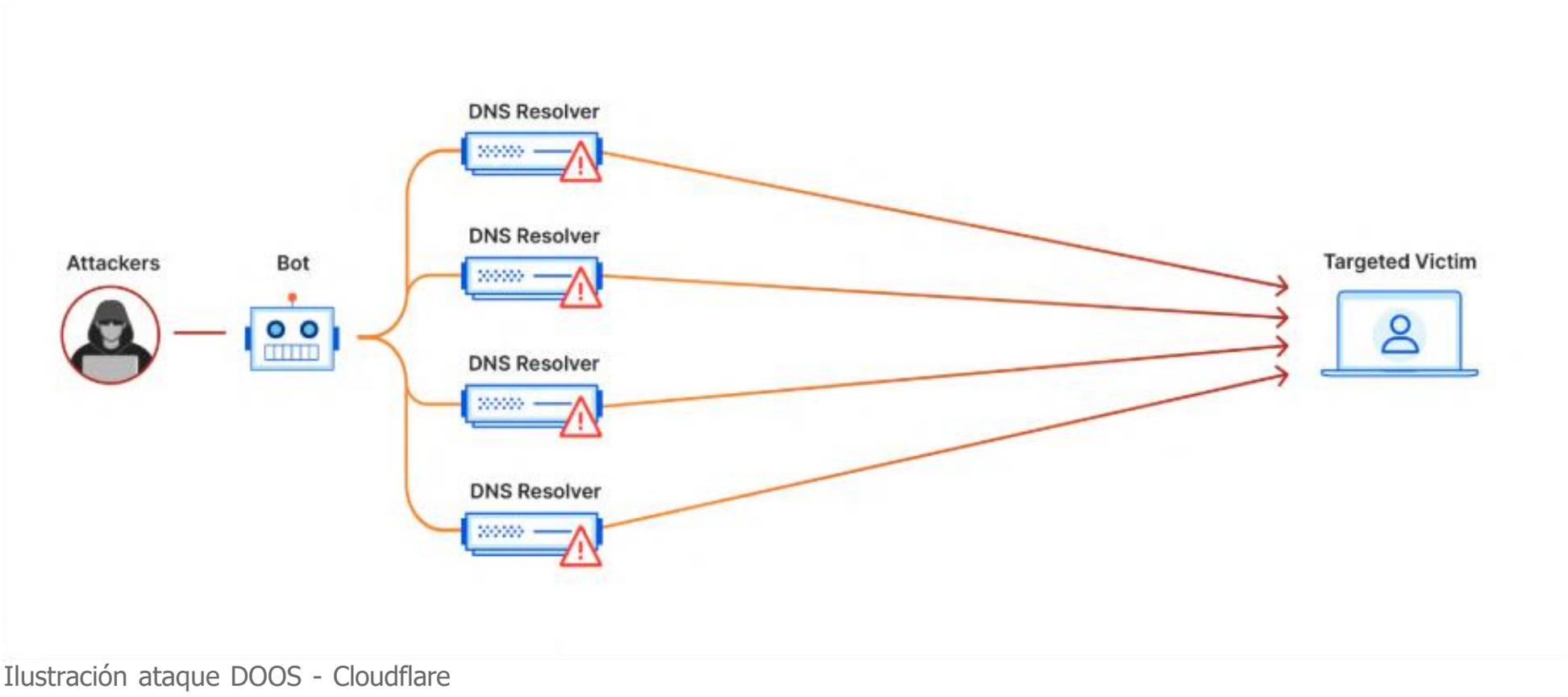


Ilustración ataque DOOS - Cloudflare

3.2.2 Hallazgos Post Explotación:

Si la auditoría permitió realizar etapas posteriores (post-exploitation), los hallazgos pueden incluir cosas como persistencia en el sistema (ejemplo: "se comprobó que es posible instalar una puerta trasera que sobrevive reinicios"), movimiento lateral (como mencionado, acceso a otros sistemas) o captura de credenciales (ej: "se extrajeron hashes de contraseñas de la memoria y se descifraron credenciales de administrador de dominio"). Estos resultados muestran el impacto en cascada: a partir de una vulnerabilidad inicial no tan crítica, quizás se encadenaron acciones que comprometieron todo el entorno. Un auditor interno podría resaltar, por ejemplo, "Escenario de Compromiso Total: aprovechando una combinación de vulnerabilidades A, B y credenciales débiles, se obtuvo control de la totalidad del dominio Windows de la organización". Este tipo de descubrimiento es el peor caso para la seguridad corporativa, pero extremadamente valioso como lección preventiva.

Cada uno de estos resultados se convierte en un hallazgo de auditoría con su descripción, evidencias (logs, capturas de pantalla, código PoC utilizado, etc.), evaluación de impacto y recomendaciones. Por ejemplo, un hallazgo de RCE incluirá la evidencia de la shell obtenida o el comando ejecutado con éxito, y recomendará aplicar el parche correspondiente o aislar el servicio vulnerable. Es importante que el auditor interno detalle claramente qué se logró y cómo, pero también hasta dónde llegó (scope) para que el reporte sea preciso y no alarmista más allá de lo demostrado. Todos estos hallazgos finalmente se usan para mejorar la seguridad, ya sea corrigiendo configuraciones, aplicando actualizaciones, fortificando controles o concienciando al personal.

Capítulo 4

CONFECCIÓN DEL REPORTE

4 CONFECCIÓN DEL REPORTE

El informe de auditoría interna de ciberseguridad es la herramienta clave para documentar y comunicar los hallazgos, conclusiones y recomendaciones de una auditoría. La calidad de este reporte impacta directamente en la capacidad de la organización para corregir debilidades, gestionar riesgos y mejorar la eficiencia.

Objetivo Auditado	Servidor FPT en IP 192.168.0.0.0	
Criterio:	ISO 27002: Controles de Seguridad y Privacidad	
Control:	**Control 5.15 – Establecer políticas y procedimientos para controlar el acceso al servicio FTP y asegurar que solo los usuarios autorizados puedan acceder a el**	
Descripción de la Prueba:	El equipo prueba que el acceso al servidor FTP solicite usuarios y contraseñas, y se gestione mediante cuentas de acceso. ➔ Revisar si es que el servicio permite el acceso de forma anónima Usando **NMAP -p 22 --script ssh-auth-methods --script-args="ssh.user=<usr>" <0.0.0>** o Probar el acceso manual al servicio FTP usando credencial anónima	
Hallazgos:	Riego: Medio	Acceso no controlado permitido a través de credenciales anónimas en el servidor FTP alojado en el servidor en 0.0.0.0; esto es potencialmente dañino porque permite a cualquier usuario acceder a los archivos del servidor FTP. Si los archivos son PII, PHI, PCI o sensibles, entonces el daño puede ser mayor.
	Riesgo: Medio	Acceso público a información técnica de la arquitectura de TI que podría proveer datos sobre como comprometer el sistema.
	Referencias	
Técnica:	Comando y Control	
Táctica:	Protocolo de Transferencia de Archivos / FTP	
Procedimiento:	G0096	
Referencias:	▪ https://www.tenable.com/plugins/nessus/10079 ▪ https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N&version=3.0 ▪ https://www.speedguide.net/port.php?port=21 ▪ https://attack.mitre.org/techniques/T1071/002/	
Recomendación	▪ Si no se usa el servicio FTP entonces deshabilitar. ▪ Solo permitir el acceso a FTP con usuario y contraseña. ▪ Mitigar: Los sistemas de detección y prevención de intrusiones de red que utilizan firmas de red para identificar el tráfico para malware adversario específico se pueden utilizar para mitigar la actividad a nivel de red. ▪ Detectar: Supervisar y analizar los patrones de tráfico y la inspección de paquetes asociados a los protocolos (s), aprovechando la inspección SSL/TLS para el tráfico cifrado, que no sigan las normas de protocolo esperadas y los flujos de tráfico). Considere la correlación con la monitorización del proceso y la línea de comando para detectar la ejecución de procesos anómalos y argumentos de línea de comando asociados a los patrones de tráfico (por ejemplo, monitorear anomalías en el uso de archivos que normalmente no inician conexiones para protocolos respectivos (s))	

4.1 CRITERIO DE AUDITORÍA (NORMAS, POLÍTICAS Y MARCOS APLICABLES)

Conjunto de políticas, procedimientos o requisitos usados como referencia frente a la cual se compara la evidencia de la auditoría. Si los criterios son requisitos legales (incluyendo los reglamentos), los términos "cumple" o "no cumple" se utilizan a menudo en los hallazgos de auditoría. **ISO 19011 - Guía para Auditar los Sistemas de Gestión.**

El criterio de auditoría define con qué normativas, políticas internas o marcos de referencia se comparó el entorno auditado. En este apartado se listan las bases contra las cuales se evaluó la seguridad, por ejemplo: estándares ISO/IEC 27001, NIST CSF, políticas de seguridad de la empresa, procedimientos internos, regulaciones locales, u otros marcos reconocidos. Es importante especificar qué requisitos específicos se utilizaron como referencia. Cada hallazgo posteriormente deberá relacionarse con estos criterios, es decir, indicar qué norma o política no se cumplió o se incumplió parcialmente.

Los resultados del informe deben estar claramente vinculados a los **riesgos identificados y las políticas internas o externas auditadas.** Este nivel de detalle proporciona objetividad y respaldo a los hallazgos, ya que muestra la brecha entre lo esperado y lo encontrado.

Por Ejemplo:

ISO 27002: Controles de Seguridad y Privacidad.

> Control 5.15 – Establecer políticas y procedimientos para controlar el acceso al servicio FTP y asegurar que solo los usuarios autorizados puedan acceder a él.

ATENCIÓN:

Para poder determinar la criticidad de los hallazgos de forma precisa, es necesario conocer los requerimientos de confidencialidad, integridad y disponibilidad del activo que está siendo auditado. Esta información debería estar documentada en el inventario de activos de información, pero, si no lo está. considéralo como una mejora potencial para su reporte.

- **Confidencialidad:**
  ¿Qué nivel de secreto o restricción necesita la información manejada por el activo? Por ejemplo, un servidor que almacena datos personales o secretos comerciales tendrá alta confidencialidad requerida (acceso muy limitado).

- **Integridad:**
  ¿Qué tan crítico es que la información del activo sea exacta y no alterada indebidamente? Por ejemplo, los registros financieros en una base de datos deben tener integridad alta (cualquier modificación no autorizada sería grave).

- **Disponibilidad:**
  ¿Qué nivel de disponibilidad (tiempo activo y accesible) se espera? Por ejemplo, un sistema que soporta operaciones 24/7 de la organización tendrá alta disponibilidad requerida (tolerancia mínima a caídas).

Si el sistema o proceso auditado maneja **información personal identificable (PII)**, el informe debe señalar los **requisitos de privacidad aplicables**. Describa qué tipo de datos personales se procesan (ej. nombres, direcciones, datos financieros, datos sensibles de salud, etc.) y qué **nivel de protección de privacidad** requieren conforme a leyes y políticas.
Por ejemplo: *"El sistema CRM procesa datos de clientes (nombres, RUT, direcciones, historial de compras) y está sujeto a la Ley 19.628 de Protección de Datos Personales, por lo que requiere confidencialidad y medidas de privacidad elevadas."*

4.2 DESCRIPCIÓN DE LA PRUEBA DE AUDITORÍA (PROCEDIMIENTOS REALIZADOS)

En este elemento, detalle el procedimiento de auditoría o prueba que se llevó a cabo para evaluar cada control o detectar cada posible vulnerabilidad. Debe describir cómo el auditor obtuvo la evidencia del hallazgo. Por ejemplo, puede incluir: revisión de documentos y configuraciones, entrevistas realizadas, pruebas técnicas ejecutadas. Sea específico pero conciso: qué se hizo, dónde y con qué propósito.

Esta descripción permite que el lector y los responsables del área entiendan el alcance de la prueba realizada y confíen en que el hallazgo está respaldado por una técnica de auditoría sólida. Documentar los procedimientos ayuda a la repetibilidad: otro auditor podría replicar la prueba siguiendo estos pasos, si fuera necesario.

Para asegurar la repetibilidad, documente con precisión los comandos utilizados y los elementos de configuración específicos de los instrumentos que se están utilizando. A demás, recuerde documentar la versión de las herramientas utilizadas.

Por Ejemplo:

El equipo prueba que el acceso al servidor FTP solicite usuarios y contraseñas, y se gestione mediante cuentas de acceso.

> Revisar si es que el servicio permite el acceso de forma anónima

> Usando NMAP -p 22 --script ssh-auth-methods --script-args="ssh.user=<usr>" <0.0.0>

4.3 DESCRIPCIÓN DEL HALLAZGO (CONDICIÓN OBSERVADA E IMPLICANCIAS)

Documente claramente cada hallazgo identificado. Comienza describiendo la situación observada de forma objetiva y precisa, indicando qué se encontró que representa una desviación del criterio de auditoría. Debe incluir evidencia concreta (descripciones de registros, configuraciones, y en especial, capturas de pantalla relevantes). Las evidencias extensas pueden colocarse en anexos.

ATENCIÓN:

Para resguardar la validez de la información, **sobre todo cuando esta puede ser utilizada como un medio de prueba en un proceso legal**, es importante asegurar su integridad. Para ello, utilice funciones hash sobre la evidencia extraída para crear una huella digital que pueda ser utilizada para comprar la evidencia en un futuro, comparando el hash original con el valor de función hash de la evidencia en cualquier momento.

Ojo: Es importante utilizar un algoritmo hash seguro.

A continuación, explique las implicancias o consecuencias potenciales del hallazgo: ¿por qué es importante?, ¿qué riesgo genera para la organización? Los hallazgos deben presentarse basados en evidencia verificable y sin juicios subjetivos, conectando claramente la condición observada con el riesgo asociado. Un buen hallazgo debería responder ¿Qué pasó? ¿Dónde, cuándo y cuán extendido es el problema? ¿Por qué importa?

Tenga en consideración que el hallazgo debería ser redactado con un lenguaje claro y neutral, con el nivel técnico necesario, pero también suficientemente interpretable por el lector.

Por Ejemplo:

Acceso no controlado permitido a través de credenciales anónimas en el servidor FTP alojado en el servidor en 0.0.0.0; esto es potencialmente dañino porque permite a cualquier usuario acceder a los archivos del servidor FTP. Si los archivos son PII, PHI, PCI o sensibles, entonces el daño puede ser mayor. Se considera un riesgo MODERADO.

Tras describir el hallazgo, proporcione una **valoración** o **evaluación del riesgo** que este representa. Esto típicamente incluye estimar el **impacto** y la **probabilidad** de ocurrencia del riesgo asociado, para finalmente asignar una **categoría o nivel de riesgo**. El **impacto** describe las posibles consecuencias si el hallazgo no se mitiga. Puede abarcar impactos **financieros** (pérdidas económicas, sanciones), **operativos** (interrupción de servicio, ineficiencias), **reputacionales** (daño a la imagen, pérdida de confianza de clientes) e incluso **legales/regulatorios** (multas por incumplimiento de leyes).

4.4 ASOCIACIÓN DEL HALLAZGO CON TÁCTICAS, TÉCNICAS O PROCEDIMIENTOS (TTPS) – MITRE ATT&CK (SI APLICA)

Es muy valioso enriquecer el hallazgo vinculándolo con tácticas, técnicas o procedimientos conocidos de amenazas, por ejemplo usando el marco MITRE ATT&CK. Esto aporta contexto desde la perspectiva del adversario: ayuda a ilustrar cómo podría explotarse la debilidad encontrada o con qué patrones de ataque está relacionada.

El vínculo con las TTP también permite realizar una investigación mucho más profunda, que puede entregar información sobre:

- **Casos históricos documentados donde se ha materializado el riesgo por un adversario.**
- **Consejos para la remediación y detección.**
- **Valoración de las vulnerabilidades asociadas (a través del método CVE).**
- **Una visión contextualizada del ataque.**

Por Ejemplo:

Técnica:	Comando y Control
Táctica:	Protocolo de Transferencia de Archivos / FTP
Procedimiento:	G0096
Referencias:	▪ https://www.tenable.com/plugins/nessus/10079 ▪ https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N&version=3.0 ▪ https://www.speedguide.net/port.php?port=21

Recomendación del Auditor (Acciones Sugeridas)

Después de cada hallazgo, presente la **recomendación** correspondiente, es decir, las **acciones sugeridas** para corregir o mitigar el problema identificado. Una buena recomendación debe ser **específica, práctica y orientada a reducir el riesgo**. Evite formulaciones genéricas; en lugar de "Mejorar la seguridad de la red", detalle qué se debe hacer y dónde. Por ejemplo: "Implementar un proceso de revisión trimestral de cuentas de usuario inactivas y dar de baja inmediatamente aquellas cuentas de ex-empleados", o "Configurar el firewall para restringir el acceso desde internet solo a los puertos indispensables (80/443) y bloquear el resto, conforme a la política de acceso seguro."

Por Ejemplo:

Mitigar: Los sistemas de detección y prevención de intrusiones de red que utilizan firmas de red para identificar el tráfico para malware adversario específico se pueden utilizar para mitigar la actividad a nivel de red.

Detectar: Supervisar y analizar los patrones de tráfico y la inspección de paquetes asociados a los protocolos (s), aprovechando la inspección SSL/TLS para el tráfico cifrado, que no sigan las normas de protocolo esperadas y los flujos de tráfico). Considere la correlación con la monitorización del proceso y la línea de comando para detectar la ejecución de procesos anómalos y argumentos de línea de comando asociados a los patrones de tráfico (por ejemplo, monitorear anomalías en el uso de archivos que normalmente no inician conexiones para protocolos respectivos (s))