

Análisis de seguridad de un servidor web vulnerable mediante ataque de Path Traversal, exposición de claves privadas y conexión remota vía SSH.

Resolución de reto Manolo

Resumen Ejecutivo y Resumen Técnico

Julio de 2025

Presentado por: Alondra J. Jacinto Sanchez

CONTROL DE VERSIONES

Versión	Fecha	Autor	Revisor/Aprobador	Cambios realizados
1.0	26/07/2025	Alondra Jacqueline Jacinto Sanchez		Versión inicial del informe.

ÍNDICE

RESUMEN EJECUTIVO	3
Introducción.....	4
Alcance y limitaciones.....	4
Metodología empleada.....	4
Recomendaciones.....	6
RESUMEN TÉCNICO.....	8
Explotación.....	9
Paso 1. Identificación.....	9
Paso 2. Enumeración de directorios.....	10
Paso 3. Inspección de la web “Manolo”.....	11
Paso 4. Acceso al servidor.....	15
Vulnerabilidades encontradas	16
V01 – Security Misconfiguration.....	16
V02 – Broken Access Control.....	17
Conclusiones.....	18

RESUMEN EJECUTIVO

Introducción

En el presente informe se documenta el análisis de seguridad realizado sobre un entorno web, con el objetivo de evaluar su nivel de exposición frente a ataques reales. El ejercicio se llevó a cabo en un escenario controlado que simula un servidor en producción, implementado en una máquina virtual accesible en red y diseñada para permitir la interacción con usuarios.

Se identificaron debilidades en los mecanismos de validación y filtrado de entradas de usuario, lo que posibilita la enumeración de usuarios y el acceso no autorizado a información sensible. Se han aprovechado estas vulnerabilidades para comprometer el sistema y obtener control parcial del servidor durante el ejercicio de *pentesting*.

El análisis completo, las técnicas empleadas y las recomendaciones de mitigación se desarrollan en las siguientes secciones.

Alcance y limitaciones

El presente análisis se llevó a cabo sobre un entorno controlado, limitado exclusivamente a la máquina virtual proporcionada para el ejercicio. El objetivo fue identificar, explotar y documentar vulnerabilidades en el servicio web y en la configuración del servidor, con el propósito de evaluar su nivel de exposición ante un atacante real.

La prueba se desarrolló bajo un enfoque de **caja negra**, es decir, sin contar con información previa sobre la infraestructura o credenciales de acceso. Esto permite simular el comportamiento de un atacante externo con recursos limitados.

No se evaluaron otros servicios fuera del objetivo definido ni se realizaron acciones que pudieran afectar la disponibilidad del sistema o comprometer la integridad de los datos.

Metodología empleada

El Gráfico 1 resume las **vulnerabilidades** detectadas, valoradas conforme a una escala CVSS¹ aproximada para priorizar su impacto potencial. Estas vulnerabilidades se alinean con categorías nombradas en el **OWASP Top 10 (2021)**, marco de referencia que destaca los principales riesgos de seguridad en aplicaciones web modernas.

¹ El CVSS (*Common Vulnerability Scoring System*) se calculó utilizando la [calculadora oficial de CVSS v3.1](#)

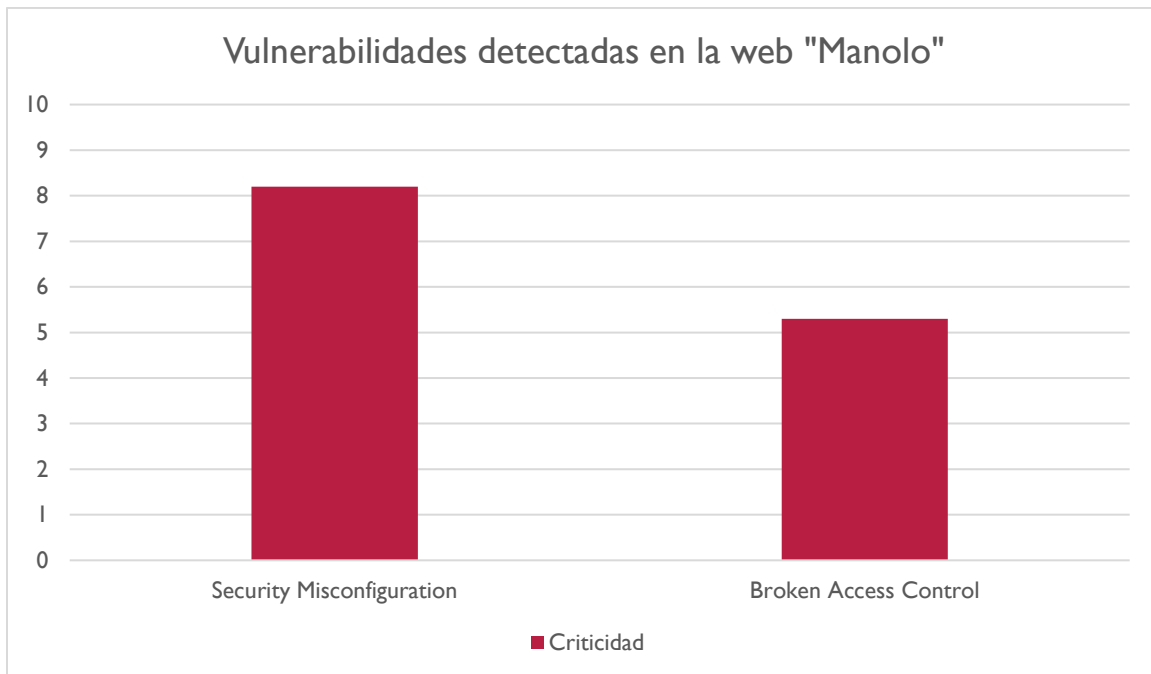


Gráfico 1. Vulnerabilidades detectadas y su nivel de criticidad (escala CVSS aproximada)

La inclusión de **Security Misconfiguration** se justifica por la presencia de configuraciones inseguras, evidenciada por una clave privada SSH almacenada sin controles de acceso adecuados. Por otro lado, **Broken Access Control** aplica debido a la existencia de un parámetro vulnerable en la aplicación web, el cual permitió realizar un ataque de **Path Traversal** y acceder a archivos internos del sistema.

Ambas vulnerabilidades, facilitaron la obtención de información sensible y, en última instancia, el **compromiso del servidor**.

La evaluación de seguridad se llevó a cabo siguiendo el marco **MITRE ATT&CK**, ampliamente reconocido para mapear tácticas, técnicas y procedimientos empleados por actores maliciosos. Este enfoque permitió estructurar las acciones de manera ordenada, simulando el comportamiento real de un atacante, desde el reconocimiento inicial hasta la obtención de acceso al sistema objetivo.

Las fases del ejercicio de *pentesting* se detallan en la Tabla 1.

Acción	Táctica (MITRE)	Técnica (ID)	Descripción
Descubrimiento de host y servicios	<i>Reconnaissance</i>	<i>T1595 – Active Scanning</i>	Se identificó la IP del objetivo, los puertos abiertos, y la ruta de la web “Manolo” ubicada en /football/
Inspección de código fuente	<i>Reconnaissance</i>	<i>T1592 – Gather Victim Identity Information</i>	Se encontró en el HTML una línea comentada que revelaba parámetros internos.
Enumeración de usuarios	<i>Collection</i>	<i>T1005 – Data from Local System</i>	Se explotó un parámetro no sanitizado que permitió consultar a los usuarios existentes en el sistema.
Identificación de rutas sensibles	<i>Reconnaissance</i>	<i>T1083 – File and Directory Discovery</i>	Se realizaron técnicas para enumerar directorios internos
Acceso a claves privadas	<i>Credential Access</i>	<i>T1552.004 – Private Keys</i>	Se recuperó una llave privada dentro de los directorios del servidor
Acceso al sistema	<i>Initial Access</i>	<i>T1078 – Valid Accounts</i>	Se estableció una conexión remota al servidor usando la clave obtenida.

Tabla 1. Resumen de metodología empleada MITRE ATT&CK

Recomendaciones

Basado en los resultados obtenidos en el análisis, se sugieren las siguientes recomendaciones para mitigar los riesgos de seguridad del sistema:

1. Revisión de código fuente y sanitización de entradas

- a. **Justificación:** Sin validaciones adecuadas, un atacante puede manipular parámetros y acceder a recursos no autorizados. La revisión del código y la sanitización de entradas reducen la posibilidad de explotación de fallos lógicos.
- b. **Beneficio esperado:** Mitigación de la vulnerabilidad **Broken Access Control**, evitando ataques como *Path Traversal* y accesos no autorizados a archivos internos.

2. Control de accesos a archivos sensibles y credenciales

- a. **Justificación:** Las claves privadas comprometidas otorgan acceso directo al sistema, lo que representa un riesgo crítico. Limitar los permisos y aislar

estos archivos reduce significativamente la posibilidad de que un atacante los obtenga incluso si compromete la aplicación web.

- b. **Beneficio esperado:** Protección frente a *Security Misconfiguration* y reducción del riesgo de accesos no autorizados.

3. Fortalecimiento de la configuración del servidor web.

- a. **Justificación:** Una configuración insegura puede exponer rutas internas, permitir listados de directorios o filtrar información sensible. Endurecer el servidor minimiza estos puntos débiles, dificultando la fase de reconocimiento de un atacante.

- b. **Beneficio esperado:** Reducción de la superficie de ataque y mayor resistencia ante intrusiones.

4. Revocación y rotación de credenciales comprometidas

- a. **Justificación:** Una vez que una clave privada o credencial ha sido expuesta, debe considerarse irremediablemente comprometida. La revocación y sustitución inmediata evita que un atacante continúe accediendo al sistema con credenciales filtradas.

- b. **Beneficio esperado:** Eliminación de riesgos de accesos persistentes mediante credenciales previamente comprometidas.

RESUMEN TÉCNICO

Explotación

Paso 1. Identificación

Al tratarse de un ejercicio de caja negra, el primer paso consistió en identificar la máquina virtual en la red mediante un escaneo ARP. La Ilustración 1 muestra el resultado de este análisis, en el cual se determinó que la dirección IP del servidor es 10.x.x.16.

```
kali)-[~]
$ sudo arp-scan -l -I eth0
[sudo] password for 
Interface: eth0, type: EN10MB, MAC: , IPv4: 10 6
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10 1 (Unknown: locally administered)
10 2 (Unknown: locally administered)
10 3 (Unknown)
10 16 (Unknown)
4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.922 seconds (133.19 hosts/sec). 4 responded
```

Ilustración 1. Escaneo ARP en la red NAT

Una vez identificado el objetivo, se realizó un escaneo de puertos utilizando la herramienta **Nmap**, activando las siguientes banderas:

- -sV: Permite conocer la versión del servicio que aloja cada puerto.
- -p-: Para escanear todos los puertos del objetivo

La Ilustración 2 detalla el resultado del escaneo, donde se determinó que la máquina virtual aloja un servicio web Apache **HTTP en el puerto 80**, y que permite conexiones remotas mediante **SSH en el puerto 22**.

```
kali)-[~]
$ nmap -sV 10. 16 -p-
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-16 06:42 EDT
Nmap scan report for 10 .16
Host is up (0.00017s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
MAC Address: (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: 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 7.46 seconds
```

Ilustración 2. Escaneo de puertos con Nmap

Posteriormente, al acceder a `http://10.x.x.16:80` desde el navegador, solo se puede ver la página por defecto del servidor, sin exponer ninguna aplicación web interactiva para el usuario, tal como se observa en la Ilustración 3.

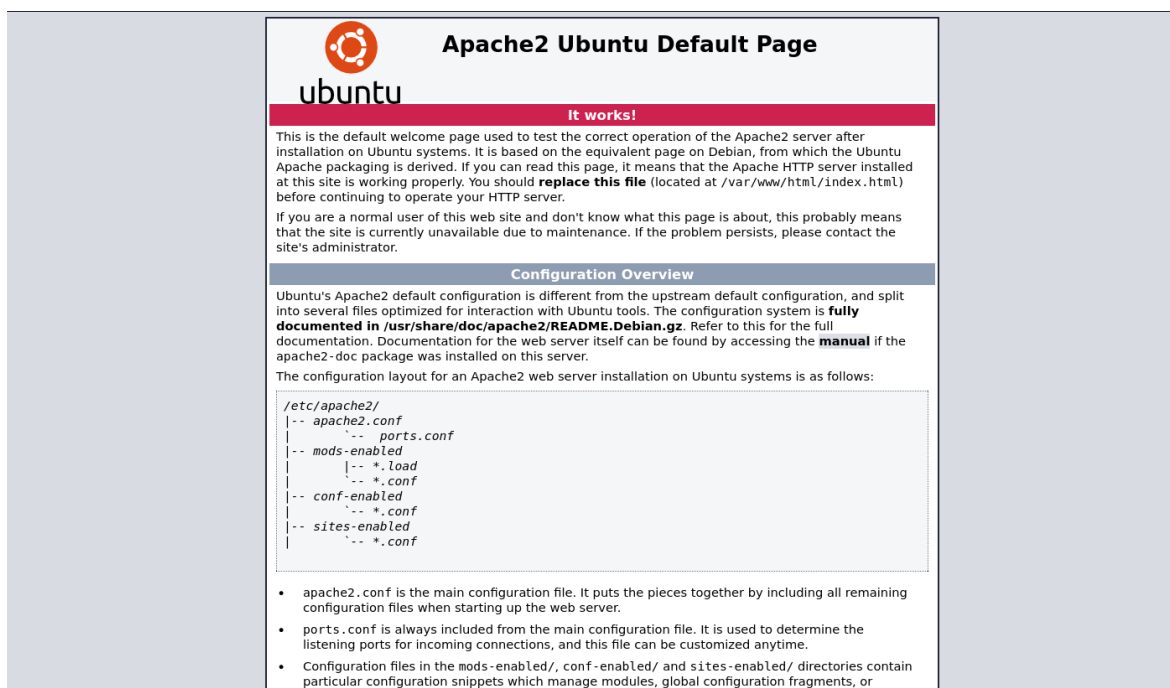


Ilustración 3. Página por defecto del servidor Apache

Paso 2. Enumeración de directorios

Para encontrar posibles alojadas en el servidor, se realizó un *fuzzing* de directorios, para descubrir rutas ocultas o no indexadas. Para ello, se empleó la herramienta **Gobuster** junto con el diccionario **directory-list-2.3-medium**, desarrollado por el proyecto **OWASP**. Este diccionario contiene rutas y nombres de archivos comúnmente empleados en aplicaciones web, facilitando la identificación de recursos ocultos.

Los resultados de esta técnica se presentan en la Ilustración 4, donde se muestran los directorios identificados `/uploads` y `/football`.

```
(kali)-[~]
$ gobuster dir -u http://10.16 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.16
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/uploads (Status: 301) [Size: 308] [→ http://10.16/uploads/]
/football (Status: 301) [Size: 309] [→ http://10.16/football/]
/server-status (Status: 403) [Size: 274]
Progress: 220560 / 220561 (100.00%)

Finished
```

Ilustración 4. Enumeración de directorios web mediante Fuzzing

Paso 3. Inspección de la web “Manolo”

Al acceder a la ruta `http://10.x.x.16/football/`, se visualizó una web estática con información biográfica de un personaje de la cultura española, tal como aparece en la Ilustración 5.

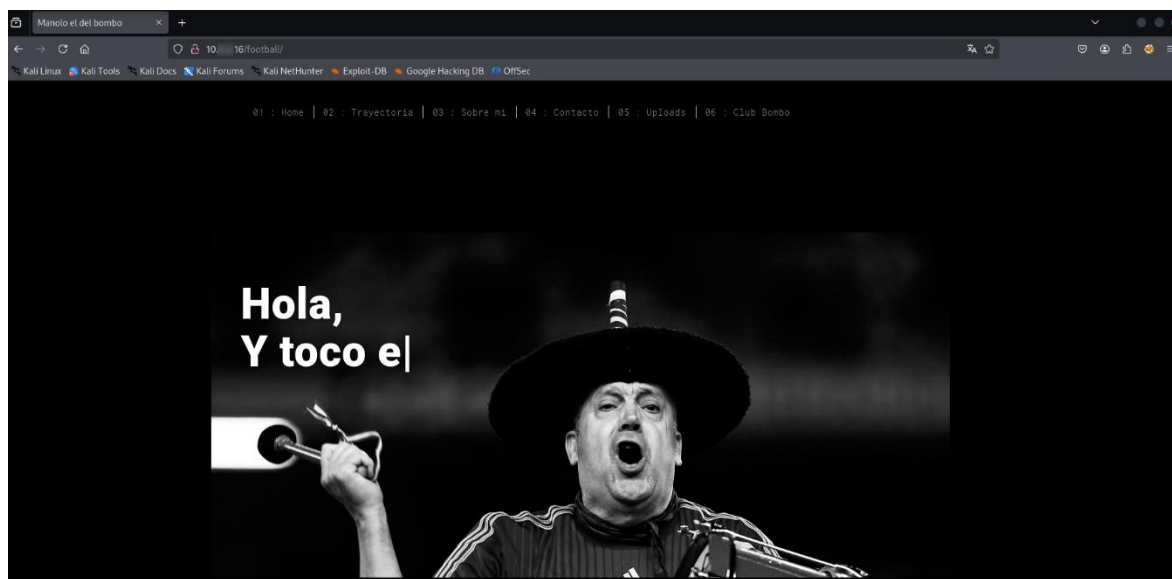


Ilustración 5. Acceso inicial a la web “Manolo”

La web presenta un formulario de inicio de sesión, aunque no dispone de opción de registro. Se realizó un ataque de fuerza bruta sobre este formulario utilizando **BurpSuite** y el diccionario **fasttrack**, incluido en Kali Linux.

Los resultados, mostrados en la Ilustración 6, evidenciaron la existencia de una **contraseña débil** asociada al usuario seleccionado. Dichas credenciales permitieron acceder correctamente al sitio, como se observa en la Ilustración 7.

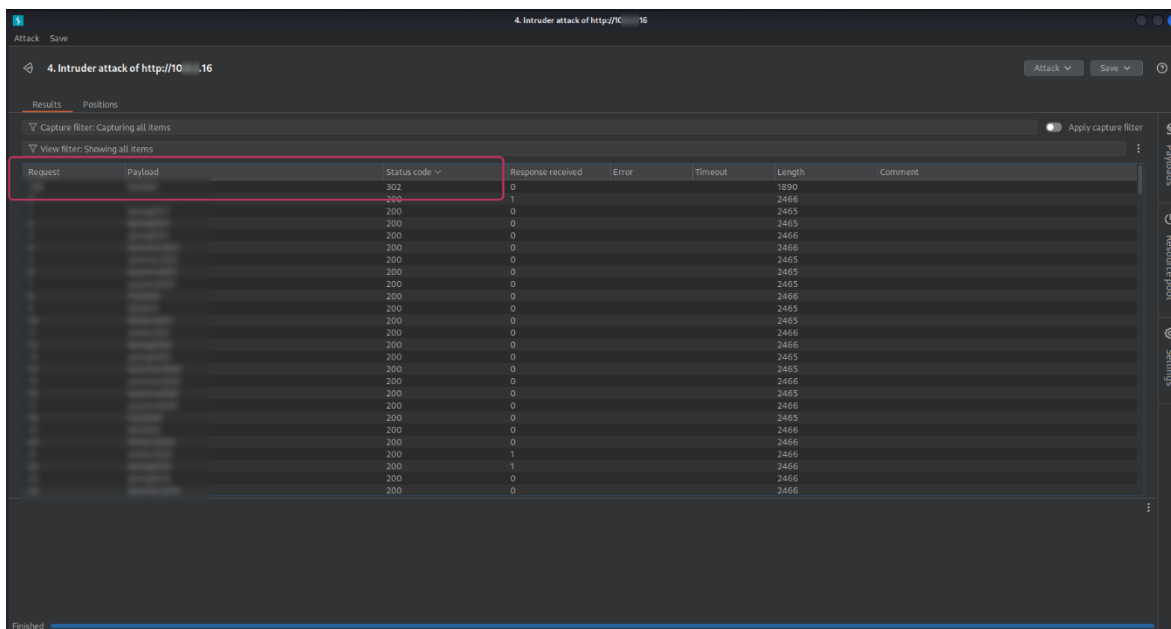


Ilustración 6. Ataque de fuerza bruta utilizando BurpSuite

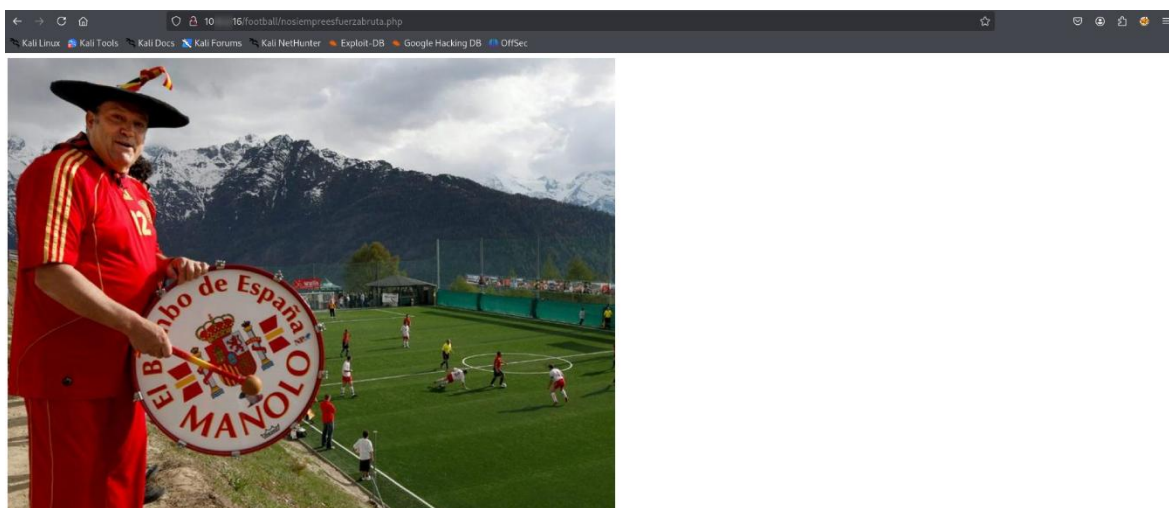


Ilustración 7. Acceso como usuario de la web

Una vez dentro de la sesión autenticada, no se identificaron nuevos vectores de ataque, ya que el área accesible se limitaba a una página estática sin interacción ni funcionalidades adicionales.

Al inspeccionar el código fuente de la página principal ubicada en <http://10.x.x.16/football>, como lo muestra la Ilustración 8, se identificó un comentario HTML, el cual revela

información interna de la aplicación. En dicho comentario se encuentra referenciado un parámetro susceptible a ser manipulado.

```
1 <!DOCTYPE html>
2 <html lang="es">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta content="IE=edge" http-equiv="X-UA-Compatible">
7   <meta content="width=device-width,initial-scale=1" name="viewport">
8
9   <title>Manolo el del bombo</title>
10
11 <link href="/main.3f6952e4.css" rel="stylesheet"></head>
12
13 <body class="minimal">
14 <div id="site-border-left"></div>
15 <div id="site-border-right"></div>
16 <div id="site-border-top"></div>
17 <div id="site-border-bottom"></div>
18 <header>
19   <nav class="navbar navbar-fixed-top navbar-inverse">
20     <div class="container">
21       <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar-collapse" aria-expanded="false">
22         <span class="sr-only">Toggle navigation</span>
23         <span class="icon-bar"></span>
24         <span class="icon-bar"></span>
25         <span class="icon-bar"></span>
26       </button>
27
28       <div class="collapse navbar-collapse" id="navbar-collapse">
29         <ul class="nav navbar-nav">
30           <li><a href="/index.php" title="">01 : Home</a></li>
31           <li><a href="/works.html" title="">02 : Trayectoria</a></li>
32           <!-- <li><a href="index.php?view=about.html" title="">03 : Sobre mi</a></li> -->
33           <li><a href="/about.html" title="">03 : Sobre mi</a></li>
34           <li><a href="/contact.html" title="">04 : Contacto</a></li>
35           <li><a href="/uploads.php" title="">05 : Uploads</a></li>
36           <li><a href="/admin.php" title="">06 : Club Bombo</a></li>
37         </ul>
38
39       </div>
40     </div>
41   </nav>
42 </header>
43 <!-- badum tss -->
44 <div class="hero-full-container background-image-container white-text-container" style="background-image: url('/assets/images/manoloindex.jpg')">
45   <div class="container">
46     <div class="row">
47       <div class="col-xs-12">
48         <div class="hero-full-wrapper">
49           <div class="text-content">
50             <h1>Hola,<br>
51             <span id="typed-strings">
52               <span>soy Manolo,</span>
53               <span>Y tnc el bombo.</span>
54             </span>
```

Ilustración 8. Inspección de código fuente

El parámetro “view” indica que la página carga dinámicamente archivos HTML a través de un script PHP. Esto sugiere la posibilidad de realizar un ataque de Path Traversal, ya que podría ser vulnerable a la manipulación de rutas en el servidor.

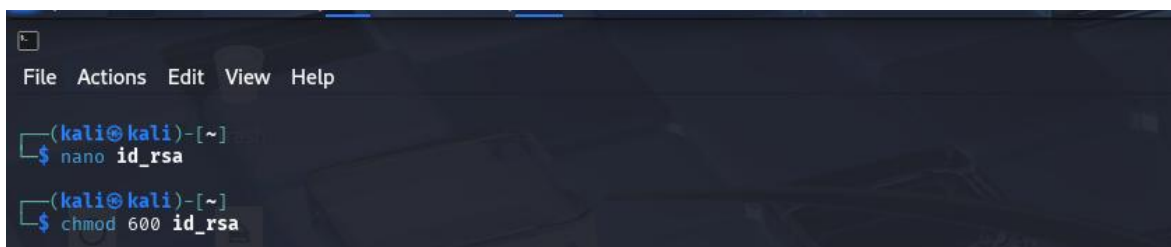
Con esta hipótesis, se probó la manipulación del parámetro “view” directamente en la URL, solicitando archivos del sistema. Tal como se muestra en la Ilustración 9, el servidor devolvió correctamente el contenido del archivo /etc/passwd, confirmando la vulnerabilidad de Path Traversal.

Paso 4. Acceso al servidor

Una vez recuperado el contenido del archivo **id_rsa**, este fue copiado manualmente en la máquina atacante mediante el editor de texto **nano**, con el objetivo de guardarlo como un archivo local. Posteriormente, se modificaron los permisos del archivo utilizando el comando **chmod**, con el fin de garantizar su funcionamiento correcto con el cliente SSH.

Todo este proceso se documenta en la Ilustración 11.

Finalmente, se estableció una **conexión SSH exitosa** con el servidor utilizando la clave obtenida, comprometiendo así la máquina del servidor, tal como se muestra en la Ilustración 12.



```
File Actions Edit View Help
(kali@kali)-[~]
$ nano id_rsa
(kali@kali)-[~]
$ chmod 600 id_rsa
```

Ilustración 11. Configuración de llave privada obtenida

```
kali)-[~]  
$ ssh -i id_rsa @10 .16  
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-90-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Wed 16 Jul 2025 11:04:44 AM UTC  
  
System load:  0.0          Processes:            120  
Usage of /:   53.7% of 8.79GB Users logged in:          0  
Memory usage: 28%         IPv4 address for enp0s3: 10 .16  
Swap usage:   0%  
  
* Super-optimized for small spaces - read how we shrank the memory  
  footprint of MicroK8s to make it the smallest full K8s around.  
  
https://ubuntu.com/blog/microk8s-memory-optimisation  
  
0 updates can be applied immediately.  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
Last login: Thu Jul 10 19:10:51 2025 from 10 .6  
@retaso:~$ whoami
```

Ilustración 12. Acceso remoto al servidor utilizando una llave privada

Vulnerabilidades encontradas

V01 – Security Misconfiguration

Descripción: El servidor permite el acceso a archivos sensibles debido a una configuración inapropiada del sistema de archivos y de los permisos asociados al servidor web. Durante el análisis se identificó que una **llave privada SSH** se encuentra almacenada sin restricciones dentro del directorio personal de un usuario local, y es accesible mediante un vector de **Path Traversal** desde la aplicación web.

Vector de ataque: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:H/A:N → Criticidad: 8.2 (Alta)

Impacto: La exposición de claves privadas permite a un atacante obtener acceso remoto vía SSH al servidor objetivo. Esto implica un **compromiso completo del sistema** con persistencia.

Justificación: Este tipo de error representa un fallo de seguridad crítico, ya que involucra credenciales reutilizables, control remoto sobre el sistema y evidencia una falla de políticas de protección de información sensible en entornos accesibles por servicios web.

Recomendación:

- Aislar archivos sensibles fuera del alcance del servidor web
- Establecer una política de permisos mínima para usuarios del servidor
- Implementar mecanismos de control de acceso y validación en el backend para evitar exposición de rutas del sistema
- Eliminar o actualizar inmediatamente todas las credenciales comprometidas.

Explotación: Mediante el parámetro vulnerable **“view”**, se accedió a la ruta desde donde se extrajo una llave privada. Con la información obtenida, es posible establecer una sesión SSH con el servidor. (Véase Paso 3. Inspección de la web “Manolo” y Paso 4. Acceso al servidor).

V02 – Broken Access Control

Descripción: Se identificó un fallo en el control de acceso del parámetro view, presente en la URL de la aplicación web. Este parámetro, utilizado para cargar contenido dinámico, no contaba con mecanismos de validación adecuados, lo que permitió ejecutar un ataque de **Path Traversal** para acceder a archivos arbitrarios fuera del directorio raíz de la aplicación.

Vector de ataque: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N → Criticidad: 5.3 (Media)

Impacto: Es posible acceder a archivos del sistema como **/etc/passwd**, lo que permite identificar usuarios válidos del sistema operativo. Esta información fue clave durante el ejercicio para localizar rutas sensibles y escalar el ataque.

Justificación: La posibilidad de acceder a archivos internos mediante la manipulación de un parámetro vulnerable representa un fallo grave. Aunque no implicó por sí solo un compromiso total, facilitó una cadena de ataque que derivó en la exposición de credenciales y control del sistema.

Recomendación:

- Validar y sanitizar los parámetros de entrada en el lado del servidor
- Eliminar referencias internas o comentadas dentro del código fuente HTML

Explotación: En el código fuente de la página principal se encontró un comentario HTML con una referencia al parámetro “**view**”. Al modificar su valor con rutas relativas fue posible acceder a archivos del sistema, confirmando la ausencia de controles de acceso adecuados. (Véase Paso 3. Inspección de la web “Manolo”).

Conclusiones

El análisis de seguridad realizado evidenció fallos significativos en la configuración y control de accesos de la aplicación web evaluada. Las vulnerabilidades identificadas, clasificadas como **Security Misconfiguration** y **Broken Access Control** (ambas incluidas en el OWASP Top 10 - 2021), permitieron a un atacante simular acciones reales, como la exposición de archivos críticos y el acceso remoto no autorizado al sistema.

A través de una secuencia lógica de reconocimiento, explotación y abuso de funciones mal protegidas, fue posible obtener una **clave privada SSH**, establecer una sesión remota con el servidor y comprometer la máquina objetivo.

Estas debilidades no solo representan un riesgo técnico alto, sino que reflejan **una falta de controles defensivos fundamentales** en el desarrollo e implementación de la aplicación, tales como la validación de entradas, el aislamiento de recursos sensibles y el endurecimiento del entorno de ejecución.

La aplicación de las recomendaciones descritas en este informe contribuirá significativamente a **reducir la superficie de ataque, elevar la resiliencia frente a amenazas comunes, y alinear el entorno con buenas prácticas de seguridad web**. Se recomienda realizar una revisión periódica de seguridad, tanto a nivel de aplicación como de infraestructura, para mantener un entorno robusto frente a futuras amenazas.