Informe de ejemplo con datos ficticios

GROOTITECH · AUDITORÍA DE SEGURIDAD WEB

Informe de auditoría de seguridad web

ejemplo-cliente.es

Cliente	Ejemplo Cliente, S.L. (empresa ficticia)
Objetivo	https://ejemplo-cliente.es y https://www.ejemplo-cliente.es
Periodo de pruebas	5 y 6 de octubre de 2026
Fecha del informe	8 de octubre de 2026
Versión	1.0
Autor	Alejandro Sánchez Moreno (eJPTv2) · contacto@grootitech.es
Clasificación	Confidencial: solo para el cliente

Todos los nombres, direcciones IP, fechas y resultados de este documento son inventados y sirven solo para mostrar la estructura y el nivel de detalle del informe. Las IP pertenecen a rangos reservados para documentación.

1. Resumen ejecutivo

Entre el 5 y el 6 de octubre de 2026 se revisó desde Internet la web corporativa de Ejemplo Cliente, S.L., con autorización escrita y sin credenciales de acceso, para identificar configuraciones que faciliten un ataque.

Se han encontrado **6 hallazgos**. El más relevante es que el panel de administración del gestor de contenidos está accesible desde cualquier lugar y sin límite de intentos de acceso, lo que permite ataques de fuerza bruta contra las cuentas de administrador. El resto son configuraciones del servidor web que se corrigen en pocas horas: protocolos TLS antiguos activos, cabeceras de seguridad ausentes, una cookie de sesión sin las protecciones habituales y la versión del servidor visible.

0	1	3	2	0
Crítico	Alto	Medio	Bajo	Informativo

Prioridad recomendada: restringir el acceso al panel de administración esta misma semana (H-01) y aplicar el resto de correcciones en la siguiente ventana de mantenimiento. Tras las correcciones se incluye una nueva comprobación de los puntos afectados sin coste adicional.

2. Alcance y autorización

- **Incluido:** https://ejemplo-cliente.es y https://www.ejemplo-cliente.es (IP 203.0.113.45), puertos TCP expuestos de esa IP y formularios públicos.
- **Excluido:** otros subdominios, el servidor de correo, ingeniería social, ataques de denegación de servicio y cualquier prueba con usuarios reales.
- **Autorización:** documento firmado por la gerencia de Ejemplo Cliente, S.L. el 1 de octubre de 2026, con el alcance anterior, las fechas, el horario (9:00–18:00) y un contacto técnico para incidencias.
- **Tipo de prueba:** caja negra, desde Internet, sin credenciales.

3. Metodología

Las pruebas siguen las fases de la OWASP Web Security Testing Guide (WSTG) aplicables a una revisión externa:

1. **Reconocimiento pasivo:** DNS, certificados y tecnologías visibles en las respuestas.
2. **Configuración del servidor:** protocolos y cifrados TLS, cabeceras de seguridad, versiones expuestas y puertos abiertos.
3. **Gestión de sesiones:** atributos de las cookies.

4. **Superficie de administración:** rutas de administración conocidas y controles de acceso a ellas.
5. **Verificación manual** de cada hallazgo para descartar falsos positivos.

Cada hallazgo se valora según su probabilidad e impacto con la escala del Anexo B. No se ha modificado ni extraído ningún dato del cliente.

4. Tabla de hallazgos

ID	Hallazgo	Riesgo	Estado
H-01	Panel de administración accesible desde Internet sin límite de intentos	Alto	Abierto
H-02	TLS 1.0 y TLS 1.1 habilitados	Medio	Abierto
H-03	Cookie de sesión sin atributos Secure ni HttpOnly	Medio	Abierto
H-04	Content-Security-Policy ausente	Medio	Abierto
H-05	Strict-Transport-Security (HSTS) ausente	Bajo	Abierto
H-06	Versión del servidor web expuesta en las cabeceras	Bajo	Abierto

5. Detalle de los hallazgos

H-01 · Panel de administración accesible desde Internet Alto

Referencia: OWASP WSTG-CONF-05 · WSTG-ATHN-03

Descripción. La ruta `/wp-admin/` redirige al formulario de acceso del gestor de contenidos, que responde desde cualquier IP. Tras 20 intentos fallidos consecutivos con un usuario inexistente, el formulario sigue aceptando intentos sin bloqueo, retardo ni captcha.

Evidencia.

```
$ curl -sI https://ejemplo-cliente.es/wp-admin/
HTTP/1.1 302 Found
Location: https://ejemplo-cliente.es/wp-login.php?redirect_to=...

$ for i in $(seq 1 20); do curl -s -o /dev/null -w "%{http_code}\n" \
    -d "log=prueba-auditoria&pwd=incorrecta$i" https://ejemplo-cliente.es/wp-login.php;
done | sort | uniq -c
     20 200
```

Impacto. Un atacante puede probar contraseñas de forma automatizada contra las cuentas de administrador. Si acierta, controla la web: puede cambiar contenidos, insertar código malicioso para los visitantes o acceder a los datos de los formularios.

Corrección.

- Limitar el acceso a `/wp-admin/` y `/wp-login.php` a las IP de la oficina o a una VPN.
- Activar el bloqueo tras varios intentos fallidos (por ejemplo, fail2ban o un plugin de limitación de accesos) y el doble factor para todas las cuentas de administrador.

```
# Nginx: solo la IP de la oficina puede llegar al acceso de administración
location ~ ^/(wp-admin|wp-login\.php) {
    allow 198.51.100.20;
    deny  all;
    # ... resto de la configuración de PHP
}
```

H-02 · TLS 1.0 y TLS 1.1 habilitados Medio

Referencia: OWASP WSTG-CRYP-01 · RFC 8996

Descripción. El servidor acepta conexiones con TLS 1.0 y TLS 1.1, protocolos retirados oficialmente desde 2021.

Evidencia.

```
$ nmap --script ssl-enum-ciphers -p 443 ejemplo-cliente.es
PORT    STATE SERVICE
443/tcp open  https
| ssl-enum-ciphers:
|   TLSv1.0:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|   TLSv1.1:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|_  least strength: A
```

Impacto. Permite que un atacante en la misma red fuerce el uso de un protocolo débil con clientes antiguos. Además, incumple los requisitos de PCI DSS y las recomendaciones del CCN-CERT y empeora la valoración en herramientas externas.

Corrección. Dejar solo TLS 1.2 y 1.3 y reiniciar el servicio.

```
# Nginx
ssl_protocols TLSv1.2 TLSv1.3;
```

H-03 · Cookie de sesión sin Secure ni HttpOnly Medio

Referencia: OWASP WSTG-SESS-02

Descripción. La tienda en línea establece la cookie de sesión `PHPSESSID` sin los atributos `Secure`, `HttpOnly` ni `SameSite`.

Evidencia.

```
$ curl -sI https://ejemplo-cliente.es/tienda/ | grep -i set-cookie
Set-Cookie: PHPSESSID=3f9a1c0e8b7d4a62e5c1b0f9d8e7a6b5; path=/
```

Impacto. Sin `HttpOnly`, un fallo de XSS permitiría leer la cookie y suplantar la sesión del cliente. Sin `Secure`, la cookie podría enviarse por HTTP si el usuario llega a una dirección sin cifrar.

Corrección.

```
; php.ini
session.cookie_secure   = 1
session.cookie_httponly = 1
session.cookie_samesite = "Lax"
```

H-04 · Content-Security-Policy ausente Medio

Referencia: OWASP WSTG-CONF-12

Descripción. Ninguna respuesta incluye la cabecera `Content-Security-Policy`.

Evidencia.

```
$ curl -sI https://ejemplo-cliente.es/ | grep -i content-security-policy
(sin resultados)
```

Impacto. Si aparece un fallo de inyección de código (por ejemplo, en un plugin), el navegador ejecutará cualquier script inyectado sin restricciones.

Corrección. Publicar primero la política en modo informe para detectar qué recursos necesita la web y, cuando no haya avisos, pasarla a modo bloqueo.

```
# Nginx (fase 1: solo informa)
add_header Content-Security-Policy-Report-Only "default-src 'self'; img-src 'self' dat
a:; frame-ancestors 'self'" always;
```

H-05 · HSTS ausente Bajo

Referencia: OWASP WSTG-CONF-07

Descripción. La web redirige de HTTP a HTTPS, pero no envía la cabecera `Strict-Transport-Security`.

Evidencia.

```
$ curl -sI http://ejemplo-cliente.es/ | head -2
HTTP/1.1 301 Moved Permanently
Location: https://ejemplo-cliente.es/

$ curl -sI https://ejemplo-cliente.es/ | grep -i strict-transport-security
(sin resultados)
```

Impacto. La primera petición de cada visita puede viajar sin cifrar y ser interceptada en una red no confiable (por ejemplo, una Wi-Fi pública) antes de la redirección.

Corrección.

```
# Nginx, solo en el bloque HTTPS
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
```

H-06 · Versión del servidor expuesta Bajo

Referencia: OWASP WSTG-INFO-02

Descripción. Las respuestas indican la versión exacta del servidor web y de PHP.

Evidencia.

```
$ curl -sI https://ejemplo-cliente.es/
HTTP/1.1 200 OK
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/7.4.3
```

Impacto. Facilita a un atacante buscar vulnerabilidades conocidas de esas versiones concretas. Además, PHP 7.4 dejó de recibir actualizaciones de seguridad en 2022 y Apache 2.4.41 es muy anterior a la versión actual de la rama 2.4, por lo que conviene planificar su actualización.

Corrección.

```
# Apache (security.conf)
ServerTokens Prod
ServerSignature Off

; php.ini
expose_php = Off
```

6. Conclusiones y próximos pasos

1. Restringir el acceso al panel de administración y activar el doble factor (H-01).
2. Ajustar la configuración del servidor: TLS, cabeceras y cookies (H-02 a H-06). Se estima medio día de trabajo.
3. Planificar la actualización de Apache y PHP a versiones con soporte.
4. Nueva comprobación de los hallazgos corregidos y entrega de la versión 1.1 de este informe.

Anexo A. Herramientas utilizadas

Herramienta	Uso
curl	Peticiones manuales y revisión de cabeceras y cookies.
Nmap (con el script ssl-enum-ciphers)	Puertos abiertos y protocolos y cifrados TLS.
testssl.sh	Comprobación cruzada de la configuración TLS y del certificado.
OWASP ZAP (modo pasivo)	Revisión de las respuestas mientras se navega por la web.
Navegador con herramientas de desarrollo	Verificación manual de formularios, cookies y redirecciones.

Anexo B. Escala de riesgo

Nivel	Criterio
Crítico	Explotable de forma directa y con impacto grave inmediato (por ejemplo, ejecución de código o acceso a datos sin autenticación).
Alto	Explotable con poco esfuerzo y con impacto importante sobre la web o sus datos.
Medio	Requiere condiciones adicionales o tiene un impacto limitado, pero debe corregirse.
Bajo	Facilita otros ataques o da información útil a un atacante.
Informativo	Buena práctica recomendada sin riesgo directo.

Informe de ejemplo con datos ficticios