

Informe técnico de Auditoría y Bastionamiento del sitio web

(Basado en OWASP Server Hardening Guide, CIS Benchmarks y NIST SP 800-123)



Autor: Rafael M. Pérez | Administrador de sistemas & Analista en Ciberseguridad

Fecha: Octubre 2025

Versión del informe: 1.0

Herramientas de auditoría y bastionado utilizadas:

Cloudflare, Nikto, SSL Labs, Security Headers, testssl.sh, SSLyze, OWASP ZAP, Nuclei

Sitio web bastionado: rafaelperez.com

Objetivo: Fortalecer la seguridad del sitio web rafaelperez.com mediante la implementación de medidas de bastionamiento en la capa web y de transporte, aplicando buenas prácticas basadas en la OWASP Server Hardening Guide. El propósito es reducir la superficie de ataque, garantizar la integridad de las comunicaciones y mejorar la resiliencia del sitio frente a amenazas comunes sin afectar su disponibilidad ni rendimiento.

Informe Técnico Hardening Web

Sysadminweb@proton.me

Portafolio Autor | rafaelperez.com





Índice

Portada

1. Introducción.....	3
2. Tecnologías utilizadas.....	3
3. Pasos realizados.....	4
4. Conclusiones y recomendaciones finales.....	15
5. Tabla resumen de herramientas utilizadas.....	17
6. Anexo técnico.....	18

Proyecto: Bastionamiento del sitio web



1. Introducción:

El principal reto consistía en transformar un sitio web funcional pero vulnerable en una plataforma resiliente frente a amenazas comunes y avanzadas. Con la configuración predeterminada del servidor y la ausencia de políticas de seguridad explícitas, el sitio quedaba expuesto a múltiples vectores de ataque, como la suplantación de contenido, la interceptación de datos y la manipulación de cabeceras HTTP.

Además, no se contaba con mecanismos de auditoría ni visibilidad sobre el estado real de la infraestructura.

Este desafío exigía una estrategia progresiva que permitiera bastionar el entorno sin comprometer la disponibilidad ni la experiencia del usuario, integrando herramientas de escaneo y análisis que ofrecieran una visión clara y continua del nivel de seguridad alcanzado en cada fase.

2. Tecnologías utilizadas:



Cloudflare Plataforma de seguridad y rendimiento que protege el servidor mediante mitigación de ataques DDoS, firewall de aplicaciones web (WAF), gestión de certificados SSL y políticas de acceso.

Nikto Escáner de vulnerabilidades que se ejecuta desde una máquina cliente para detectar configuraciones inseguras, archivos expuestos y cabeceras mal implementadas en el servidor web.

SSL Labs Plataforma de análisis profundo de configuración TLS/SSL que permite evaluar la calidad del cifrado, la compatibilidad con navegadores modernos y la presencia de cabeceras como HSTS. Ideal para validar la robustez del canal seguro.

Security Headers Herramienta que analiza las cabeceras HTTP de seguridad implementadas en el sitio, como X-Frame-Options, Content-Security-Policy y Strict-Transport-Security, ayudando a detectar omisiones o configuraciones débiles.



testssl.sh Script local de escaneo TLS que permite detectar vulnerabilidades criptográficas conocidas (Heartbleed, ROBOT, POODLE, etc.), validar protocolos activos y revisar la calidad de las suites de cifrado ofrecidas por el servidor.

3. Pasos realizados:

Fase 0: Inicio desde configuración predeterminada:

El proceso de bastionamiento comienza desde una instalación limpia, sin reglas personalizadas ni configuraciones avanzadas. Se parte de la configuración predeterminada de Github Pages, lo que permite documentar cada ajuste de seguridad desde su origen y evaluar su impacto de forma progresiva.

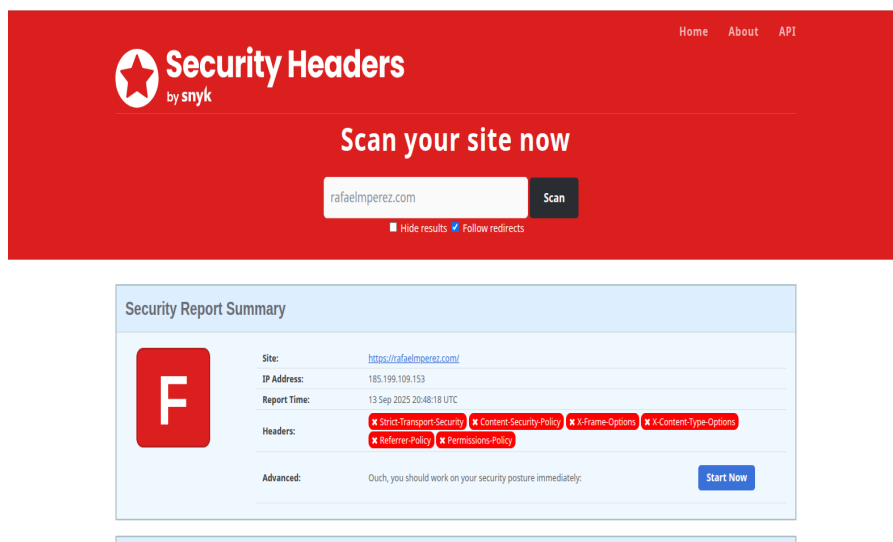
El sitio web se encuentra alojado en Github Pages, sin reglas de firewall, certificados personalizados ni optimizaciones activas. El servidor web mantiene su configuración por defecto, sin cabeceras HTTP de seguridad ni políticas de acceso definidas. Los registros DNS se limitan a los básicos (A, CNAME, MX), sin proxy ni enrutamiento inteligente, y el modo SSL/TLS se encuentra en estado flexible para facilitar auditorías iniciales. No hay herramientas activas de escaneo, monitoreo o protección adicional.

El objetivo de esta fase es establecer un punto de partida claro y documentado para aplicar medidas de bastionamiento. Se busca capturar el estado original del sitio, identificar vulnerabilidades básicas y preparar el entorno para la implementación progresiva de políticas de seguridad.

Fase 1: Auditorías de seguridad:

Esta fase tiene como objetivo identificar vulnerabilidades, configuraciones inseguras y debilidades criptográficas en el sitio web rafaelperez.com. Se han utilizado herramientas especializadas para evaluar distintos aspectos de la seguridad, desde cabeceras HTTP hasta cifrado SSL/TLS. Los resultados obtenidos servirán como punto de partida para aplicar medidas de bastionado en fases posteriores.

Security Headers:



Utilicé la herramienta web [Security Headers](https://securityheaders.com/?q=rafaelperez.com) para evaluar la implementación de cabeceras HTTP de seguridad en el sitio web. Esta herramienta analiza si el servidor incluye cabeceras que protegen contra ataques comunes como XSS, clickjacking, fuga de datos y ejecución de contenido malicioso.

<https://securityheaders.com/?q=rafaelperez.com>

El análisis arrojó una calificación **F**, indicando una postura de seguridad débil. Aunque algunas cabeceras están presentes, muchas están mal configuradas o ausentes. Entre las más críticas se encuentran:

Strict-Transport-Security: ausente. Recomendado: max-age=31536000; includeSubDomains.

Content-Security-Policy: presente pero débil. Debe restringir fuentes externas para prevenir XSS.

X-Frame-Options: ausente. Recomendado: SAMEORIGIN para evitar clickjacking.

X-Content-Type-Options: ausente. Recomendado: nosniff para evitar MIME-sniffing.

Referrer-Policy: ausente. Controla la información compartida al navegar entre sitios.

Permissions-Policy: ausente. Limita el acceso a APIs del navegador como cámara, micrófono, etc.

Además, se detectó una política CORS muy permisiva (access-control-allow-origin: *), lo cual puede exponer el sitio a riesgos si no se gestiona adecuadamente. Se recomienda revisar y aplicar una configuración más estricta y personalizada de cabeceras para mejorar la resiliencia del sitio frente a amenazas comunes.



Nikto:

```
+ Target Port: 443
+-----+
+ SSL Info: Subject: /CN=*.github.io
            Ciphers: TLS_AES_128_GCM_SHA256
            Issuer: /C=GB/ST=Greater Manchester/L=Salford/O=Sectigo Limited/CN=Sectigo RSA Domain Validation Secure Server CA
+ Start Time: 2025-09-13 23:01:20 (GMT2)
+-----+
+ Server: GitHub.com
+ Retrieved via header: 1.1 varnish
+ Retrieved x-served-by header: cache-toj-let02350040-T0J
+ Server leaks inodes via ETags, header found with file /, fields: 0x68c5daf1 0xe7c9
+ The anti-clickjacking X-Frame-Options header is not present.
+ Uncommon header 'x-served-by' found, with contents: cache-toj-let02350040-T0J
+ Uncommon header 'x-cache' found, with contents: HIT
+ Uncommon header 'x-fastly-request-id' found, with contents: f7897d779f6940f354d2fc5eb627ecb7f5948fb1
+ Uncommon header 'x-cache-hits' found, with contents: 1
+ Uncommon header 'access-control-allow-origin' found, with contents: *
+ Uncommon header 'x-timer' found, with contents: S1757797281.216890,V50,VE2
+ Uncommon header 'x-github-request-id' found, with contents: AAE2:280C3B:220554C:224738A:68C50B9F
+ Uncommon header 'x-proxy-cache' found, with contents: MISS
+ Uncommon header 'content-security-policy' found, with contents: default-src 'none'; style-src 'unsafe-inline'; img-src data; connect-src 'self'
+ Uncommon header 'x-origin-cache' found, with contents: HIT
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ "robots.txt" retrieved but it does not contain any 'disallow' entries (which is odd).
+ Server banner has changed from 'GitHub.com' to 'Varnish' which may suggest a WAF, load balancer or proxy is in place
+ Server is using a wildcard certificate: '*.github.io'
+ Hostname 'rafaelperez.com' does not match certificate's CN '*.github.io'
```

Seguidamente utilicé la herramienta [Nikto](#) para realizar un escaneo de seguridad sobre el sitio web. Nikto permite detectar configuraciones inseguras, cabeceras mal implementadas, archivos expuestos y comportamientos anómalos del servidor web.

```
nikto -h https://rafaelperez.com -p 443
```

El escaneo reveló varios aspectos que requieren atención. Entre los hallazgos más relevantes se encuentran:

Cabeceras de seguridad ausentes: como X-Frame-Options y X-Content-Type-Options, lo que aumenta el riesgo de clickjacking y sniffing de contenido.

Certificado SSL no coincidente: el dominio utiliza un certificado wildcard para *.github.io, lo cual no valida correctamente rafaelperez.com.

Fuga de metadatos vía ETag: el servidor expone identificadores internos que podrían facilitar fingerprinting.

Cabeceras inusuales: como x-fastly-request-id, x-github-request-id y x-proxy-cache, que revelan detalles del entorno de alojamiento.

robots.txt sin restricciones: lo que permite indexación completa por parte de motores de búsqueda.

Servidor detrás de proxy o WAF: el banner del servidor cambia entre GitHub.com y Varnish, lo que indica la presencia de un balanceador de carga o firewall de aplicaciones.

Estos hallazgos sugieren que el sitio carece de medidas básicas de protección en su capa HTTP y presenta una configuración SSL poco robusta. Se recomienda implementar cabeceras de seguridad, revisar el certificado y limitar la exposición de metadatos para fortalecer la postura de seguridad.



testssl.sh:

```

#openssl s_client -connect https://rafaelmperez.com
Cookie(s) (none issued at '/')
Security headers
Access-Control-Allow-Origin: *
X-Served-By: cache-mad22043-MAD
Cache-Control: max-age=600
Reverse Proxy banner
Via: 1.1 varnish
X-Cache: HIT
X-Cache-Hits: 1

Testing vulnerabilities
Heartbleed (CVE-2014-0160) not vulnerable (OK), no heartbeat extension
CCS (CVE-2014-0224) not vulnerable (OK)
Ticketbleed (CVE-2016-9244), experimental not vulnerable (OK)
Opussum (CVE-2025-49812) not vulnerable (OK)
ROBOT not vulnerable (OK)
Secure Renegotiation (RFC 5746) supported (OK)
Secure Client-Initiated Renegotiation not vulnerable (OK) -- mitigated (1 successful renegotiation within 10 in 33s(timeout))
CRIME, TLS (CVE-2012-4929) not vulnerable (OK)
BREACH (CVE-2013-3587) potentially not ok: 'gzip' HTTP compression detected, -- only supplied '/' tested
Can be ignored for static pages or if no secrets in the page
POODLE, SSL (CVE-2014-3566) not vulnerable (OK), no SSLv3 support
TLS_FALLBACK_SCSV (RFC 7507) no fallback possible (OK), no protocol below TLS 1.2 offered
SWEET32 (CVE-2016-2183, CVE-2016-6329) not vulnerable (OK)
FREAK (CVE-2015-0204) not vulnerable (OK)
DROWN (CVE-2016-0808, CVE-2016-0703) not vulnerable on this host and port (OK)
make sure you don't use this certificate elsewhere with SSLv2 enabled services, see
https://search.censys.io/search?resource=hosts&virtual_hosts=INCLUDE&q=E8BA7394EBC4C778BCDAEF249B73906E7523FDEF180B
915AB8C13A2A41C2ACB03
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
BEAST (CVE-2011-3309) not vulnerable (OK), no SSL3 or TLS1
LUCKY13 (CVE-2013-0169), experimental potentially VULNERABLE, uses cipher block chaining (CBC) ciphers with TLS. Check patches
Winshock (CVE-2014-6321), experimental not vulnerable (OK)
RC4 (CVE-2013-2566, CVE-2015-2808) no RC4 ciphers detected (OK)

Running client simulations (HTTP) via sockets

```

Posteriormente utilicé la herramienta [testssl.sh](https://github.com/dargaz/testssl.sh) para realizar un escaneo técnico profundo del canal HTTPS del sitio web. Esta herramienta permite detectar vulnerabilidades conocidas, evaluar la configuración de cifrado y verificar el soporte de protocolos SSL/TLS.

```
./testssl.sh https://rafaelmperez.com
```

El escaneo reveló que el servidor no es vulnerable a ataques conocidos como:

Heartbleed

POODLE

FREAK

DROWN

CRIME

CCS Injection

También se confirmó el soporte para protocolos modernos como TLS 1.2 y TLS 1.3, y el uso de cipher suites seguras como TLS_AES_128_GCM_SHA256. No obstante, se detectaron aspectos que requieren atención:

Cabeceras laxas: como access-control-allow-origin: *, que permiten solicitudes desde cualquier origen.

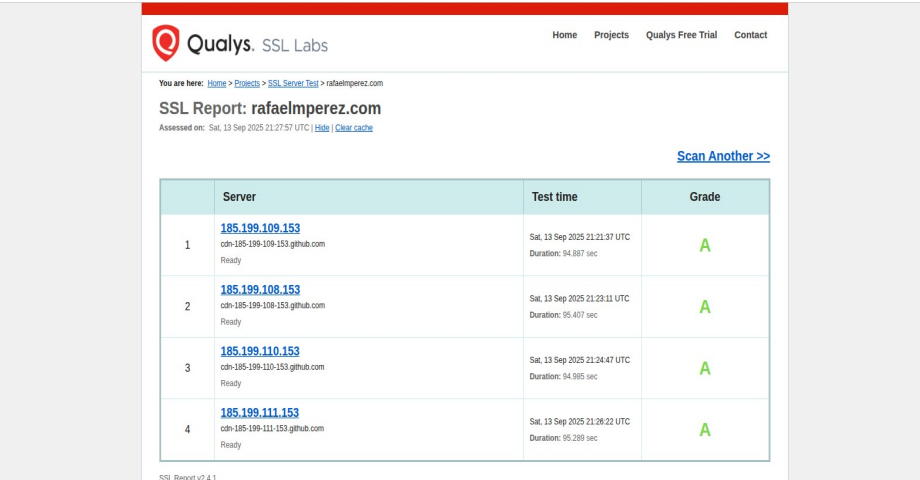
Presencia de cabeceras internas: como x-fastly-request-id, x-github-request-id, y x-served-by, que revelan detalles del entorno de alojamiento.

Servidor detrás de proxy o WAF: el banner indica que el sitio está protegido por un balanceador de carga o firewall de aplicaciones.

En conjunto, los resultados muestran una configuración criptográfica sólida, pero con oportunidades de mejora en la gestión de cabeceras HTTP y en la personalización del entorno de seguridad. Se recomienda revisar las políticas CORS, ocultar cabeceras internas y reforzar la configuración TLS con políticas como HSTS y OCSP stapling.



SSL Labs:



	Server	Test time	Grade
1	185.199.109.153 cdn-185-199-109-153.github.com Ready	Sat, 13 Sep 2025 21:21:37 UTC Duration: 94.887 sec	A
2	185.199.108.153 cdn-185-199-108-153.github.com Ready	Sat, 13 Sep 2025 21:23:11 UTC Duration: 95.407 sec	A
3	185.199.110.153 cdn-185-199-110-153.github.com Ready	Sat, 13 Sep 2025 21:24:47 UTC Duration: 94.985 sec	A
4	185.199.111.153 cdn-185-199-111-153.github.com Ready	Sat, 13 Sep 2025 21:26:22 UTC Duration: 95.289 sec	A

Por último, en esta primera fase de detección utilicé la herramienta web [SSL Labs](#) de Qualys para evaluar la configuración SSL/TLS del sitio. Esta herramienta realiza un análisis profundo del cifrado, certificados, compatibilidad con navegadores y resistencia frente a vulnerabilidades conocidas.

<https://www.ssllabs.com/ssltest/analyze.html?d=rafaelperez.com>

El escaneo arrojó una calificación **A** para los subdominios asociados al servidor, lo que indica una configuración SSL/TLS robusta. Entre los aspectos positivos se destacan:

Soporte para TLS 1.2 y TLS 1.3: protocolos modernos y seguros.

Uso de certificados válidos: emitidos por autoridades reconocidas.

Compatibilidad con navegadores actuales: sin advertencias ni errores de validación.

Ausencia de vulnerabilidades críticas: como Heartbleed, POODLE o FREAK.

Aunque la calificación es alta, se observó que el certificado utilizado es un wildcard para *.github.io, lo que no valida directamente el dominio rafaelperez.com. Además, no se detectó la implementación de políticas como HSTS o OCSP stapling, que podrían reforzar aún más la seguridad del canal HTTPS.

Se recomienda personalizar el certificado para el dominio principal y aplicar políticas avanzadas de seguridad TLS para maximizar la protección frente a ataques de intermediarios y mejorar la confianza del usuario.

Fase 2: Bastionado y medidas aplicadas

En esta fase se aplican medidas concretas para reforzar la seguridad del sistema. Se parte de los hallazgos obtenidos en la auditoría inicial y se implementan configuraciones que reducen la superficie de ataque, mejoran el cifrado y protegen el entorno frente a amenazas comunes.

Vinculación con Cloudflare:



Cloudflare Theworldoffragrance@gmail.com's Account

rafaelmperez.com ✓ Activo ☆ Estrella Plan Free

<input type="checkbox"/>	Tipo	Nombre	Contenido	Estado de proxy	TTL	Acciones
<input type="checkbox"/>	A	rafaelmperez.com	185.199.110.153	Redirigido mediante proxy Automático		Editar
<input type="checkbox"/>	A	rafaelmperez.com	185.199.109.153	Redirigido mediante proxy Automático		Editar
<input type="checkbox"/>	A	rafaelmperez.com	185.199.111.153	Redirigido mediante proxy Automático		Editar
<input type="checkbox"/>	A	rafaelmperez.com	185.199.108.153	Redirigido mediante proxy Automático		Editar
<input type="checkbox"/>	CNAME	www	rafaelmperez.github.io	Redirigido mediante proxy Automático		Editar

Servidores de nombres de Cloudflare
A cada zona DNS en Cloudflare se le asigna un conjunto de servidores de nombres de la marca Cloudflare.

Tipo	Valor
NS	amos.ns.cloudflare.com
NS	bailey.ns.cloudflare.com

Para comenzar se configuró el dominio `rafaelmperez.com` en Cloudflare para gestionar el tráfico web, aplicar políticas de seguridad y optimizar el rendimiento. En el panel de DNS se añadieron los registros necesarios para apuntar a GitHub Pages:

Registro tipo **A** apuntando a 185.199.108.153 (IP de GitHub Pages).

Registro tipo **CNAME** para subdominios si aplica.

Activación del proxy de Cloudflare (nube naranja) para proteger y filtrar el tráfico.

Nota:

El dominio ya se encontraba previamente vinculado y activo en GitHub Pages. Esto indica que los registros DNS estaban correctamente configurados y que GitHub había verificado la titularidad del dominio en una etapa anterior.

Gracias a esta vinculación previa, no fue necesario añadir un nuevo registro TXT para verificación. El dominio se resolvía correctamente y contaba con soporte HTTPS desde el inicio de la auditoría.



Configuración SSL/TLS:

The screenshot shows the Cloudflare account page for 'rafaelperez.com'. The left sidebar lists various settings like 'Información general', 'Auditoría de IA', 'Log Explorer', 'DNS', 'Correo electrónico', 'SSL/TLS', 'Seguridad', and 'Access'. The 'SSL/TLS' section is selected, showing the 'Cifrado SSL/TLS' configuration. The current encryption mode is 'Completo (estricto)'. A diagram shows the flow from 'Navegador' to 'Cloudflare' to 'Servidor de origen'. A 'Configurar' button is present. Below the diagram, it states 'Tráfico entregado mediante TLS Últimas 24 horas'.

Se activó el modo **Full (Strict)** en la sección SSL/TLS de Cloudflare, lo que garantiza que la conexión entre el navegador y Cloudflare, y entre Cloudflare y GitHub Pages, esté cifrada con certificados válidos. Esta configuración evita el uso de certificados autofirmados y mejora la seguridad del canal HTTPS.

Aplicación de cabeceras HTTP seguras:

The screenshot shows the Cloudflare 'Reglas' (Rules) page. The left sidebar lists various settings like 'Seguridad', 'Access', 'Speed', 'Almacenamiento en caché', 'Rutas de Workers', 'Reglas', 'Páginas de error', 'Red', 'Tráfico', 'Scrape Shield', and 'Web3'. The 'Reglas' section is selected, showing a list of rules for adding security headers. The rules are: Permissions-Policy, Referrer-Policy, Strict-Transport-Security, X-Content-Type-Options, and X-Frame-Options. Each rule has a 'Quitar' button. A 'Establecer nuevo encabezado' button is at the bottom.

Se utilizaron **Transform Rules** en Cloudflare para añadir cabeceras de seguridad directamente desde la red de distribución. Esto permite aplicar políticas como:

Strict-Transport-Security

X-Frame-Options

X-Content-Type-Options

Referrer-Policy

Permissions-Policy

Content-Security-Policy con fuentes específicas



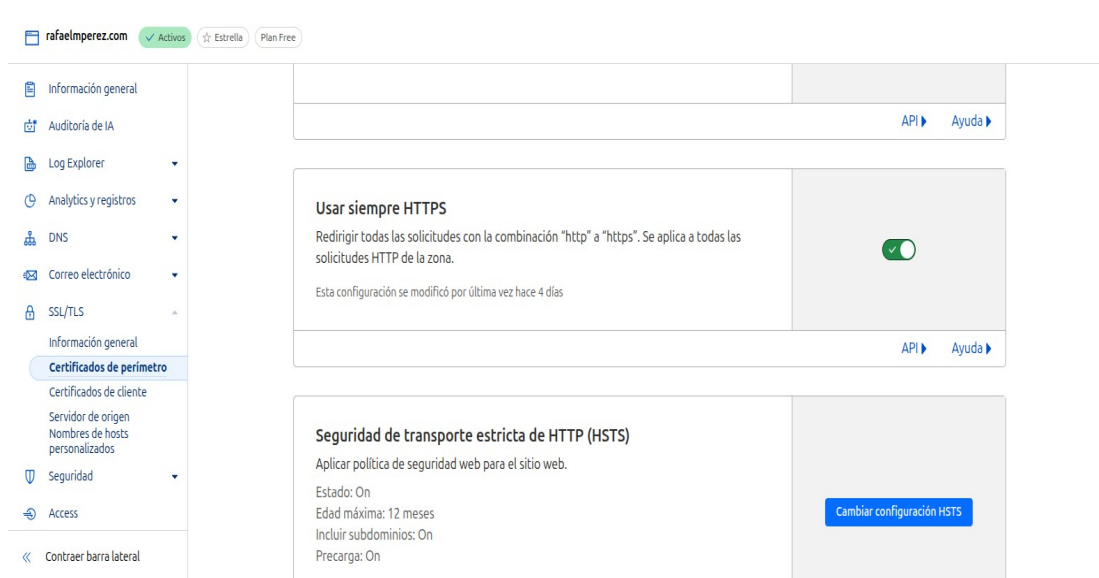
Estas cabeceras se aplican sin necesidad de modificar el código fuente del sitio, lo que es ideal para proyectos estáticos como los alojados en GitHub Pages.

Limpieza de cabeceras internas:

Se revisaron las cabeceras expuestas por el proxy de Cloudflare mediante herramientas como curl y SecurityHeaders. No se detectaron cabeceras internas como x-github-request-id, x-fastly-request-id o x-served-by, por lo que no fue necesario aplicar reglas de eliminación.

En caso de que estas cabeceras aparezcan en el futuro, se recomienda aplicar reglas de transformación en Cloudflare para eliminarlas desde el perímetro.

Edge Certificates y configuración de HSTS:



Se activó el certificado SSL universal gestionado por Cloudflare para el dominio. Este certificado es emitido por la autoridad **SSL.com** y validado mediante registros TXT internos (_acme-challenge), sin necesidad de intervención manual.

Además, se activó la política **HTTP Strict Transport Security (HSTS)** en Cloudflare, con los siguientes parámetros:

max-age: 12 meses

includeSubDomains: activado

preload: activado

También se habilitaron las siguientes opciones complementarias:

Usar siempre HTTPS: activado

Reescrituras automáticas HTTPS: activado

TLS mínimo: 1.2

TLS 1.3: activado



Esta configuración garantiza que todas las conexiones al sitio se realicen de forma segura, evitando el uso de HTTP y protegiendo contra ataques de tipo downgrade o MITM.

Bloqueo de tráfico fuera de Europa:

Se creó una regla personalizada de seguridad en Cloudflare para restringir el acceso al sitio web a visitantes ubicados en países europeos específicos y el resto del mundo. Esta medida reduce el riesgo de tráfico malicioso proveniente de regiones no relevantes para el proyecto.

Nombre de la regla: Bloqueo fuera de Europa

Campo: http.request.geo.country

Operador: no está en

Valor: {"ES", "FR", "DE", "IT", "PT"}

Acción: Managed Challenge (desafío automático para visitantes fuera de Europa)

Esta regla se aplica desde el perímetro de Cloudflare, antes de que el tráfico alcance el servidor de origen (GitHub Pages). En lugar de bloquear directamente, se utiliza **Managed Challenge** para permitir el acceso solo a usuarios legítimos que superen la verificación.

Orden	Nombre	Contra coincidencia	Acción	CSR	Eventos de últimas 24h	Estado
1	Bloqueo fuera de Europa	País no está en FR, DE, PT, ES, GB, IT	Bloquear	-	0	Activo

Configuraciones adicionales de seguridad:

Se aplicaron múltiples configuraciones avanzadas de seguridad desde el panel de Cloudflare para reforzar la protección del dominio rafaelimperez.com. Estas medidas se implementaron desde el plan Free, aprovechando las funcionalidades disponibles para mitigar amenazas comunes como bots, scraping, ataques DDoS y rastreadores de IA.

DNSSEC (Domain Name System Security Extensions) Sirve para proteger el dominio contra manipulaciones maliciosas en el sistema DNS.

Modo "I'm Under Attack": activado. Presenta un desafío JavaScript a todos los visitantes antes de cargar el sitio, ideal para mitigar ataques DDoS.



Bloqueo de bots de IA: activado. Se implementó una regla administrada para bloquear rastreadores utilizados en el entrenamiento de modelos de inteligencia artificial.

Modo Bot Fight: activado. Detecta y desafía bots automatizados que realizan scraping, fraude de clics o fuerza bruta.

Protección contra ataques DDoS: activa en todas las capas (red, HTTP, SSL/TLS). Cloudflare mitiga automáticamente ataques como SYN flood, UDP flood y ataques de agotamiento SSL.

Pasaje de desafío: configurado a 30 minutos. Los visitantes que superan un desafío no reciben otro durante ese periodo, mejorando la experiencia sin comprometer la seguridad.

Detección de credenciales filtradas: activa. Limita intentos de autenticación con contraseñas comprometidas.

AI Labyrinth: activado. Inserta enlaces nofollow generados por IA para confundir rastreadores que ignoran las normas de exclusión.

Bloqueo por agente de usuario: disponible. Se puede aplicar para bloquear navegadores o herramientas específicas que presenten encabezados sospechosos.

Protección contra exploits de aplicaciones web: activa mediante reglas administradas por Cloudflare.

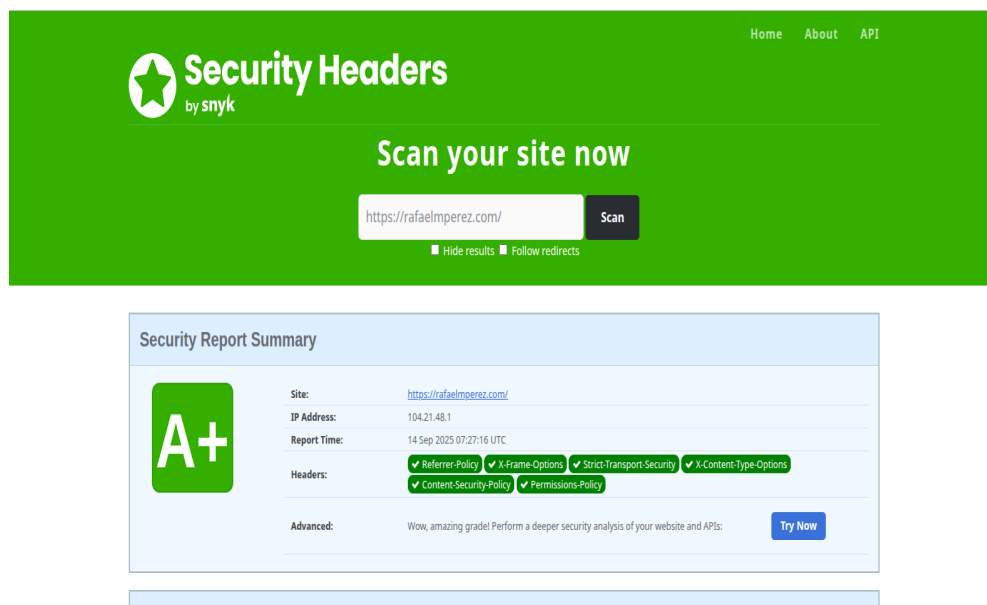
Reemplazo de bibliotecas JavaScript inseguras: habilitado. Cloudflare sustituye automáticamente bibliotecas vulnerables por versiones seguras desde cdnjs.

Estas configuraciones se gestionan desde el nuevo panel de seguridad de aplicaciones de Cloudflare, que permite filtrar por tipo de amenaza, configurar reglas personalizadas y aplicar plantillas de protección automatizada. La postura de seguridad del dominio se considera endurecida al máximo dentro del alcance del plan Free.

Nota: Aunque el bastionado de servidores suele incluir la configuración de firewalls como UFW, esta medida no aplica en el presente caso, ya que el sitio rafaelperez.com está alojado en **GitHub Pages**. La infraestructura subyacente es gestionada por GitHub, y no se tiene acceso directo al entorno de red ni al sistema operativo. Por tanto, las medidas de seguridad se enfocan exclusivamente en la capa web: cabeceras HTTP, cifrado TLS, certificados y políticas de navegador.



Resultado: Auditoría de cabeceras HTTP (Security Headers)



Tras aplicar las medidas de bastionado en Cloudflare, el sitio web pasó de una calificación **F** a **A+** en **SecurityHeaders.com**, validando la correcta implementación de cabeceras críticas como Strict-Transport-Security, Content-Security-Policy y Permissions-Policy.

Cabeceras duplicadas corregidas mediante reglas de transformación

Redirección forzada a HTTPS y TLS 1.2/1.3 activos

Política CSP personalizada aplicada con fuentes seguras

Cabeceras internas eliminadas desde el perímetro

Este resultado demuestra que el entorno web ha sido reforzado con éxito, alcanzando un nivel de seguridad óptimo sin afectar la disponibilidad ni la experiencia del usuario.

Escaneo con Nikto:

Volvimos a ejecutar un escaneo con **Nikto** sobre el dominio usando el comando `nikto -h https://rafaelperez.com -p 443` para detectar vulnerabilidades comunes en servidores web. El resultado fue positivo desde el punto de vista defensivo, ya que el sitio respondió con código 403, lo que indica que el firewall bloqueó el escaneo.

Servidor detectado: Cloudflare

CGI directories: no se encontraron rutas vulnerables

Código de respuesta: 403 Forbidden (escaneo bloqueado)

Aunque Nikto no pudo completar el escaneo por bloqueo del firewall, esto es deseable en entornos productivos. Se recomienda revisar la cabecera X-Frame-Options para confirmar que esté activa y no duplicada, como se detectó en escaneos anteriores.



Escaneo con SSL Labs:

Seguidamente un escaneo completo del dominio utilizando la herramienta **SSL Labs**, con el objetivo de validar la configuración TLS, el certificado SSL y la seguridad del canal de transporte. El resultado fue una calificación **A+**, lo que indica cumplimiento total con las mejores prácticas de cifrado y seguridad.

Protocolos activos: TLS 1.2 y TLS 1.3

Certificado: válido, emitido por Cloudflare, con cadena completa

HSTS: activo con preload habilitado

Forward Secrecy: soportado en todos los navegadores modernos

Servidores detectados: se pasó de 4 a 14 servidores distribuidos globalmente, lo que mejora la disponibilidad y la tolerancia a fallos

Vulnerabilidades conocidas: no se detectaron (Heartbleed, ROBOT, etc.)

Este resultado confirma que el sitio no solo está bastionado en la capa de aplicación, sino también en la capa de transporte. La expansión de infraestructura con más servidores mejora la resiliencia y el rendimiento global.

Escaneo con testssl.sh:

Por último se volvió a ejecutar un escaneo local utilizando la herramienta **testssl.sh** para validar la seguridad del canal TLS. Los comandos utilizados fueron:

```
cd testssl.sh
```

```
./testssl.sh https://rafaelperez.com
```

Heartbleed, CCS, Ticketbleed, POODLE, FREAK, DROWN: no vulnerabilidades detectadas

ROBOT: no se soportan suites RSA inseguras (OK)

Secure Renegotiation: soportado correctamente

CRIME, BEAST, LOGJAM, Winshock: no vulnerabilidades detectadas

RC4 y SSLv3: no soportados

TLS_FALLBACK_SCSV: activo, evita degradación de protocolo

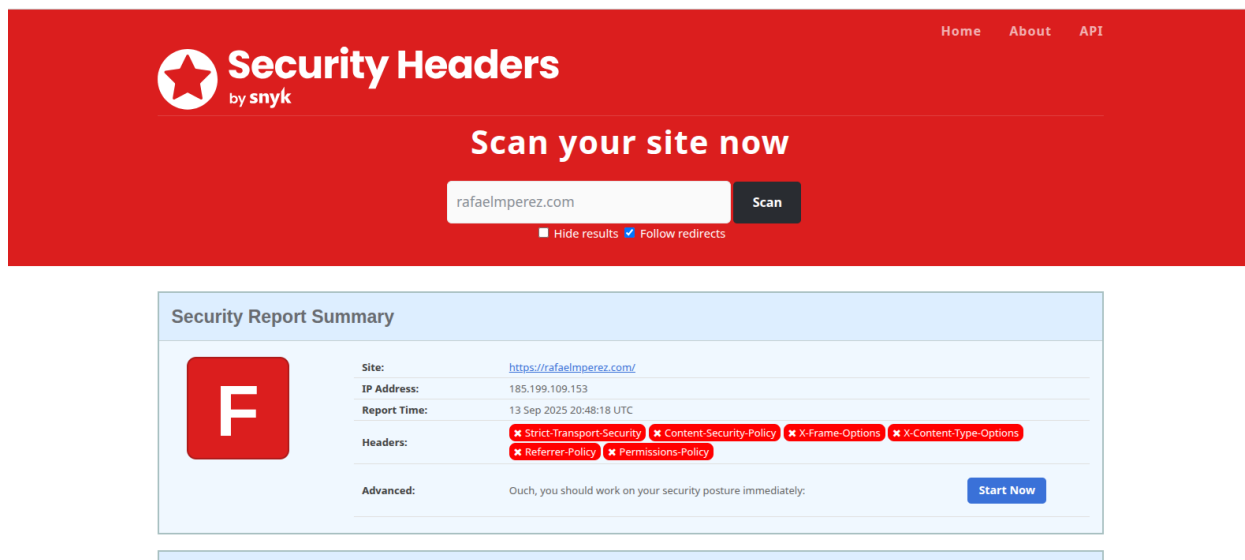
El escaneo confirma que el sitio rafaelperez.com opera bajo una configuración TLS moderna y segura, sin exposición a vulnerabilidades críticas conocidas. Se valida así la efectividad del bastionado aplicado en la capa de transporte.

Integral, escalable y sostenible de protección, especialmente en entornos con soporte extendido como Ubuntu 24.04 LTS.

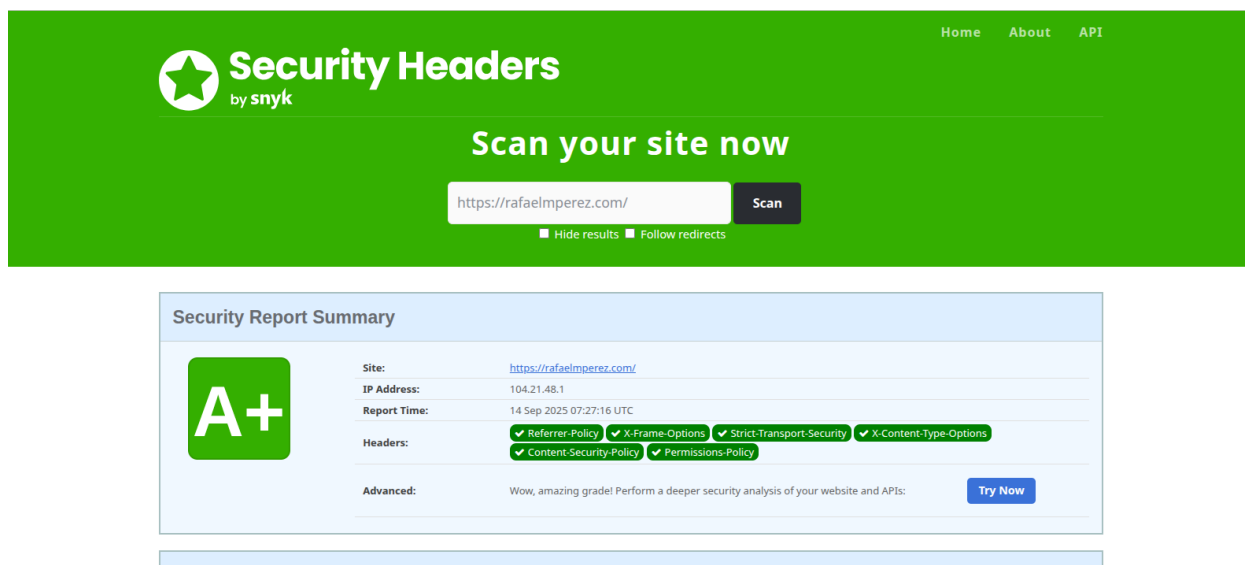


4. Conclusiones y Recomendaciones Finales:

El sitio web ha evolucionado de una configuración predeterminada y vulnerable a una plataforma bastionada con cabeceras clave, cifrado robusto y escaneos técnicos que validan su seguridad. La infraestructura ha crecido de 4 a 14 servidores distribuidos, lo que mejora la resiliencia y el rendimiento global.



Antes



Después

Aplicación de buenas prácticas: Se han implementado cabeceras HTTP seguras, redirección forzada a HTTPS, y compatibilidad con TLS 1.2/1.3.

Escaneos técnicos: Se han utilizado herramientas como Nikto, SSL Labs y testssl.sh para validar la seguridad desde distintas capas.

Firewall perimetral: Cloudflare actúa como primera línea de defensa, bloqueando escaneos y mitigando ataques comunes.



Compromiso personal: Aunque ningún sistema está exento de ser atacado, este sitio no representa un objetivo atractivo. Aun así, se ha reforzado con convicción y se recomienda aplicar estas prácticas, ya que muchas brechas ocurren por descuidos evitables.

Este ejercicio no solo mejora la postura de seguridad del sitio, sino que refleja una cultura de prevención y responsabilidad digital que debería ser estándar en cualquier proyecto web.

***Nota: Este proyecto no termina aquí. Se seguirán realizando auditorías periódicas con herramientas como OWASP ZAP para análisis de vulnerabilidades web y Nuclei para escaneos automatizados de seguridad. La mejora continua es parte del compromiso con la ciberseguridad.**



5. Resumen de herramientas utilizadas:

Herramienta	Función principal
Cloudflare	Plataforma de seguridad perimetral: WAF, mitigación DDoS, TLS, cabeceras HTTP seguras y control geográfico
Security Headers	Analiza cabeceras HTTP (CSP, HSTS, X-Frame-Options, etc.) y califica la postura de seguridad
Nikto	Escáner de vulnerabilidades web que detecta configuraciones inseguras, archivos expuestos y cabeceras erróneas
SSL Labs (Qualys)	Evalúa la configuración SSL/TLS del sitio, certificados, cifrado y compatibilidad con navegadores
testssl.sh	Analiza localmente protocolos, suites de cifrado y vulnerabilidades criptográficas conocidas
SSLyze	Realiza pruebas de configuración SSL/TLS, soporte de protocolos y análisis de cifrado avanzado
OWASP ZAP	Escáner automatizado de vulnerabilidades en aplicaciones web (XSS, inyección, configuración insegura, etc.)
Nuclei	Escáner automatizado con plantillas personalizadas para detectar vulnerabilidades conocidas
GitHub Pages	Plataforma de alojamiento estático del sitio web, base del entorno analizado
curl	Verificación manual de cabeceras y respuestas HTTP durante la validación



5. Anexo Técnico

1. Alcance del proyecto

El presente anexo detalla los procedimientos, configuraciones y validaciones técnicas aplicadas durante el proceso de auditoría y bastionamiento del sitio web **rafaelmperez.com**, alojado en **GitHub Pages** y protegido mediante **Cloudflare**.

El objetivo principal fue reforzar la capa de seguridad web y de transporte, aplicando medidas que aumentaran la resiliencia del entorno sin afectar la disponibilidad ni la experiencia del usuario.

2. Entorno y configuración base

- **Hosting:** GitHub Pages (infraestructura gestionada).
- **DNS y Seguridad perimetral:** Cloudflare (modo proxy “nube naranja”).
- **Certificado SSL:** Universal Certificate (SSL.com) gestionado por Cloudflare.
- **Protocolos activos:** TLS 1.2 / TLS 1.3.
- **Configuración inicial:** Sin cabeceras HTTP seguras, sin políticas CSP ni HSTS, modo SSL flexible.

3. Metodología aplicada

El proceso de bastionamiento se basó en las siguientes referencias normativas:

- **OWASP Server Hardening Guide** (mejores prácticas para servidores web).
- **CIS Benchmarks for Web Servers** (control de configuraciones seguras).
- **NIST SP 800-123** (guía técnica para la seguridad de servidores).

Las fases ejecutadas fueron:

1. **Auditoría inicial** con herramientas de análisis TLS y cabeceras HTTP.
2. **Implementación progresiva** de políticas de seguridad en Cloudflare.
3. **Revisión y validación** mediante escaneos técnicos posteriores.

4. Resultados obtenidos

Métrica evaluada	Estado inicial	Estado final	Mejora
Calificación en Security Headers	F	A+	Alta
Calificación en SSL Labs (Qualys)	A	A+	Media
Configuración TLS	Parcial (Flexible)	Completa (Full Strict)	Alta
Cabeceras HTTP críticas	Ausentes / Incorrectas	Aplicadas correctamente	Alta
Protección DDoS / Bot Fight	Inactiva	Activa en Cloudflare	Alta

5. Evidencias técnicas

- Escaneos iniciales y finales con **Nikto**, **Security Headers**, **SSL Labs**, **testssl.sh** y **SSLyze**.
- Capturas de resultados y reportes en PDF adjuntos en el repositorio del proyecto.
- Registro de configuraciones aplicadas en Cloudflare (HSTS, TLS, CSP, geo-blocking).



6. Conclusión del anexo

El proyecto de bastionamiento web logró **alinearse la configuración del dominio rafaelperez.com con estándares internacionales de seguridad** (OWASP, CIS, NIST), garantizando comunicaciones seguras, políticas HTTP robustas y protección activa frente a amenazas automatizadas.

El entorno actual es **estable, seguro y fácilmente mantenible**, y servirá como base para futuras prácticas avanzadas de auditoría web con **OWASP ZAP** y **Nuclei**, orientadas al mantenimiento continuo de la postura de seguridad.