



**Agencia
Nacional Digital**



REPORTE DE HARDENING

Superintendencia de Transporte

Entregables de la fase orientados al fortalecimiento
institucional de la ciberseguridad

ABRIL 2026

Documento técnico para entrega institucional

CONTROL DE CAMBIOS DEL DOCUMENTO



Agencia Nacional Digital



REVISIÓN No.	FECHA	DESCRIPCIÓN	AUTOR DEL CAMBIO
1	09/04/2026	Creación del documento	Virgilio Salgado



Agencia Nacional Digital



Contenido

1. Introducción	4
2. Objetivo General	4
3. Objetivos Específicos	5
4. Marco Teórico.....	5
4.1 Seguridad en Aplicaciones Web.....	5
4.3 OWASP Top 10	6
4.4 Principales Vectores de Ataque	7
4.4.1 SQL Injection (SQLi)	7
4.4.2 Cross-Site Scripting (XSS)	8
4.4.3 Path Traversal (Directory Traversal)	8
4.4.4 Gestión Insegura de Dependencias	8
4.5 Herramientas de Referencia Ofensiva	9
4.6 Defensa en Profundidad (<i>Defense in Depth</i>)	9
5. Marco Legal.....	9
5.1 Normativa Nacional (Colombia)	9
5.2 Estándares Internacionales.....	10
6. Justificación	12
7. Recomendaciones Técnicas	13
7.1 Prevención de SQL Injection	13
Ejemplo vulnerable (Node.js):	13
Ejemplo seguro:.....	13
Ejemplo en PHP (PDO):	14
Referencia ofensiva:	14
7.2 Prevención de Cross-Site Scripting (XSS)	14
7.3 Prevención de Path Traversal (Directory Traversal).....	15
8. Configuración Segura de Servidores	16
9. Conclusión	18



Agencia Nacional Digital



1. Introducción

En el contexto actual de la transformación digital, las organizaciones enfrentan un ecosistema de amenazas cibernéticas en constante evolución, donde los actores maliciosos disponen de herramientas cada vez más sofisticadas para comprometer la confidencialidad, integridad y disponibilidad de los sistemas de información. Las aplicaciones web, por su naturaleza expuesta y su rol crítico en la operación institucional, constituyen uno de los vectores de ataque más frecuentemente explotados a nivel global.

El proceso de *hardening* o endurecimiento de plataformas tecnológicas consiste en la reducción deliberada de la superficie de ataque mediante la aplicación de controles técnicos, configuraciones seguras y buenas prácticas de desarrollo y operación. Este proceso no se limita al ámbito del código fuente, sino que abarca capas transversales de la arquitectura: servidores web, bases de datos, sistemas operativos, mecanismos de autenticación y comunicación cifrada.

Elaboramos el presente documento en el marco de un proceso de evaluación de seguridad sobre aplicaciones web institucionales. Dado que no se contó con acceso al código fuente de las aplicaciones analizadas, el enfoque adoptado se sustenta en la identificación de vectores de ataque conocidos, el análisis de comportamiento externo de las aplicaciones y la formulación de lineamientos técnicos basados en estándares internacionales como OWASP, NIST, ISO/IEC 27001 y CIS Controls.

Hacemos especial énfasis en vulnerabilidades de alta criticidad como SQL Injection, Cross-Site Scripting (XSS) y Path Traversal, así como en la gestión de dependencias inseguras, la configuración de servidores web (Apache/Nginx) y la implementación de mecanismos de monitoreo y respuesta ante incidentes. La finalidad es proporcionar una guía técnica aplicable, replicable y alineada con la estrategia de ciberseguridad institucional.

2. Objetivo General

Establecer un conjunto de lineamientos técnicos y recomendaciones de hardening orientados a reducir la superficie de ataque, mitigar vulnerabilidades críticas identificadas en entornos web y fortalecer la postura de seguridad de la infraestructura tecnológica institucional, con base en estándares internacionales y buenas prácticas de la industria de ciberseguridad.



3. Objetivos Específicos

1. **Identificar y caracterizar** las principales vulnerabilidades presentes en aplicaciones web institucionales, con énfasis en las categorías definidas por el OWASP Top 10, incluyendo SQL Injection, XSS y Path Traversal.
2. **Formular recomendaciones técnicas concretas** para la prevención y mitigación de vulnerabilidades críticas, aplicables a tecnologías como PHP, Node.js y JavaScript, sin requerir acceso directo al código fuente.
3. **Definir estándares de configuración segura** para servidores web Apache y Nginx, incluyendo headers de seguridad HTTP, restricción de métodos, implementación de WAF y políticas de control de acceso.
4. **Proponer un modelo de gestión de dependencias seguras** (SCA — Software Composition Analysis) que permita identificar, monitorear y remediar librerías con vulnerabilidades conocidas (CVE).
5. **Establecer medidas transversales de seguridad** que incluyan la implementación de HTTPS/TLS, políticas HSTS, gestión segura de logs, y lineamientos para la integración con capacidades de monitoreo continuo (SIEM/SOC).
6. **Generar conciencia técnica** sobre las técnicas ofensivas documentadas en repositorios de referencia como *PayloadsAllTheThings*, con el fin de fortalecer los controles defensivos desde la comprensión del adversario.

4. Marco Teórico

4.1 Seguridad en Aplicaciones Web

La seguridad en aplicaciones web es una disciplina de la ciberseguridad que se ocupa de la protección de sistemas accesibles a través de protocolos HTTP/HTTPS frente a amenazas externas e internas. A diferencia de los sistemas de información tradicionales, las aplicaciones web presentan una



Agencia Nacional Digital



superficie de ataque ampliada por su naturaleza pública, su interacción con múltiples usuarios simultáneos y su dependencia de componentes de terceros. El modelo de amenazas en entornos web puede clasificarse según la tríada CIA (*Confidentiality, Integrity, Availability*):

- **Confidencialidad:** Exposición no autorizada de datos sensibles (credenciales, PII, información financiera).
- **Integridad:** Modificación no autorizada de datos o funcionalidades del sistema.
- **Disponibilidad:** Interrupciones del servicio originadas por ataques de denegación de servicio (DoS/DDoS) o explotación de recursos.

4.2 Hardening de Sistemas

El término *hardening* hace referencia al proceso sistemático de reducción de vulnerabilidades en un sistema informático mediante la eliminación de funciones innecesarias, la aplicación de parches de seguridad, la restricción de privilegios y la implementación de controles preventivos y detectivos. Este proceso se aplica a múltiples niveles:

- **Hardening de sistema operativo:** Deshabilitación de servicios innecesarios, gestión de usuarios y permisos, configuración de firewall local.
- **Hardening de servidor web:** Configuración segura de Apache/Nginx, restricción de cabeceras HTTP expuestas, implementación de WAF.
- **Hardening de aplicación:** Validación de entradas, codificación de salidas, gestión segura de sesiones, control de errores.
- **Hardening de base de datos:** Principio de mínimo privilegio, cifrado en reposo y en tránsito, auditoría de consultas.

4.3 OWASP Top 10

La fundación *Open Web Application Security Project* (OWASP) es la referencia global más reconocida en materia de seguridad de aplicaciones web. Su documento *OWASP Top 10* cataloga las diez categorías de riesgo más críticas,



Agencia Nacional Digital



basadas en datos reales de incidentes y vulnerabilidades reportadas a nivel mundial. Las categorías más relevantes para el presente documento son:

- **A01 - Broken Access Control:** Control de acceso inadecuado que permite a usuarios acceder a recursos o funciones no autorizadas.
- **A03 - Injection:** Incluye SQL Injection, Command Injection, LDAP Injection y otras formas de inyección de código malicioso a través de entradas no validadas.
- **A07 - Identification and Authentication Failures:** Debilidades en la gestión de identidad, sesiones y credenciales.
- **A06 - Vulnerable and Outdated Components:** Uso de librerías, frameworks o componentes con vulnerabilidades conocidas (CVE).
- **A05 - Security Misconfiguration:** Configuraciones por defecto inseguras, exposición de información sensible en errores o cabeceras HTTP.

4.4 Principales Vectores de Ataque

4.4.1 SQL Injection (SQLi)

La inyección SQL es una vulnerabilidad que ocurre cuando una aplicación incorpora datos no confiables en una consulta SQL sin la debida validación o parametrización. Un atacante puede manipular la lógica de la consulta para extraer, modificar o eliminar datos, eludir mecanismos de autenticación o ejecutar comandos en el sistema operativo subyacente (en configuraciones con privilegios elevados).

Se clasifica en tres variantes principales:

- **In-band SQLi:** El resultado se obtiene directamente en la respuesta de la aplicación (Error-based, Union-based).
- **Blind SQLi:** No hay respuesta directa, pero se infiere información a través de comportamientos booleanos o temporales (Boolean-based, Time-based).



Agencia Nacional Digital



- **Out-of-band SQLi:** El atacante recibe la información a través de un canal alternativo (DNS, HTTP externo).

4.4.2 Cross-Site Scripting (XSS)

El XSS es una vulnerabilidad que permite a un atacante inyectar código JavaScript malicioso en páginas web visualizadas por otros usuarios. El código se ejecuta en el contexto del navegador de la víctima, con acceso a cookies, tokens de sesión y el DOM del sitio afectado.

Existen tres tipos:

- **Reflected XSS:** El payload es parte de la solicitud HTTP y se refleja directamente en la respuesta.
- **Stored XSS (Persistent):** El payload se almacena en la base de datos y se sirve a todos los usuarios que visitan la página afectada.
- **DOM-based XSS:** La vulnerabilidad reside en el código JavaScript del lado del cliente que procesa datos del DOM sin sanitización.

4.4.3 Path Traversal (Directory Traversal)

Esta vulnerabilidad permite a un atacante acceder a archivos y directorios fuera del directorio raíz de la aplicación web, utilizando secuencias como `../` o codificaciones equivalentes (`%2e%2e%2f`). Su explotación puede exponer archivos de configuración del sistema, claves privadas, credenciales o cualquier archivo accesible por el proceso del servidor web.

4.4.4 Gestión Insegura de Dependencias

El uso de librerías y componentes de terceros con versiones desactualizadas o vulnerabilidades conocidas representa un riesgo crítico ampliamente subestimado. Herramientas como *npm audit*, *Snyk*, *OWASP Dependency-Check* y *Dependabot* permiten automatizar la identificación de componentes vulnerables mediante el cruce con bases de datos de CVE (Common Vulnerabilities and Exposures).



Agencia
Nacional Digital



4.5 Herramientas de Referencia Ofensiva

El repositorio **PayloadsAllTheThings** (GitHub) es una colección colaborativa de payloads y técnicas de bypass utilizados en pruebas de penetración. Cubre vectores como SQL Injection, XSS, SSRF, XXE, Command Injection, Open Redirect y evasión de WAF, entre otros. Su estudio desde una perspectiva defensiva permite comprender la lógica del atacante y fortalecer los controles de validación y sanitización.

4.6 Defensa en Profundidad (*Defense in Depth*)

El modelo de *defensa en profundidad* es una estrategia de seguridad que plantea la implementación de múltiples capas de control, de manera que el fallo de un control no implique el compromiso total del sistema. Aplicado al contexto web, implica controles en el nivel de red (firewall, IDS/IPS), de aplicación (WAF, validación de entradas), de datos (cifrado, control de acceso) y de monitoreo (SIEM, alertas en tiempo real).

5. Marco Legal

La implementación de medidas de seguridad informática en entornos institucionales no es únicamente una decisión técnica, sino también una obligación legal y regulatoria. A continuación se presentan los principales referentes normativos aplicables al contexto colombiano e internacional:

5.1 Normativa Nacional (Colombia)

Ley 1273 de 2009 — De la protección de la información y de los datos

Esta ley modificó el Código Penal colombiano y creó un bien jurídico autónomo denominado "la protección de la información y de los datos". Tipifica como delitos conductas como el acceso abusivo a sistemas informáticos (Art. 269A), la interceptación de datos informáticos (Art. 269C), el daño informático (Art. 269D), la violación de datos personales (Art. 269F) y el uso de software malicioso (Art. 269E), entre otros. Las organizaciones que no adopten medidas de seguridad adecuadas pueden verse expuestas a responsabilidades civiles y penales derivadas de incidentes que pudieron prevenirse.

Ley 1581 de 2012 — Ley de Protección de Datos Personales (Habeas Data)



Agencia Nacional Digital



Regula el tratamiento de datos personales en Colombia y establece principios como finalidad, libertad, veracidad, transparencia, acceso restringido, seguridad y confidencialidad. Su artículo 17 impone a los responsables del tratamiento la obligación de adoptar medidas técnicas, humanas y administrativas para garantizar la seguridad de los datos personales frente a riesgos de adulteración, pérdida, consulta, uso o acceso no autorizado.

Decreto 1377 de 2013

Reglamenta parcialmente la Ley 1581 de 2012 y precisa las obligaciones de los responsables y encargados del tratamiento de datos personales, incluyendo la implementación de políticas de privacidad y seguridad de la información.

Resolución 1519 de 2020 — MinTIC

Adopta los estándares de publicación y divulgación de información a través de los sitios web y canales digitales de las entidades del Estado, incluyendo requisitos de seguridad en las plataformas digitales gubernamentales.

CONPES 3995 de 2020 — *Política Nacional de Confianza y Seguridad Digital*

Define los lineamientos estratégicos de Colombia en materia de seguridad digital, incluyendo la gestión de riesgos digitales, la protección de infraestructuras críticas y el fortalecimiento de capacidades nacionales de ciberseguridad. Establece responsabilidades para entidades públicas en la adopción de controles de seguridad y la gestión de incidentes.

Decreto 338 de 2022 — MinDefensa / CCP

Crea la Comandancia Conjunta Cibernética y define el marco institucional para la defensa cibernética nacional, incluyendo la protección de activos digitales del Estado.

5.2 Estándares Internacionales

ISO/IEC 27001:2022 — *Sistema de Gestión de Seguridad de la Información (SGSI)*

Es el estándar internacional más reconocido para la gestión de la seguridad de la información. Define los requisitos para establecer, implementar, mantener y mejorar continuamente un SGSI. Su Anexo A incluye controles específicos relevantes para el presente documento, como:

- **A.8.28** — Codificación segura.
- **A.8.9** — Gestión de la configuración.
- **A.8.25** — Ciclo de vida de desarrollo seguro.



Agencia Nacional Digital



- **A.8.29** — Pruebas de seguridad en el desarrollo y aceptación.

ISO/IEC 27002:2022

Proporciona directrices detalladas para la implementación de los controles definidos en ISO/IEC 27001. Es una referencia técnica complementaria para la definición de políticas de hardening, gestión de vulnerabilidades y configuración segura de sistemas.

NIST Cybersecurity Framework (CSF) 2.0

Desarrollado por el *National Institute of Standards and Technology* de los Estados Unidos, el CSF organiza las prácticas de ciberseguridad en seis funciones: **Identificar, Proteger, Detectar, Responder, Recuperar y Gobernar**. Su aplicación es transversal a las medidas propuestas en este documento, especialmente en las funciones de Proteger (controles técnicos) y Detectar (monitoreo continuo, SIEM).

NIST SP 800-53 Rev. 5 — Security and Privacy Controls for Information Systems

Define un catálogo exhaustivo de controles de seguridad y privacidad para sistemas de información federales y organizaciones en general. Controles relevantes incluyen SI-10 (Validación de entradas), SI-15 (Sanitización de salidas de información), SC-28 (Protección en reposo) y AU-2 (Registro de eventos).

CIS Controls v8 — Center for Internet Security

Define 18 controles de seguridad críticos ordenados por prioridad de implementación. Los más aplicables al contexto de este documento son:

- **CIS Control 2:** Inventario y control de activos de software.
- **CIS Control 7:** Gestión continua de vulnerabilidades.
- **CIS Control 13:** Monitoreo y defensa de la red.
- **CIS Control 16:** Seguridad del software de aplicación.

PCI DSS v4.0 — Payment Card Industry Data Security Standard

Aunque está orientado al sector de pagos, sus controles representan una referencia robusta para cualquier entorno que maneje datos sensibles. Requerimientos relevantes incluyen el desarrollo seguro de software (Req. 6), la protección de datos en tránsito mediante cifrado (Req. 4) y la implementación de WAF (Req. 6.4.2).



Agencia Nacional Digital



GDPR — *General Data Protection Regulation* (Reglamento UE 2016/679)

Aunque es una regulación europea, aplica a cualquier organización que procese datos de ciudadanos de la Unión Europea. Su artículo 32 exige la adopción de medidas técnicas y organizativas apropiadas para garantizar un nivel de seguridad adecuado al riesgo, incluyendo el cifrado, la seudonimización y la evaluación continua de los controles implementados.

6. Justificación

Durante la ejecución del proceso de evaluación de seguridad, no fue suministrado acceso al código fuente de las aplicaciones analizadas, lo cual limita la posibilidad de realizar un proceso de hardening directo a nivel de desarrollo (secure coding). No obstante, esto no impide la identificación de riesgos ni la formulación de lineamientos técnicos robustos orientados a mitigar vulnerabilidades críticas presentes en entornos web.

En este sentido, adoptamos un enfoque basado en buenas prácticas, estándares internacionales y análisis de vectores de ataque conocidos, con el objetivo de proporcionar una guía técnica aplicable a diferentes tecnologías (PHP, JavaScript, Node.js, entre otras).

Como referencia técnica, se destaca el uso del repositorio PayloadsAllTheThings, ampliamente utilizado en pruebas de penetración. Este repositorio documenta técnicas como SQL Injection, XSS, Directory Traversal, Command Injection y bypass de WAF, permitiendo comprender cómo los atacantes construyen payloads y fortaleciendo los controles defensivos.

Enfatizamos que el uso de librerías desactualizadas representa un riesgo crítico. Recomendamos implementar gestión de dependencias seguras (SCA), actualización continua y monitoreo de vulnerabilidades (CVE).



Agencia
Nacional Digital

7. Recomendaciones Técnicas

7.1 Prevención de SQL Injection

La inyección SQL es una de las vulnerabilidades más críticas, ya que permite a un atacante manipular consultas a bases de datos, comprometiendo la confidencialidad, integridad y disponibilidad de la información.

Buenas prácticas:

- Uso obligatorio de consultas parametrizadas (Prepared Statements)
- Implementación de ORM seguros (TypeORM, Sequelize, Hibernate)
- Validación y sanitización estricta de entradas
- Principio de mínimo privilegio en usuarios de base de datos

Ejemplo vulnerable (Node.js):

</> JavaScript

```
const query = "SELECT * FROM users WHERE id = " + req.params.id;
```

Ejemplo seguro:

</> JavaScript

```
const query = "SELECT * FROM users WHERE id = ?";  
db.execute(query, [req.params.id]);
```



Agencia
Nacional Digital

Ejemplo en PHP (PDO):

```
</> PHP

$stmt = $pdo->prepare("SELECT * FROM users WHERE id = :id");
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
$stmt->execute();
```

Referencia

ofensiva:

Los payloads documentados en PayloadsAllTheThings muestran cómo un atacante puede explotar entradas como:

```
' OR 1=1 --
```

7.2 Prevención de Cross-Site Scripting (XSS)

El XSS permite a un atacante inyectar scripts maliciosos en el navegador de otros usuarios, comprometiendo sesiones, cookies y datos sensibles.

Buenas prácticas:

- Escapado de salida (output encoding)
- Uso de frameworks seguros (React, Angular con protección por defecto)
- Implementación de Content Security Policy (CSP)
- Sanitización de inputs en backend y frontend

Ejemplo vulnerable:

```
</> HTML

<div>Hola <?php echo $_GET['name']; ?></div>
```



Agencia
Nacional Digital

Ejemplo seguro:

```
</> PHP
<div>Hola <?php echo htmlspecialchars($_GET['name'], ENT_QUOTES, 'UTF-8'); ?></div>
```

CSP recomendado:

```
Content-Security-Policy: default-src 'self'; script-src 'self'
```

7.3 Prevención de Path Traversal (Directory Traversal)

Esta vulnerabilidad permite acceder a archivos fuera del directorio permitido, como archivos del sistema o configuraciones sensibles.

Ejemplo de ataque:

```
../../../../etc/passwd
```

Ejemplo vulnerable:

```
</> JavaScript
app.get('/file', (req, res) => {
  res.sendFile('/var/www/files/' + req.query.name);
});
```

Ejemplo seguro:



Agencia Nacional Digital

JavaScript

```
const path = require('path');  
const basePath = '/var/www/files/';  
const safePath = path.normalize(req.query.name).replace(/^(\\.|\\|\\$)+/, '');  
res.sendFile(path.join(basePath, safePath));
```

Buenas prácticas:

- Validación estricta de rutas
- Uso de listas blancas (whitelisting)
- Restricción de acceso a directorios críticos
- Contenedorización o sandboxing

8. Configuración Segura de Servidores

Apache

Configuraciones recomendadas:

apache

```
Options -Indexes  
ServerTokens Prod  
ServerSignature Off
```

Headers de seguridad:

apache

```
Header set X-Content-Type-Options "nosniff"  
Header set X-Frame-Options "DENY"  
Header set X-XSS-Protection "1; mode=block"
```



Agencia Nacional Digital

Recomendaciones adicionales:

- Deshabilitar módulos innecesarios
- Implementar mod_security (WAF)
- Limitar métodos HTTP (GET, POST)

Nginx

Configuraciones

recomendadas:

```
</> Nginx  
  
autoindex off;  
server_tokens off;
```

Headers de seguridad:

```
</> Nginx  
  
add_header X-Frame-Options DENY;  
add_header X-Content-Type-Options nosniff;  
add_header X-XSS-Protection "1; mode=block";
```

Buenas prácticas:

- Implementar rate limiting
- Configurar proxy seguro
- Uso de WAF (ModSecurity + Nginx)
- Validación de tamaños de payload (client_max_body_size)

Medidas Transversales

- Implementación de HTTPS obligatorio (TLS 1.2+)
- Uso de HSTS
- Monitoreo continuo (SIEM / SOC)
- Escaneo periódico de vulnerabilidades
- Gestión segura de logs



Agencia Nacional Digital



- Hardening de sistema operativo (Linux/Windows)

9. Conclusión

El hardening de plataformas tecnológicas no debe ser entendido como una actividad puntual, sino como un proceso continuo y evolutivo que requiere la integración de controles técnicos, administrativos y operativos. La ausencia de acceso al código fuente no limita la capacidad de fortalecer la seguridad, siempre que se adopten buenas prácticas, estándares reconocidos y una visión proactiva frente a los riesgos.

La implementación de las medidas aquí descritas permitirá reducir significativamente la superficie de ataque, mitigar vulnerabilidades críticas y fortalecer la resiliencia de la infraestructura tecnológica institucional. Recomendamos complementar estas acciones con auditorías periódicas, ejercicios de Red Team y la implementación de capacidades de monitoreo continuo (SOC/SIEM), alineadas con la estrategia de ciberseguridad institucional.