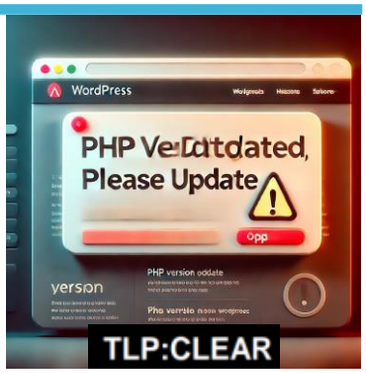
# Advertencia de Seguridad

## Actualización de PHP: Importancia de mantener versiones soportadas.

COLCERT AD-1709-018

PHP es uno de los lenguajes de programación más utilizados para el desarrollo de sitios web y aplicaciones, lo que lo convierte en un blanco atractivo para atacantes. Sin embargo, muchas organizaciones y desarrolladores pasan por alto la importancia de mantener sus versiones de PHP actualizadas. Al utilizar versiones desactualizadas o fuera de soporte, se exponen a vulnerabilidades conocidas que pueden ser explotadas fácilmente, generando riesgos críticos y la continuidad de los servicios de las entidades/organizaciones.




### Riesgos claves de utilizar versiones desactualizadas de PHP:

**Falta de soporte y parches de seguridad:**
Las versiones de PHP que han alcanzado su "End of Life" (EOL) ya no reciben parches de seguridad por parte de los desarrolladores oficiales. Esto significa que cualquier vulnerabilidad descubierta en estas versiones queda expuesta a ser explotada indefinidamente. Los atacantes suelen aprovechar estas debilidades conocidas para infiltrarse en los sistemas.

**Ejecución remota de código (RCE):**
Las versiones antiguas de PHP a menudo contienen fallas críticas que permiten la ejecución remota de código. Esto significa que un atacante podría enviar instrucciones maliciosas al servidor web y tomar control de este, lo que les permitiría comprometer datos sensibles, modificar la configuración del servidor web o incluso utilizar los recursos del servidor para realizar ataques más amplios, como lanzar ataques DDoS.

**Denegación de servicio (DoS):**
Al no gestionar adecuadamente las solicitudes o manejar incorrectamente la memoria, las versiones desactualizadas de PHP permiten que los atacantes sobrecarguen los servidores web y causen una interrupción total de los servicios. Estos ataques de denegación de servicio pueden ser críticos para la disponibilidad de un sitio web, afectando la operación diaria y la satisfacción del usuario.

**Vulnerabilidad a inyecciones de código:**
Los errores en las versiones antiguas de PHP facilitan la inyección de código malicioso, lo que puede ser utilizado por los atacantes para insertar comandos en el servidor web o para manipular el comportamiento de las aplicaciones web. Esto compromete la integridad de la aplicación y puede conducir a la modificación o robo de datos.

**Exposición de datos sensibles:**
Sin las actualizaciones de seguridad más recientes, los servidores web que ejecutan versiones antiguas de PHP pueden exponer información confidencial. Las vulnerabilidades conocidas podrían permitir a los atacantes acceder a datos de usuarios, contraseñas, información financiera o cualquier dato sensible almacenado en la aplicación. Este tipo de fugas de datos podría generar problemas legales y graves daños reputacionales para las entidades/empresas afectadas.

### Impacto

**Interrupciones en el servicio:** Los ataques de denegación de servicio podrían afectar la disponibilidad de su sitio, lo que impactaría negativamente la experiencia del usuario.

**Filtración de datos sensibles:** Los atacantes podrían obtener acceso a información confidencial, afectando la privacidad de los usuarios y poniendo a la organización en riesgo de incumplimientos normativos y multas.

**Pérdida de confianza**: Una brecha de seguridad podría afectar la confianza de los clientes en la seguridad de su sitio web, lo que conlleva un impacto significativo en la reputación de su marca.

COLCERT

# Advertencia de Seguridad

## Actualización de PHP: Importancia de mantener versiones soportadas.

*COLCERT AD-1709-018*

### Mejores prácticas de seguridad adicionales:

**Fortalecimiento del servidor (Server Hardening):** Configurar el servidor web de manera segura, limitando capacidades innecesarias y habilitando controles de acceso robustos.

**Cifrado de comunicaciones (TLS/SSL):** Asegurar que todas las comunicaciones entre el servidor web y los usuarios se realicen a través de conexiones cifradas, protegiendo los datos en tránsito

**Gestión Segura de Contraseñas y Autenticación:** Implementar políticas de contraseñas seguras y autenticación multifactor (MFA) en todos los puntos de acceso críticos.

### CVE relacionados con vulnerabilidades críticas en versiones desactualizadas de PHP:

| CVE | Criticidad (CVSS) | Tipo de vulnerabilidad | Año | Impacto |
|---|---|---|---|---|
| CVE-2019-11043 | 9.8 | Ejecución remota de código (Remote Code Execution) | 2019 | Compromiso total del servidor |
| CVE-2020-7069 | 9.8 | Ejecución remota de código (RCE) | 2020 | Compromiso total del servidor |
| CVE-2018-19518 | 9.8 | Ejecución remota de código (RCE) | 2018 | Compromiso del sistema, control total |
| CVE-2020-7067 | 7.5 | Desbordamiento de búfer | 2020 | Ejecución de código |
| CVE-2018-19935 | 7.5 | Escalación de privilegios | 2018 | Escalación de privilegios en el sistema |
| CVE-2021-21703 | 7.5 | Fuga de datos (Exfiltración de información) | 2021 | Exposición de información confidencial |
| CVE-2019-11044 | 6.1 | Inyección de código SQL (SQL Injection) | 2019 | Exposición de datos sensibles |
| CVE-2022-31625 | 6.1 | Inyección de comandos | 2022 | Explotación de comandos en servidores |
| CVE-2020-7070 | 5.3 | Falsificación de solicitudes en servidores (CSRF) | 2020 | Fuga de información |
| CVE-2020-7071 | 4.3 | Desbordamiento de búfer | 2020 | Compromiso parcial del servidor |

Fuente: https://cve.mitre.org

### Recomendaciones para mitigar los riesgos de PHP desactualizado:

**Actualización inmediata:** Si su sitio web o aplicación está utilizando una versión obsoleta de PHP, es necesario actualizar a una versión soportada lo antes posible (PHP 8.0 o superior). Las versiones modernas incluyen parches de seguridad críticos y mejoras en el rendimiento que pueden prevenir la mayoría de los ataques descritos.

**Mitigaciones temporales:**

- ❑ Restringir el acceso al servidor web mediante controles de acceso estrictos.
- ❑ Desactivar funciones potencialmente peligrosas en PHP que no sean necesarias para el funcionamiento de su aplicación.
- ❑ Implementar un firewall de aplicaciones web (WAF) para filtrar y monitorear el tráfico malicioso hacia su servidor web.

**Auditoría del Código:** Aprovechar la oportunidad para realizar una auditoría completa del código de su aplicación PHP. Asegúrese de que su código sigue las mejores prácticas de seguridad, como la sanitización de entradas y el uso de parámetros preparados para proteger contra inyecciones SQL y otros tipos de ataques basados en inyección.

COLCERT

TIC

# Advertencia de Seguridad

## Actualización de PHP: Importancia de mantener versiones soportadas.

COLCERT AD-1709-018

**Monitoreo continuo:** Implementar herramientas de monitoreo y análisis de vulnerabilidades que revisen regularmente su entorno PHP en busca de nuevas amenazas. Esto permitirá reaccionar de forma proactiva ante cualquier vulnerabilidad nueva que pueda surgir.

**Implementar mecanismos de seguridad adicionales:**

- ❑ **Hardening del Servidor:** Asegúrese de que su servidor web está configurado de manera segura, limitando las capacidades innecesarias y habilitando mecanismos de control de acceso robustos.
- ❑ **TLS/SSL:** Garantizar que todas las comunicaciones entre el servidor y los usuarios se realicen a través de conexiones cifradas para proteger los datos en tránsito.
- ❑ **Gestión de Contraseñas y Autenticación:** Verificar que se estén utilizando contraseñas seguras y políticas de autenticación multifactor en los puntos de acceso crítico.

**Recomendaciones:**

- ❑ Mantener siempre PHP actualizado a la última versión soportada.
- ❑ Desactivar funciones peligrosas como exec() o eval().
- ❑ Activar configuraciones de seguridad como open_basedir y disable_functions en el archivo php.ini
- ❑ Implementar una estricta sanitización y validación de todas las entradas de usuario para prevenir inyecciones de código.
- ❑ Usar librerías seguras para bases de datos como PDO o mysqli, y habilitar HTTPS en todos los sitios para asegurar la transmisión de datos
- ❑ Mantener entornos aislados de desarrollo y producción separados para evitar que los errores de desarrollo afecten al entorno de producción
- ❑ Realizar pruebas de penetración periódicas para identificar y corregir vulnerabilidades.
- ❑ Utilizar herramientas de correlación de eventos (SIEM) para monitorear actividades del servidor web y detectar posibles intrusiones


Mantener PHP **actualizado** es crucial para la seguridad de las aplicaciones. Las versiones desactualizadas pueden tener **vulnerabilidades conocidas** que los atacantes pueden explotar para ejecutar código malicioso y comprometer información sensible.

**NIVEL DE RIESGO**

**ALTO**

**FUENTES:**


- PHP.net - Documentación oficial de PHP - https://www.php.net/
- Common Vulnerabilities and Exposures (CVE) - https://cve.mitre.org/
- National Vulnerability Database (NVD) - Base de datos nacional de vulnerabilidades
- OWASP (Open Web Application Security Project) - Seguridad en aplicaciones web
- CISA (Cybersecurity & Infrastructure Security Agency) - https://www.cisa.gov/
- SANS Institute - Investigaciones y formación en seguridad - https://www.sans.org/
- Tenable - Análisis y gestión de vulnerabilidades - https://www.tenable.com/


COLCERT