Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

TÉCNICAS DE PROGRAMACIÓN SEGURA EN PHP

Congreso Internacional de DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

EXPOSITOR

- Joffre Monar M.
- Ingeniero en Sistemas Informáticos
- Magister en Seguridad Telemática
- ESPOCH
- jmonar@espoch.edu.ec

Congreso Internacional de DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

INTRODUCCIÓN

Actualmente, la gran mayoría de aplicaciones web contienen vulnerabilidades de seguridad. Probablemente, se deba a falta de cultura de los desarrolladores o a la ausencia de técnicas de codificación específicas. Se analizó ciertos trabajos relacionados al tema, pero considero no definen técnicas de programación precisos, ni se enfocan a un lenguaje de programación específico. El presente trabajo propone un conjunto de técnicas de programación segura para reducir las vulnerabilidades en las aplicaciones web utilizando el entorno de desarrollo PHP. Para esto se determinaron diez vulnerabilidades usando las recomendaciones OWASP TOP-10. De éstas se plantean siete técnicas que consideramos más críticas y su respectiva forma de implementarlas. Se valida las técnicas y se mide las vulnerabilidades de una aplicación web en dos escenarios; con y sin la implementación de las técnicas propuestas. Los resultados mostrarán que el uso de las técnicas propuestas se relaciona significativamente con la cantidad de vulnerabilidades encontradas y por lo tanto mejora el nivel de seguridad de las aplicaciones web en entornos PHP.

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

OBJETIVOS DEL ESTUDIO

GENERAL

Proponer un conjunto de técnicas de programación segura en entorno PHP para reducir el riesgo de recibir ataques informáticos en aplicaciones web.

ESPECÍFICOS

- Determinar las vulnerabilidades más conocidas en el desarrollo de aplicaciones web.
- Analizar las metodologías, guías o buenas prácticas existentes para desarrollar aplicaciones web seguras.
- Proponer las técnicas que contemplen las vulnerabilidades más críticas en el desarrollo de aplicaciones web en entorno PHP.
- Verificar la mejora de la seguridad aplicando las técnicas desarrolladas.

METODOLOGÍA DE LA INVESTIGACIÓN

Determinación de vulnerabilidades de estudio

En esta investigación se tomó como base las diez vulnerabilidades más críticas de OWASP, de las cuales se seleccionaron siete:

Para la selección de estas vulnerabilidades se consideraron los errores más comunes de programación: validación de entradas, gestión de sesiones, cifrado de datos, Cross Site Scripting (XSS), instalación y configuración incorrecta del servidor web y base de datos y CSRF.

Vulnerabilidades a evaluar

N°	Vulnerabilidades
1	Inyección
2	Pérdida de autenticación y gestión de sesiones
3	Secuencia de comandos en sitios cruzados (XSS)
4	Configuración de seguridad incorrecta
5	Exposición de datos sensibles
6	Falsificación de peticiones en sitios cruzados (CSRF)
7	Redirecciones y reenvíos no validados

Determinación del nivel de seguridad

La seguridad fue determinada mediante el nivel de protección de la aplicación web contra las vulnerabilidades más conocidas, es decir, el número de vulnerabilidades altas, medias y bajas detectadas utilizando *Acunetix Vulnerability Scanner*.

Congreso Internacional de DESARROLLO DE SOFTWARE

METODOLOGÍA DE LA INVESTIGACIÓN

TECNICAS DE PROGRAMACION SEGURA

Técnica	Implementación	Vulnerabilidad que previene
T1. Parametrizar consultas	- Consultar - Insertar - Actualizar - Eliminar - Restricciones en los parámetros	- Injection
T2. Codificar los datos	- Sanear el HTML - Configurar las cookies para que sean httponly y secure	- Injection - XSS
T3. Validar todas las entradas	- Expresiones regulares - Validación y saneamiento Validación de direcciones IP Redirecciones y reenvíos no validados Enlaces desde $_SERVER Remover caracteres ASCII con Valor> 127	- Injection - XSS - Redireccion es y reenvíos no validados
T4. Implementar controles de identidad y autenticación	- Medidas durante la autenticación Prevenir ataques de fuerza bruta Captcha Recordar contraseña - Gestión de sesiones Inicio de sesión segura Estado de la sesión iniciada Cierre de sesión - Falsificación de peticiones en sitios cruzados (CSRF) Crear la clase y funciones Proteger un formulario POST	- Pérdida de autenticació n y gestión de sesiones - Falsificación de peticiones en sitios cruzados (CSRF)

Técnicas Propuestas

Para definir la propuesta se tomó como base las recomendaciones de: TOP-10 Controles Proactivos, la Guía de Pruebas OWASP, así como los Controles Estándar de Seguridad (ASVS); todo esto implementado con las técnicas de programación de PHP. Para cumplir con el objetivo de este estudio se proponen siete grupos de técnicas:

T5. Proteger los datos	- Algoritmo cifrado - Cifrado de contraseñas Hash de Contraseñas	- Exposición de datos sensibles
T6. Error y control de excepciones	- Respuestas try/catch - Respuestas de autentificación	- Todas las OWASP top 10
T7. Configuración adecuada	- Instalar un servidor web, PHP y MySQL en el servidor Mantener el software actualizado - Seguridad de la base de datos Configuración de la base de datos MYSQL Conexión con la base de datos Desconexión con la base de datos Excepciones - Archivos de configuración - Reporte de errores - Estar al tanto de las actualizaciones de los productos utilizados - Comprobar la información que se obtiene de las cabeceras HTTP - Verificar los servicios visibles - Buscar vulnerabilidades - Revisar la configuración HTTPS	- Todas las OWASP top 10

Instituto Superior Tecnológico del Azuay

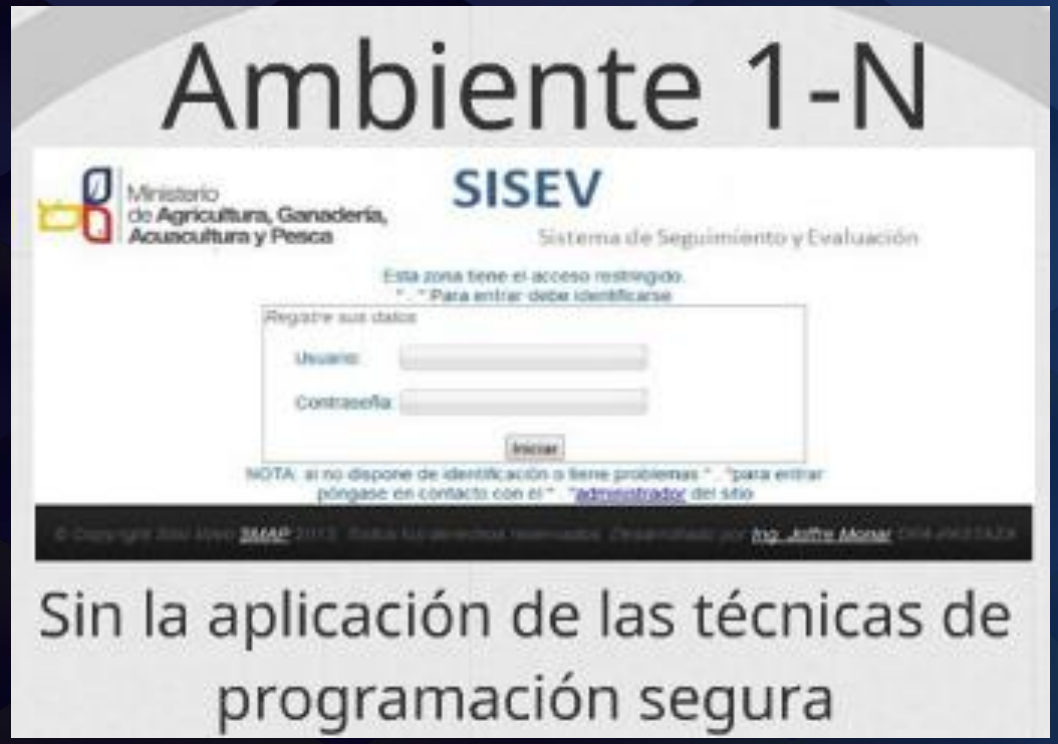
CIDE
Centro de Investigación y Desarrollo Ecuador

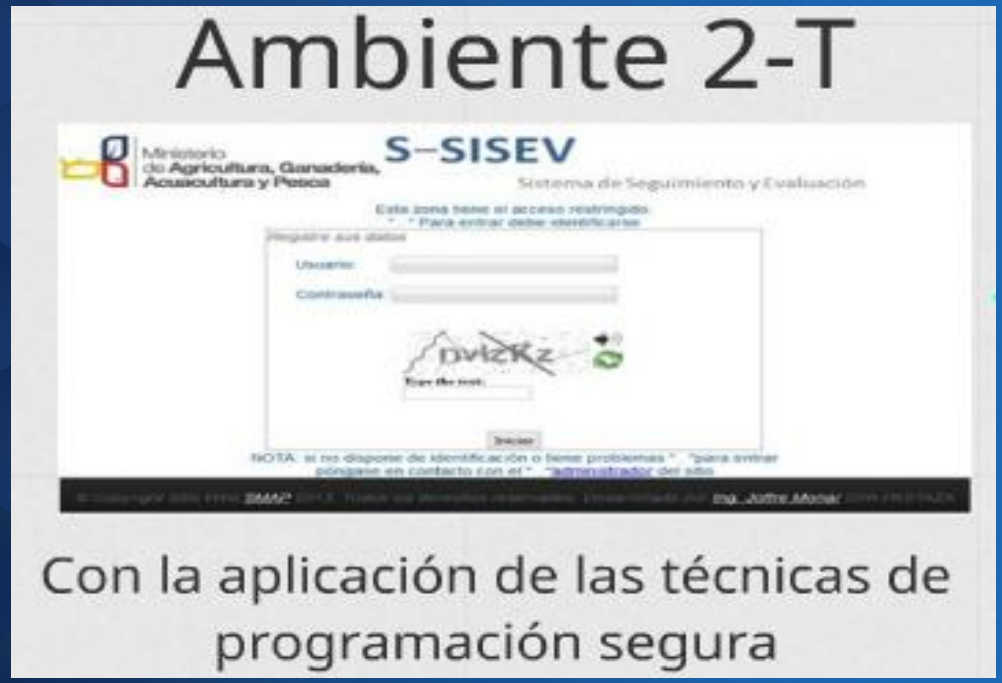
Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

METODOLOGÍA DE LA INVESTIGACIÓN

Escenarios de Prueba

Para validar las técnicas propuestas se utilizó el sistema SISEV desarrollado en PHP 5.6.15, que automatiza el seguimiento y evaluación de las actividades de programas y proyectos de una Institución Pública del Ecuador; cuenta con 5 módulos y tiene un tamaño de 12.120 líneas de código.





Congreso Internacional de DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia CET-BOLIVIA

RESULTADO Y DISCUSIONES

Resultados obtenidos en la etapa de determinación de vulnerabilidades

VULNERABILIDADES DETECTADAS ANTES DE LA APLICACIÓN DE LAS ESTRATEGIAS DE PROGRAMACIÓN SEGURAS.

Vulnerabilidad	Grado	Cant.	Descripción
Blind SQL Injection	Alto	3	Técnica de ataque que utiliza Inyección SQL
PHP Hash Collision denial of service vulnerability	Alto	1	Vulnerabilidad de tipo denegación de servicio
Directory listing	Medio	63	Es una función que lista todos los archivos cuando no hay un archivo de índex
HTML form without CSRF protection	Medio	2	Formulario HTML sin protección CSRF
Clickjacking	Bajo	1	Secuestro de sesión
Cookie without HttpOnly flag set	Bajo	1	Puede causar un desbordamiento de memoria.
Login page password-guesting attack	Bajo	1	Página de inicio de sesión puede ser atacada
OPTIONS method is enabled	Bajo	1	El método OPTIONS está activado
Possible relative path overwrite	Bajo	9	Posible sobre escritura de ruta
TRACE method is enabled	Bajo	1	El método TRACE está habilitado

Análisis e Interpretación de resultados:
Diez vulnerabilidades fueron establecidas, de las cuales dos son de alto grado, dos de grado medio y seis de bajo grado de incidencia en la seguridad de una aplicación web

RESULTADO Y DISCUSIONES

Resultados de las vulnerabilidades detectadas

VULNERABILIDADES DETECTADAS DESPUÉS DE LA APLICACIÓN DE LAS ESTRATEGIAS DE PROGRAMACIÓN SEGURAS

Vulnerabilidad	Grado	Cant.	Descripción
Directory listing	Medio	8	Es una función de servidor web que muestra una lista de todos los archivos cuando no hay un archivo de índice
Possible relative path overwrite	Bajo	1	Posible sobre escritura de ruta
TRACE method is enabled	Bajo	1	El método TRACE está habilitado

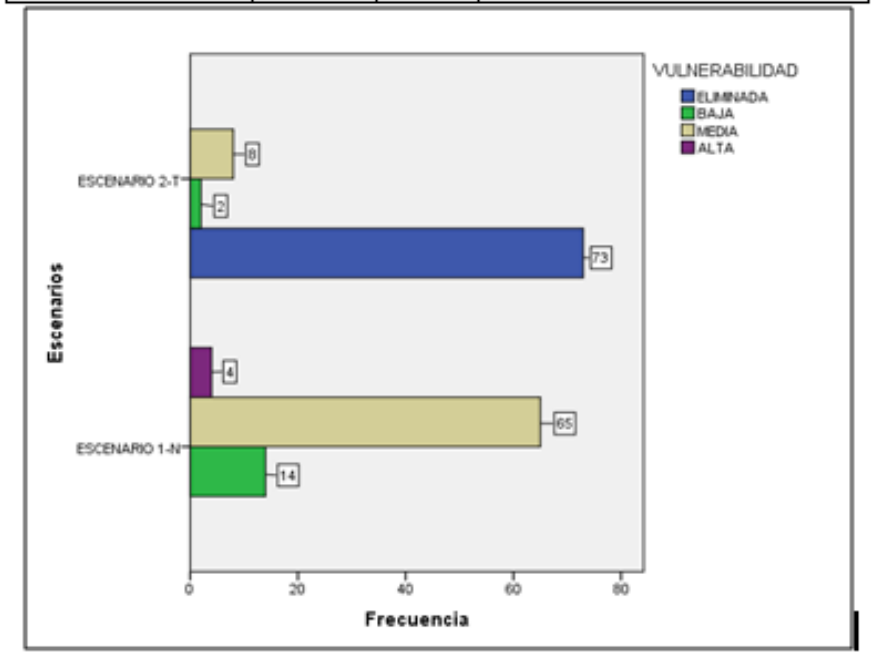


Figura 1: Frecuencia de vulnerabilidades en dos escenarios

Análisis e Interpretación de resultados:

En la Figura 1 se ilustra las frecuencias numéricas de las vulnerabilidades observadas antes y después de la utilización de las estrategias de programación segura, en la que se evidencia la reducción del 100%, 87,69% y 85,71% de las vulnerabilidades altas, medias y bajas respectivamente. Obteniendo una mejora promedio del 91,13%.

Congreso Internacional de DESARROLLO DE SOFTWARE

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

RESULTADO Y DISCUSIONES

Prueba de Hipótesis

Ha: La utilización de estrategias de programación seguras se relaciona significativamente con la cantidad de vulnerabilidades encontradas en la aplicación web.

H0: La utilización de estrategias de programación seguras no se relaciona significativamente con la cantidad de vulnerabilidades encontradas en la aplicación web.

TABLA DE CONTINGENCIA ESCENARIOS - VULNERABILIDAD

ESCENARIOS		VULNERABILIDAD				Total
		Eliminada	Baja	Media	Alta	
Escenario 1-N	Recuento	0	14	65	4	83
	Recuento esperado	36,5	8,0	36,5	2,0	83,0
	% del total	0,0%	16,9%	78,3%	4,8%	100,0%
Escenario 2-T	Recuento	73	2	8	0	83
	Recuento esperado	36,5	8,0	36,5	2,0	83,0
	% del total	88,0%	2,4%	9,6%	0,0%	100,0%
Total	Recuento	73	16	73	4	166
	Recuento esperado	73,0	16,0	73,0	4,0	166,0

RESULTADOS DE LA PRUEBA DE CHI-CUADRADO

	Valor	gl	Sig. asintótica (2 caras)
Chi-cuadrado de Pearson	130,507	3	0,000

El valor de Chi cuadrado es de 130,507 con 3 grados de libertad y un nivel de significancia asintótica P=0,000, P>0,05 por lo que se rechaza la H0 y se acepta la Ha, es decir que a utilización de estrategias de programación seguras se relaciona significativamente con la cantidad de vulnerabilidades encontradas en la aplicación web y por lo tanto mejora el nivel de seguridad de las aplicaciones web en entorno PHP

Instituto Superior Tecnológico del Azuay

CIDE
Centro de Investigación y Desarrollo Ecuador

Centro de Estudios Transdisciplinarios Bolivia
CET-BOLIVIA

CONCLUSIONES

- El principal aporte de este trabajo de investigación es proponer treinta y cinco técnicas de programación segura organizada en siete grupos, las cuales se enfocan a evitar algunos de los errores muy comunes en programación, tales como: validación de entradas, gestión de sesiones, cifrado de datos, Cross Site Scripting (XSS), instalación y configuración incorrecta del servidor web y base de datos, y CSRF. Por lo que se logra reducir el riesgo de recibir ataques informáticos en las aplicaciones web en entornos PHP.
- En este trabajo se comprueba que la utilización de técnicas de programación segura se relaciona significativamente con la cantidad de vulnerabilidades encontradas en la aplicación web y por lo tanto mejora el nivel de seguridad de las aplicaciones web en entorno PHP.
- El presente estudio puede ser complementado añadiendo otras técnicas de programación segura para cubrir vulnerabilidades que no fueron consideradas en el presente documento tales como: referencia directa insegura a objetos, ausencia de control de acceso a funciones y uso de componentes con vulnerabilidades conocidas.