PRUEBAS DE SEGURIDAD APLICACIÓN ESCUELA_INTERNA_FORMADORES

La evaluación de seguridad realizadas sobre este recurso fue desarrollada en un contexto interno, dado que las aplicaciones e infraestructura deben tener el mismo nivel de protección interno/externo garantizando de esta manera la protección de la información.

A continuación, se expone las vulnerabilidades halladas y las correspondientes evidencias que sustentan lo indicado, de igual manera y para mayor detalle se adjunta el informe de vulnerabilidades WEB (Owasp Zap, WhatWeb, Nmap y Nikto).

URL evaluada

https://escuelainternaformadorespru.educacionbogota.edu.co

1. Vulnerabilidades a nivel de Aplicación

a. Servidor web vulnerable.

Target IP	172.16.12.12
Target hostname	escuelainternaformadorespru.educacionbogota.edu.co
Target Port	443
HTTP Server	Apache/2.4.37 (centos) OpenSSL/1.1.1k
Site Link (Name)	https://escuelainternaformadorespru.educacionbogota.edu.co:443/
Site Link (IP)	https://172.16.12.12:443/

La versión del servidor web de apache evidencia diferentes vulnerabilidades como elución de los sistemas de autenticación, DoS, ominión del tiempo expiración de sesión, mas información consulte los siguientes CVE.

CVE-2019-0215
CVE-2019-0190
CVE-2018-17199

Solución esperada: Consulte cada uno de los CVE relacionados anteriormente y de solución de acuerdo con lo expuesto en cada uno de ellos, o lleve a la ultima versión estable y sin vulnerabilidades.

b. El parámetro X-Frame-Options no configurado

URI	%2f
HTTP Method	GET
Description	The anti-clickjacking X-Frame-Options header is not present.
Test Links	https://escuelainternaformadorespru.educacionbogota.edu.co:443%2f https://172.16.12.12:443%2f
OSVDB Entries	OSVDB-0

El parámetro X-Frame-Options: debe tener el valor deny en caso contrario el servidor debe rechazar la solicitud.

Solución esperada: El valor deny es para evitar posibles ataques de arrastrar y soltar clickjacking en navegadores.

c. . El parámetro Strict-Transport-Security no se encuentra definido en los encabezados de respuesta HTTP.

URI	%2f
HTTP Method	GET
Description	The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
Test Links	https://escuelainternaformadorespru.educacionbogota.edu.co:443%2f https://172.16.12.12:443%2f
OSVDB Entries	OSVDB-0

es una característica de seguridad permite a un sitio web indicar a los navegadores que sólo se debe comunicar con HTTPS.

Solución esperada: configurar el parámetro Strict-Transport-Security de acuerdo con lo recomendado en.

https://developer.mozilla.org/es/docs/Web/HTTP/Headers/Strict-Transport-Security

d. Configurar el parámetro Expect-CT

URI	%2f
HTTP Method	GET
Description	The site uses SSL and Expect-CT header is not present.
Test Links	https://escuelainternaformadorespru.educacionbogota.edu.co:443%2f https://172.16.12.12:443%2f
OSVDB Entries	OSVDB-0

Este permite a los sitios informar y/o hacer cumplir los requerimientos de Transparencia de Certificados digitales.

Solución esperada: configurar el parametro Expect-CT de acuerdo con lo recomendado en. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Expect-CT

e. Vulnerabilidad de desbordamiento de buffer en la versión OpenSSL usada.

```
[ OpenSSL ]
        The OpenSSL Project is a collaborative effort to develop a
        robust, commercial-grade, full-featured, and Open Source
        toolkit implementing the Secure Sockets Layer (SSL v2/v3)
        and Transport Layer Security (TLS v1) protocols as well as
        a full-strength general purpose cryptography library.

        Version      : 1.1.1k
        Website     : http://www.openssl.org/
```

Las cadenas ASN.1 se representan internamente en OpenSSL como una estructura ASN1_STRING que contiene un búfer que contiene los datos de la cadena y un campo que contiene la longitud del búfer. Esto contrasta con las cadenas normales en C, que se representan como un búfer para los datos de la cadena que termina con un byte NUL (0). Aunque no es un requisito estricto, las cadenas ASN.1 que se analizan utilizando las propias funciones "d2i" de OpenSSL (y otras funciones de análisis similares), así como cualquier cadena cuyo valor se haya establecido con la función ASN1_STRING_set(), terminarán además con NUL en la matriz de bytes de la estructura ASN1_STRING. Sin embargo, es posible que las aplicaciones construyan directamente estructuras ASN1_STRING válidas que no terminen en NUL la matriz de bytes estableciendo directamente los campos "data" y "length" en la matriz ASN1_STRING. Esto también puede ocurrir utilizando la función ASN1_STRING_set0(). Se ha descubierto que numerosas funciones de OpenSSL que imprimen datos ASN.1 asumen que la matriz de bytes ASN1_STRING terminará en NUL, aunque esto no está garantizado para cadenas que han sido construidas directamente. Cuando una aplicación solicita que se imprima una estructura ASN.1, y cuando esa estructura ASN.1 contiene ASN1_STRINGs que han sido construidos directamente por la aplicación sin NUL terminando el campo "data", **entonces puede ocurrir un desbordamiento del buffer de lectura**. Lo mismo puede ocurrir durante el procesamiento de las restricciones de nombre de los certificados (por ejemplo, si un certificado ha sido construido directamente por la aplicación en lugar de cargarlo a través de las funciones de análisis de OpenSSL, y el certificado contiene estructuras ASN1_STRING sin terminación NUL). También puede ocurrir en las funciones X509_get1_email(), X509_REQ_get1_email() y X509_get1_ocsp(). Si un actor malicioso puede hacer que una aplicación construya directamente un ASN1_STRING y luego lo procese a través de una de las funciones OpenSSL afectadas, entonces se podría dar este problema. Esto podría resultar en un fallo (causando un ataque de denegación de servicio). También podría dar lugar a la divulgación del contenido de la memoria privada (como claves privadas o texto plano sensible). Corregido en OpenSSL 1.1.1l (Afectado 1.1.1-1.1.1k). Corregido en OpenSSL 1.0.2za (Afectado 1.0.2-1.0.2y).

Mas información **CVE-2021-3712**

Solución esperada: Actualice a la ultima versión estable y sin vulnerabilidades.

f. Revela Información de Identificación Personal.

https://escuelainternaformadorespru.educacionbogota.edu.co (1)

PII Disclosure (1)

▶ GET https://escuelainternaformadorespru.educacionbogota.edu.co/moodle_EM/

La herramienta de evaluación de vulnerabilidades, identifica un número de tarjeta de credito/debito de la franquicia maestro.

Other info	Credit Card Type detected: Maestro Número de identificación bancaria: 564964 Marca: MAESTRO Categoría: Editor:
Evidence	564964511815

Solución esperada: Verifique la información expuesta a traves de la apliación, si existe el numero de evidencia eliminelo y garantice que la aplicaión web no exponga información privada.

g. Vulnerabilidad de XSS Cross Site Scripting (Reflected).

https://escuelainternaformadorespru.educacionbogota.edu.co (2)

Cross Site Scripting (Reflected) (1)

▶ GET https://escuelainternaformadorespru.educacionbogota.edu.co/moodle_EM/course/search.php?areaids=core_course-course&q=%22+onMouseOver%3D%22alert%281%29%3B

Attack	" onMouseOver="alert(1);
Evidence	" onMouseOver="alert(1);

Cross_site Scripting (XSS) es una técnica de ataque que comprende hacer eco del código que fue proporcionado por el atacante en la instancia del navegador de un usuario. Una instancia de navegador puede ser un cliente de navegador web corriente, o un objeto de navegador integrado e un producto de software, como el navegador que se encuentra dentro de WinAmp, un lector de RSS o un cliente de correos electrónicos. El código por sí mismo se encuentra escrito en HTML/JavaScript, pero también puede extenderse a VBScript, ActiveX, Jave, Flash o cualquier otra tecnología que sea compatible con el navegador.

Cuando un atacante consigue el navegador de un usuario para poder ejecutar su código, el código se ejecutará dentro del contexto de seguridad (o zona) del sitio web de hospedaje. Con este nivel de privilegio, el código tiene la extensión de leer, modificar y transmitir cualquier dato que sea sensible al que pueda ingresar al navegador.

Solución esperada:
Fase: Implementación

Para cada una de las páginas web que se origina, utilice y especifique una codificación de caracteres como ISO-8859 o UTF-8. Cuando no se puede especificar una codificación, el navegador web podría seleccionar una codificación distinta adivinando que codificación está siendo utilizada en verdad por

la página web. Esto puede permitir que el navegador web trate varias secuencias como especiales, abriendo al cliente a leves ataques XSS. Consulte CWE-116 para conseguir más mitigaciones con respecto a la codificación/escape.

Para ayudar a mitigar los ataques XSS contra las cookies de la sesión del usuario, es necesario establecer que la cookie de la sesión sea HttpOnly. En navegadores que son compatibles con la característica HttpOnly (como las versiones más actualizadas de internet explorer y firefox), esta característica puede prevenir que la cookie de sesión del usuario sea accesible para las secuencias de comandos del lado del cliente malignas que utilizan document.cookie. Esta no es una solución muy completa, ya que HttpOnly no es compatible con todos los navegadores que hay. Más importante aún, XMLHTTPRequest y otras tecnologías poderosas de navegador otorgan acceso de lectura a los encabezados HTTP, incluido el encabezado Set-Cookie en el cual se establece el indicador HttpOnly.

Asuma que toda la entrada es maliciosa. Utilizar una estrategia de validación de entradas de tipo "aceptar lo bueno conocido", es decir, utilizar una lista de entradas aceptables que se ajusten estrictamente a las especificaciones. Rechace cualquier entrada que no se adapte de forma estricta a las especificaciones, o cambielas por algo que sí lo haga. No confíe exclusivamente en la búsqueda de entradas maliciosas o malformadas (es decir, no confíe en una lista de denegación). Sin embargo, las listas de denegación pueden ser útiles para detectar posibles ataques o para determinar qué entradas están tan malformadas que deben ser rechazadas directamente.

Al realizar la validación de entrada, usted debe considerar todas las propiedades potencialmente destacadas, incluida la longitud, el tipo de entrada, el rango completo de valores aceptables, las entradas faltantes o adicionales, la sintaxis, el sentido entre los campos que se encuentran relacionados y la conformidad con todas las reglas comerciales. Como ejemplo de lógica de regla de negocio, "barco" puede ser sintácticamente válido porque sólo contiene caracteres alfanuméricos, pero no es válido si se esperan colores como "rojo" o "azul".

Asegúrese de realizar la validación de entradas en interfaces bien definidas dentro de la aplicación. Esto ayudará a cuidar la aplicación, incluso si un elemento se utiliza de nuevo o traslada a otro sitio.

h. Vulnerabilidad de inyección SQL.

Inyección SQL (1)

▶ GET
https://escuelainternaformadorespru.educacionbogota.edu.co/moodle_EM/user/view.php?course=3&id=5-2

Alert description	Inyección SQL puede ser posible
Other info	Los resultados de la página original se replicaron correctamente utilizando la expresión [5-2] como valor de parámetro El valor de parámetro que se está modificando se eliminó de la salida HTML a efectos de comparación.
Parameter	id
Attack	5-2

Solución esperada:

- No confíe en los datos de entrada del lado del cliente, incluso si existe una validación del

lado del cliente.

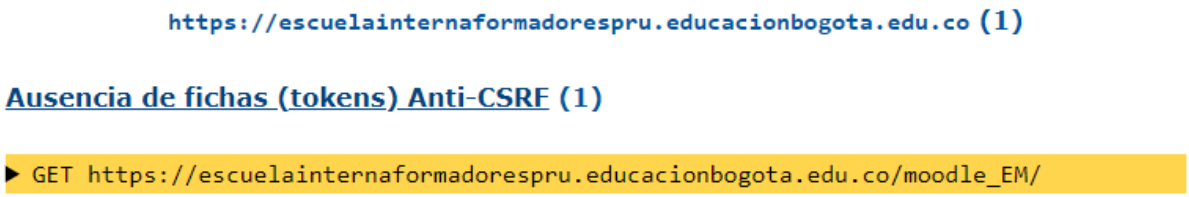
- Como norma general, escriba la verificación de los datos en el lado del servidor.
- Si la aplicación usa JDBC, use PreparedStatement o CallableStatement, con parámetros pasados por '?'
- Si la aplicación usa ASP, use objetos de comando ADO con verificación de tipo fuerte y consultas parametrizadas.
- Si se pueden usar los procedimientos almacenados de la base de datos, utilícelos.
- ¡*No* concatene cadenas en consultas en el procedimiento almacenado, o use 'ejec', 'ejex immediate' o una función equivalente!
- No cree consultas SQL dinámicas mediante la concatenación de cadenas simples.
- Escape todos los datos recibidos del cliente.
- Aplique una 'lista de permitidos' para caracteres permitidos o una 'lista de denegados' para caracteres no permitidos en la entrada del usuario.
- Aplique el privilegio mínimo utilizando el usuario de base de datos con menos privilegios posible.
- En particular, evite usar los usuarios de la base de datos 'sa' o 'db-owner'. Esto no elimina la inyección SQL, pero minimiza su impacto.
- Otorgue el acceso mínimo a la base de datos que sea necesario para la aplicación.

i. Política CSP sin configurar.

La política de seguridad de contenidos (CSP) es una capa de seguridad añadida que ayuda a detectar y mitigar ciertos tipos de ataques, como los de Cross Site Scripting (XSS) y los de inyección de datos.

Solución esperada: establecer la cabecera Content-Security-Policy del servidor WEB.

j. No se encuentran configurado CSRF tokens.



Un token CSRF es un valor único, secreto e impredecible que es generado por la aplicación del lado del servidor y transmitido al cliente de tal manera que se incluye en una solicitud HTTP posterior realizada por el cliente. Cuando se realiza la solicitud posterior, la aplicación del lado del servidor valida que la solicitud incluya el token esperado y rechaza la solicitud si el token falta o no es válido. Los tokens CSRF pueden prevenir los ataques CSRF al hacer imposible que un atacante construya una petición HTTP totalmente válida y adecuada para alimentar a un usuario víctima. Dado que el atacante no puede determinar o predecir el valor del token CSRF de un usuario, no puede construir una solicitud con todos los parámetros necesarios para que la aplicación atienda la solicitud.

Solución esperada:

- Genere un nonce único para cada formulario, coloque el nonce en el formulario y verifique
- el nonce al recibir el formulario. Asegúrese de que el nonce no es predecible (CWE-330).
- No utilice el método GET para ninguna solicitud que desencadene un cambio de estado.
- Compruebe la cabecera HTTP Referer para ver si la solicitud se origina en una página

esperada.
- Los tokens CSRF no deben ser transmitidos usando cookies.

Información adicional:
- Cross-Site Request Forgery Prevention - OWASP Cheat Sheet Series
- CSRF tokens | Web Security Academy (portswigger.net)

ANEXOS

- Análisis de vulnerabilidades WEB
    - Informe Owasp Zap
    - Informe Nikto
    - Informe WhatWeb

Elaborado:
Juan Carlos Parra M.
Especialista Seguridad Digital