ELEMENTOS DE SEGURIDAD A TOMAR EN CUENTA EN LA IMPLEMENTACIÓN DE UN SITIO WEB

Autor: Ing. Mario Alberto Arredondo Guzmán
Revisor: MATIE. Juan Manuel Castillejos Reyes
Subdirección de Visibilidad Web, DGTIC, UNAM.

Octubre 2022

Resumen

En este documento se expone una lista mínima de elementos a considerar para fortalecer la seguridad de los sitios web, tomando en cuenta diferentes niveles o capas, desde la infraestructura, el software y hasta políticas de uso o buenas prácticas, involucrando a varios perfiles que intervienen en la publicación de un sitio Web.

1. Contraseñas seguras y recomendaciones para su uso

Los sistemas web[1] necesitan un mecanismo de autenticación que involucre el uso de contraseñas seguras para mitigar posibles intrusiones que pongan en riesgo la información. A continuación se enlistan algunas recomendaciones para su uso:

- Se recomienda controlar el primer acceso al sistema con una contraseña segura[2], es decir que otras personas no la puedan determinar fácilmente, adivinándola o utilizando programas automáticos. Ésta servirá como ejemplo del tipo de contraseña a utilizar.

- Generar de forma automática contraseñas aleatorias.
- Establecer contraseñas de 8 a 12 caracteres, como mínimo.
- Combinar caracteres, números, símbolos, letras en mayúsculas y minúsculas. Se sugiere evitar combinaciones obvias, secuencias de números continuos, palabras conocidas o

[1] Se entiende como sistema web a aquella herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. (https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)

[2] Una contraseña segura debe cumplir con las siguientes características:
Larga: al menos, 12 caracteres; y cuanto más larga es más segura.
Aleatoria: emplean una combinación de letras, números, mayúsculas y símbolos para formar una cadena impredecible de caracteres que no se asemejen a palabras o nombres.
Única: Debe ser única para cada cuenta a fin de reducir su vulnerabilidad en caso de hackeo.

asociadas al usuario: fecha de nacimiento, apellidos o el nombre de alguna mascota, por ejemplo.

El sistema debe verificar la existencia de estos elementos al modificar o crear una nueva contraseña de usuario.

- Autenticación en dos factores. Este método incrementa notablemente la seguridad. Se requiere la confirmación del usuario al entrar en una sesión para verificar que es él quien realmente está solicitando el acceso.

  Para su implementación se puede hacer uso de mensajes SMS o de correo electrónico. También es posible utilizar otras opciones, como factores biométricos; todo dependerá del dispositivo en el que se haya instalado el sistema.

- Es recomendable limitar el número de intentos de acceso para reducir la posibilidad de ataques de terceros. Por ello se debe de fijar un máximo de oportunidades y bloquear temporal o definitivamente al usuario en caso de que las haya sobrepasado.

2. Cambiar el nombre del usuario administrador por defecto

Si se utilizan Sistemas de Gestión de Contenidos (CMS por sus siglas en inglés, Content Management System), es altamente recomendable cambiar el nombre del usuario "*Administrador*" que viene por defecto; y de esta forma se mitigará un posible ataque por fuerza bruta[3] utilizando el nombre de usuario conocido.

3. Validar las entradas de los formularios

Si cuenta con mecanismos de entrada de información, como formularios de contacto o encuestas, es necesario validarlos para que sólo se almacenen los datos adecuados. Este tipo de validaciones son una primera barrera para evitar ataques de SQL Injection, por ejemplo.

Una primera capa de validación podría implementarse desde el navegador del usuario, antes de enviar la información al servidor. Aunque existen lenguajes de programación -como JavaScript- que nos permiten hacer este tipo de validaciones, siempre es necesario realizar esta práctica del lado del servidor porque no es posible tener control de la procedencia de los datos o del navegador de usuario.

[3] Un ataque de fuerza bruta es un intento de averiguar una contraseña o un nombre de usuario, o de encontrar una página web oculta o la clave utilizada para cifrar un mensaje, mediante un enfoque de prueba y error, con la esperanza de acertar.

¿Qué es un ataque de fuerza bruta?, sitio web de Kaspersky
https://latam.kaspersky.com/resource-center/definitions/brute-force-attack

4. Usar CAPTCHA para formularios

El CAPTCHA[4] es una medida de seguridad “pregunta-respuesta” que ayuda a proteger las entradas del sistema de posible spam y del descifrado de contraseñas, solicitando al usuario demuestre que es un ser humano, no una computadora (robot) tratando de acceder.

Existen diferentes opciones para el uso de CAPTCHA en un sistema, una de las más populares es reCAPTCHA de Google. reCAPTCHA se puede utilizar en desarrollos propios “a medida” y en diferentes sistemas administradores de contenido, como WordPress o Drupal.

Para implementar reCAPTCHA en un sistema diseñado a la medida, se puede consultar la documentación del lenguaje o framework utilizado pues cada uno tienen sus propios métodos.

Si se trata de un Sistema Administrador de Contenidos (Content Management System), como WordPress o Drupal, existen plugins o módulos que facilitan la configuración del CAPTCHA o reCAPTCHA. Algunas opciones son:

WordPress

- reCaptcha

  https://es.wordpress.org/plugins/google-captcha/
- Advanced Google reCAPTCHA

  https://es.wordpress.org/plugins/advanced-google-recaptcha/
- Really Simple CAPTCHA

  https://es.wordpress.org/plugins/really-simple-captcha/

Drupal

- CAPTCHA

  https://www.drupal.org/project/captcha
- reCAPTCHA v3

  https://www.drupal.org/project/recaptcha_v3

[4] CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart: test de Turing público y automático para distinguir a los ordenadores de los humanos). Un CAPTCHA te ayuda a protegerte del spam y del descifrado de contraseñas pidiéndote que completes una simple prueba que demuestre que eres humano y no un ordenador que intenta acceder a una cuenta protegida con contraseña.

Developer's Guide, Google
https://developers.google.com/recaptcha/intro

5. Configurar adecuadamente las HTTP cookies

Las cookies contienen datos que el servidor envía al navegador web de un usuario, éste los guarda y regresa en cada petición. Por lo general, estos datos se usan para indicar al servidor qué peticiones provienen del mismo navegador web, permitiendo entre otras cosas: mantener una sesión de usuario activa en un sistema, personalizar las preferencias del usuario y rastrear su comportamiento en un sitio web.

Las cookies mantienen estados de sesión mediante la información que contienen, ya que HTTP es un protocolo sin estados definidos. Es recomendable utilizar otras alternativas para almacenar la información que se guarda en cookies. Actualmente existen opciones más seguras, como Web Storage[5] (*localStorage* y *sessionStorage*) e IndexedDB[6].

Sin embargo, si se va a hacer uso de cookies, se recomienda nunca guardar información *sensible* en ellas y recurrir a las directivas Secure y HttpOnly.

6. Usar certificados SSL, TLS y protocolo HTTPS

SSL son las siglas de *Secure Sockets Layer* (capa de sockets seguros). Esta tecnología que ayuda a mantener una conexión segura en Internet y protege la información que se envía entre dos sistemas, utilizando algoritmos de cifrado para codificar la información que se transmite entre ellos.

El protocolo TLS (*Transport Layer Security*) o seguridad de la capa de transporte es una versión actualizada y más segura que SSL.

El protocolo HTTPS (*Hyper Text Transfer Protocol Secure*) o protocolo seguro de transferencia de hipertexto permite una conexión segura entre el servidor web y el navegador. Para usar el protocolo HTTPS en un sitio web es necesario adquirir un certificado digital SSL, éste autentica la identidad de un sitio web y habilita la conexión cifrada.

En la UNAM se pueden obtener los certificados de seguridad SSL a través del área de Firma Electrónica Avanzada[7] (FEA), un servicio que se ofrece a través de la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC).

[5] https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API
[6] https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API
[7] https://www.fea.unam.mx/SSL/

7. Actualizar la plataforma de publicación

Para contemplar mejoras de seguridad, así como de rendimiento y desempeño en las aplicaciones web, es importante actualizar las versiones del software del servidor y de los gestores de contenido utilizados.

Infraestructura del servidor

Algunos de los administradores de contenido más populares, así como un gran número de *Frameworks* para crear aplicaciones (Laravel, Cake PHP o Symfony), trabajan sobre un servidor web configurado con PHP y MySQL. Esto se conoce como LAMP, acrónimo de Linux, Apache, MySQL/MariaDB y PHP.

Es necesario actualizar las versiones de este software para mejorar su compatibilidad, tener acceso a parches que remedien problemas de seguridad y aprovechar mejor los recursos de hardware disponibles.

Gestor de contenidos

Actualizar el gestor de contenidos utilizado en un sitio web es de vital importancia para obtener nuevas funcionalidades, mejorar la usabilidad y experiencia del usuario, optimizar el rendimiento y atender riesgos en materia de seguridad de la información.

En el proceso de actualización se deben tomar en cuenta los siguientes puntos:

- Comprobar la compatibilidad de los elementos a actualizar: plugins, módulos, plantillas, temas gráficos, entre otros.
- Activar el “modo de mantenimiento” para informar a los usuarios que el sitio se encuentra en una fase de actualización.
- Crear copias de seguridad. Es necesario contar con un respaldo actualizado del sitio para recuperar la información, en caso de alguna falla.

8. Restringir el acceso a archivos y directorios

Un aspecto básico de seguridad es proteger el listado de archivos contenidos en los directorios públicos de un servidor web, ya que podrían contener información importante respecto al servidor: configuraciones o versiones de software utilizadas, por ejemplo. Si esto es visible, existe la posibilidad de que un tercero aproveche alguna vulnerabilidad específica.

Existen diferentes mecanismos[8] para evitar que un servidor web muestre el listado de los archivos contenidos en un directorio. Éstos dependerán del servidor que se utiliza y de los conocimientos del administrador del sitio.

9. Implementar un firewall de aplicaciones web

Un WAF[9] (*Web Application Firewall*) protege a las aplicaciones web de posibles ataques de XSS (*Cross - Site - Scripting*), SQL *Injection*, envenenamiento de *cookies* y algunos otros, desde la capa de la aplicación. Su adopción es vital para evitar que terceros tengan acceso a los datos sin autorización.

Existen diferentes modos de implementar[10] un WAF, desde soluciones locales de hardware o virtuales hasta soluciones en la nube administradas como servicio.

Aunque es importante contemplar el uso de este tipo de soluciones para un sitio web, no en todos los casos resulta necesario. Su uso dependerá de la cantidad de recursos con los que se cuente y de la naturaleza de la información que se maneje.

---

[8] https://www.visibilidadweb.unam.mx/sites/default/files/recursos/2020-06/SVW_%20RestriccionDeAccesoDir.pdf

[9] Un Web Application Firewall (WAF) protege de múltiples ataques al servidor de aplicaciones web en el backend. La función del WAF es garantizar la seguridad del servidor web mediante el análisis de paquetes de petición HTTP / HTTPS y modelos de tráfico.

¿Cuáles son las ventajas de un WAF?,
https://www.oracle.com/es/database/security/que-es-un-waf.html

[10] Firewall de aplicación Web - Parte I
https://revista.seguridad.unam.mx/numero-16/firewall-de-aplicaci%C3%B3n-web-parte-i

REFERENCIAS

[1] reCAPTCHA |. (s. f.). Google Developers. Recuperado 12 de septiembre de 2022, de https://developers.google.com/recaptcha

[2] HTTP cookies - HTTP | MDN. (2022, 14 Agosto). Recuperado 12 de septiembre de 2022, de https://developer.mozilla.org/es/docs/Web/HTTP/Cookies

[3] Arredondo, M. (2020, mayo). *Importancia de las restricciones de acceso a directorios*. Visibilidad Web - UNAM. Recuperado 12 de septiembre de 2022, de https://www.visibilidadweb.unam.mx/sites/default/files/recursos/2020-06/SVW_%20RestriccionDeAccesoDir.pdf

[4] Sayonara Sarahí Díaz Méndez. (2018). *FIREWALL DE APLICACIÓN WEB - PARTE I*. Revista .Seguridad. https://revista.seguridad.unam.mx/numero-16/firewall-de-aplicaci%C3%B3n-web-parte-i