VSAQ

Cuestionarios de evaluación de seguridad de proveedores

Cargar respuestas desde archivo:
Elegir archivos Ninguno archivo selec.

Cuestionario de seguridad de aplicaciones web

Aplicación

Metadatos de la aplicación

El nombre de la aplicación:

Virtual assistant

Una breve descripción:

La aplicación web se compone de un asistente virtual desplegado sobre una web estática que permite la recogida de las opiniones de los alumnos de un curso de una manera interactiva, rápida y más detallada. Este proceso del asistente forma parte de un sistema mayor donde:
1. Los alumnos acuden al centro y realizan el curso.
2. Tras la finalización del curso, los profesor dan un cuestionario a los estudiantes para que evalúen el curso, su grado de satisfacción, al instructor y el centro. Los profesores les proporcionan un código QR con un enlace y un código para que los alumnos puedan acceder a la web del asistente.
3. En función de las respuestas de los alumnos, los profesores clasifican a estos alumnos (de forma anónima, a través del código proporcionado) en un tipo de alumno (tipo 0, 1, 2 y 3). Estos tipos representan el camino del flujograma que debe seguir el asistente virtual cuando el alumno acceda a conversar.
4. Los alumnos acceden a la web y conversan con el asistente. El asistente únicamente se inicia si el alumno ha introducido su código y es correcto.
5. El asistente detecta el tipo del alumno que es y lanza un flujograma específico para extraer una opinión más detallada.

¿Qué marcos (si los hay) requiere esta aplicación?

Para el backend se utiliza Node.js (JavaScript) y microservicio de Python (mediante el framework Flask). Como base de datc

Informes y gestión de vulnerabilidades

Debido a que ningún sistema está completamente libre de problemas de seguridad, es importante proporcionar formas para que los usuarios externos ofrezcan información y reporten vulnerabilidades.

¿Tiene una forma fácil de descubrir para que los investigadores externos informen las vulnerabilidades de seguridad en sus sistemas?

○ Sí, tenemos un contacto de correo electrónico de seguridad publicado o proporcionamos otra forma para que los usuarios informen problemas de seguridad. Los informes entrantes se revisan y clasifican oportunamente.
◉ No, actualmente no ofrecemos una forma de informar vulnerabilidades de seguridad para el manejo prioritario.

Advertencia: posible problema de riesgo medio

Facilite que otros le informen sobre problemas de seguridad en sus productos. De esa manera, aprenderá antes sobre las vulnerabilidades y podrá responder a ellas rápidamente. Además, sin una manera fácil de informarle directamente sobre los problemas, los investigadores externos podrían publicar los problemas ampliamente.

Si tiene controles de compensación establecidos o cree que este problema no constituye un riesgo en sus circunstancias específicas, explique a continuación. Si está trabajando para abordar este problema, incluya una estimación de cuándo se

resolverá:

Riesgos de HTTPS y contenido mixto

Seleccione la opción que mejor describa su aplicación web:

- (●) La aplicación web es accesible exclusivamente a través de HTTPS. Incluso si el usuario edita manualmente la URL para comenzar con `http://`, no funcionará o se redirigirá a `https://`.
- ( ) La aplicación web es flexible: los usuarios pueden acceder a ella a través de HTTP o HTTPS.
- ( ) La aplicación web solo admite HTTP y no se puede acceder a través de HTTPS, incluso si edita la URL.

Configuración de SSL/TLS

¿Revisó recientemente su configuración SSL para asegurarse de que solo se ofrezcan cifrados y protocolos seguros a los clientes?

- (●) Sí, revisamos periódicamente el paquete de cifrado anunciado por el servidor y los protocolos que utiliza.
- ( ) No estamos seguros de si nuestra configuración SSL/TLS es segura.

¿Su servidor ofrece confidencialidad directa para los clientes que lo admiten?

- (●) Sí, el servidor admite cifrados ECDHE y DHE que ofrecen confidencialidad directa.
- ( ) No, no se habilitan cifrados que proporcionen confidencialidad directa.

¿Están sus claves privadas SSL/TLS debidamente protegidas en sus servidores web?

- (●) Sí, hemos tomado todas las medidas necesarias para proteger nuestras claves privadas.
- ( ) No estoy seguro de lo bien protegidos que están.

¿Dónde finaliza la conexión SSL entre el usuario y su aplicación?

- (●) En el servidor de aplicaciones
- ( ) En el balanceador de carga
- ( ) En algún otro lugar

Las aplicaciones servidas a través de SSL aún pueden ser vulnerables a los ataques si los recursos (a menudo JavaScript, hojas de estilo u otro contenido activo) se incluyen a través de HTTP simple. Esto anula el propósito de SSL, porque el contenido activo cargado a través de HTTP simple tendrá acceso al DOM del contenido protegido por SSL. Asegúrese de que no se incluyan recursos de sitios HTTP simples. Por lo general, los navegadores ayudarán a identificar los casos en los que se incluyen recursos de sitios que no son SSL, mostrando advertencias de contenido mixto .

Para evitar estos problemas, ¿tiene comprobaciones para garantizar que todas las referencias a los recursos apunten a SSL o sean relativas al protocolo?

- (●) Sí, somos muy cuidadosos y contamos con controles específicos para evitar problemas de contenido mixto.
- ( ) No sería muy difícil que algo pasara desapercibido e introdujera errores de contenido mixto.

Para mejorar aún más la seguridad de sus usuarios, ¿ha implementado HTTP Strict Transport Security (HSTS) en su servidor?

- ( ) Sí, tenemos HSTS configurado con un valor de edad máxima de al menos 6 meses.
- (●) No, no usamos HTTP Strict Transport Security, o tenemos configurado un período máximo corto.

Advertencia: posible problema de alto riesgo

Strict Transport Security es un encabezado de respuesta HTTP que les dice a los clientes que inicien conexiones solo a través de HTTPS. Este mecanismo de defensa en profundidad ayuda a mitigar ciertos problemas, como la inclusión accidental de (o la vinculación a) recursos a través de HTTP simple. En HSTS, el encabezado debe configurarse con un valor de 'edad máxima'; esto les dice a los clientes cuánto tiempo deben respetar la directiva.

Las aplicaciones web confidenciales deben habilitar HSTS para garantizar que los usuarios estén siempre protegidos por SSL/TLS. Para que esta protección sea efectiva, el valor de edad máxima debe estar en el rango de más de 6 meses.

Si su aplicación admite la autenticación, ¿las cookies de autenticación están marcadas con el `secure`atributo?

- ○ Sí, las cookies de autenticación están marcadas `secure`.
- ◉ No estoy seguro de si la aplicación hace eso.

Advertencia: posible problema de riesgo medio

A menos que una cookie tenga el `secure`atributo establecido, el navegador agregará la cookie a las solicitudes HTTP de texto sin formato, incluso si la aplicación es solo SSL. Un ataque man-in-the-middle puede usar esta vulnerabilidad para robar cookies de autenticación.

Si tiene controles de compensación establecidos o cree que este problema no constituye un riesgo en sus circunstancias específicas, explique a continuación.

Autenticación y Autorización

Información básica

Para comenzar, cuéntenos un poco acerca de su aplicación para que podamos hacerle las preguntas correctas.

- ☐ Nuestra aplicación requiere que los usuarios regulares inicien sesión. La mayoría de las funciones no están disponibles sin iniciar sesión.
- ☐ Además de una interfaz para usuarios habituales, nuestra aplicación proporciona una interfaz de administración.
- ☐ Nuestra aplicación presenta una gestión de usuarios compleja. Se pueden asignar varios roles a las cuentas de usuario.

Vulnerabilidades web comunes

Ciertas funciones pueden generar problemas de seguridad si se usan incorrectamente. Para ayudarnos a identificar problemas potenciales, seleccione las declaraciones que describen su aplicación:

- ☑ La aplicación utiliza un back-end de base de datos o cualquier otro back-end de persistencia que se pueda consultar con SQL o un lenguaje relacionado (p. ej., GQL, FQL, SOQL, etc.).
- ☐ La aplicación requiere un complemento, como Java, Flash, Silverlight, etc.
- ☐ La aplicación tiene una función de carga de archivos.
- ☐ La aplicación carga contenido activo, como secuencias de comandos, subprogramas u hojas de estilo, desde servidores de terceros (es decir, cualquier servidor que no esté bajo su control directo).
- ☐ La aplicación procesa o manipula el XML proporcionado por el usuario.
- ☐ La aplicación utiliza criptografía para cifrar datos o proteger su integridad.

Secuencias de comandos entre sitios

Las secuencias de comandos entre sitios (o XSS para abreviar) se producen cuando una aplicación vuelve a mostrar una entrada de usuario insuficientemente desinfectada en el contexto del origen de la aplicación

(como se define en la política del mismo origen). Si la entrada del usuario contiene ciertos tipos de código de secuencias de comandos, puede leer o alterar el DOM de la página actual cuando se vuelve a mostrar. En muchos casos, XSS se usa para robar las cookies de los usuarios u otros datos relacionados con la aplicación, pero también se puede usar para ataques de phishing o incluso para desfigurar la página web. Desafortunadamente, XSS es uno de los problemas de seguridad más comunes en las aplicaciones web y, debido a las peculiaridades del navegador y otros factores, es bastante difícil protegerse. Seleccione las declaraciones que describen su estrategia:

- ☐ Utilizamos un sistema de plantillas que escapa automáticamente a todas las entradas del usuario antes de volver a mostrarlo.
- ☐ Nuestra aplicación tiene un cuello de botella central donde todas las entradas del usuario se validan y escapan, según el contexto en el que se interpretarán.
- ☐ Algunas de las páginas (o todas ellas) escapan a la entrada del usuario.
- ☐ Estamos utilizando alguna otra técnica para proteger contra XSS.
- ☑ Parte de la aplicación trata con HTML proporcionado por el usuario que se desinfecta y se vuelve a mostrar al usuario.

Advertencia: posible problema de riesgo medio

Desafortunadamente, debido al comportamiento de corrección de errores y las peculiaridades del navegador, es muy difícil obtener el saneamiento de HTML correcto. El código de desinfección tiene que crear una representación en memoria del DOM y luego serializarla en un formato seguro conocido. Algunas bibliotecas hacen esto correctamente, por lo que recomendamos usar una biblioteca bien probada.

Describa cómo su aplicación trata con el saneamiento de HTML:

Además de aplicar las estrategias que ha identificado, ¿establece la aplicación un tipo de contenido y un conjunto de caracteres válidos y apropiados para cada página (en el `Content-Type`encabezado HTTP)?

- ◉ Sí, tenemos mucho cuidado al establecer esto, sabiendo que de lo contrario podríamos estar introduciendo vulnerabilidades XSS.
- ○ No estoy seguro de que todas las páginas establezcan un tipo de contenido adecuado.

Algunas vulnerabilidades XSS funcionan exclusivamente en el lado del cliente, en el código de secuencias de comandos de una aplicación. Este tipo de XSS se conoce comúnmente como XSS basado en DOM . Debido a que el escape del lado del servidor de la entrada del usuario no protege contra XSS basado en DOM, necesita una estrategia para manejar el código de secuencias de comandos del lado del cliente que maneja la entrada del usuario, así como partes del DOM que pueden contener la entrada del usuario (como documento .localización).

- ○ Conocemos XSS basado en DOM y tomamos medidas específicas para protegernos contra este tipo de vulnerabilidad.
- ◉ Es posible que algo se haya escapado y nuestra aplicación tenga vulnerabilidades XSS basadas en DOM.

Advertencia: posible problema de riesgo medio

Recomendamos que audite minuciosamente su código en busca de vulnerabilidades XSS basadas en DOM y establezca procedimientos para que el código futuro también esté protegido.

Si tiene controles de compensación establecidos o cree que este problema no constituye un riesgo en sus circunstancias específicas, explique a continuación. Si está trabajando para abordar este problema, incluya una estimación de cuándo se resolverá:

Back-ends de persistencia y consultas

Debido a que su aplicación utiliza una base de datos o un back-end similar para conservar los datos, debemos asegurarnos de que no sea vulnerable a los ataques de inyección, como la inyección SQL .

Una aplicación es vulnerable a la inyección SQL cuando la base de datos interpreta una parte de la entrada del usuario como parte de una consulta. Cuando esto ocurre, un atacante puede leer o incluso escribir datos directamente desde o hacia la base de datos.

- [ ] Nuestra aplicación utiliza un marco de mapeo relacional de objetos (ORM). Cuando necesitamos construir manualmente consultas o condiciones, usamos uno de los mecanismos seleccionados a continuación:
- [x] Usamos declaraciones preparadas y dejamos que el marco se encargue de escapar correctamente de la entrada del usuario.
- [ ] Pasamos la entrada del usuario a la base de datos a través de procedimientos almacenados.
- [x] Escapamos manualmente de la entrada del usuario cada vez que necesitamos usarla en una consulta de base de datos.
- [ ] Hacemos otra cosa.

Advertencia: posible problema de riesgo medio

El escape manual de consultas SQL (o relacionadas) es propenso a errores y muy difícil de hacer de manera consistente. Aquí hay un par de ejemplos:

- PHP proporciona la función `mysql_escape_string`para escapar de la entrada del usuario que se utilizará en una consulta de base de datos. Desafortunadamente, esa función no tiene en cuenta el conjunto de caracteres de la conexión, por lo que aún es posible pasar de contrabando la entrada del usuario que se interpretará como parte de la consulta SQL. Las aplicaciones deberían usar en su `mysql_real_escape_string`lugar.
- En SQL, los números no necesitan estar entre comillas cuando se usan en una declaración. Por ejemplo, `SELECT username WHERE id=123`es perfectamente válido. Pero si no se confirma que la entrada proporcionada por el usuario que se usará como un número sea realmente numérica, el código resultante será vulnerable a la inyección SQL (incluso si la entrada tiene escape).

Recomendamos encarecidamente usar algo como declaraciones preparadas o usar una capa ORM de manera consistente en toda la aplicación. Asegúrese de tener procedimientos establecidos para hacer cumplir su enfoque (como pruebas estáticas cuando el código se registra en el repositorio).

Si tiene controles de compensación establecidos o cree que este problema no constituye un riesgo en sus circunstancias específicas, explique a continuación. Si está trabajando para abordar este problema, incluya una estimación de cuándo se resolverá:

Pruebas, QA y Monitoreo

Las pruebas de seguridad pueden ser parte de las pruebas de aplicación estándar. Aquí hay unos ejemplos:

- **Pruebas unitarias simples: las pruebas unitarias se utilizan normalmente para confirmar que los componentes básicos de la aplicación funcionan como se esperaba. Las pruebas unitarias son fáciles de repetir: se pueden ejecutar cada vez que se registra un nuevo código en el repositorio, para confirmar que el código aún se comporta como se esperaba. Las pruebas unitarias también pueden verificar las características de seguridad. Por ejemplo, se pueden usar para confirmar que las solicitudes fallan sin tokens XSRF; que se requiere autenticación para acceder a los datos del usuario; o que las etiquetas HTML inesperadas no pueden atravesar los filtros de entrada o escapar de las rutinas.**

- **Pruebas de lanzamiento: antes de que se lance una nueva versión de un producto, los evaluadores humanos generalmente revisan la aplicación, prueban las nuevas funciones y se aseguran de que las funciones anteriores aún funcionen correctamente (pruebas de regresión). Las pruebas de seguridad también deben incluirse en este proceso. Por ejemplo, las pruebas de lanzamiento son un buen momento para verificar que el usuario A no puede acceder a los datos del usuario B.**
  - **Monitoreo: una vez que se implementa la aplicación, el enfoque generalmente cambia de la prueba al monitoreo. Tenga cuidado con los picos inesperados en las tasas de error, las infracciones de sandbox y otros comportamientos irregulares o inexplicables (incluidos los errores de prueba intermitentes), y antes de descartar una anomalía, consulte con su equipo de seguridad. Los bloqueos y la descamación pueden indicar una condición de carrera o un error de corrupción de memoria.**

Las siguientes preguntas evalúan las pruebas y el seguimiento de su aplicación.

¿Está utilizando pruebas unitarias o métodos similares?

- ○ Sí
- ◉ No

¿Sus ingenieros y su equipo de control de calidad buscan posibles problemas de seguridad durante las pruebas de lanzamiento y han sido capacitados para hacerlo?

- ○ Sí, nuestro proceso de control de calidad incluye explícitamente pruebas de problemas de seguridad que podrían haberse introducido en la nueva versión.
- ◉ Esta es un área en la que tenemos margen de mejora.

Advertencia: posible problema de riesgo medio

Sus equipos de ingeniería y control de calidad están en la mejor posición para comprender cómo funcionan todas las partes de la aplicación, qué ha cambiado desde la iteración anterior y cómo los cambios pueden introducir vulnerabilidades de seguridad. Las pruebas de lanzamiento generalmente se realizan bajo una presión de tiempo considerable, pero es la última oportunidad de detectar vulnerabilidades de seguridad internamente. Cuando sus equipos ya están enfocados en las pruebas, agregar algunas pruebas de seguridad no aumentará mucho el esfuerzo.

Los ingenieros y probadores que han sido capacitados para buscar problemas de seguridad pueden marcar la diferencia entre un producto seguro y una vulnerabilidad grave.

Si tiene controles de compensación establecidos o cree que este problema no constituye un riesgo en sus circunstancias específicas, explique a continuación. Si está trabajando para abordar este problema, incluya una estimación de cuándo se resolverá:

Monitoreo Post-Lanzamiento

¿Cómo describiría su seguimiento posterior al lanzamiento?

- ○ **Sólido** : contamos con procedimientos para registrar y monitorear bloqueos inesperados, excepciones y otras condiciones de error. Si algo parece sospechoso, un ingeniero preocupado por la seguridad lo evalúa.
- ○ **Débil** : si algo sale terriblemente mal, como picos masivos en las tasas de accidentes u otras anomalías a gran escala, probablemente nos demos cuenta. Pero nuestro monitoreo es bastante tosco y hay espacio para mejorar.
- ◉ **Inexistente** : por el momento, no estamos realizando ningún tipo de seguimiento posterior al lanzamiento que busque signos de explotación o aumentos en bloqueos/excepciones.

Advertencia: posible problema de riesgo medio

Las excepciones y los bloqueos a menudo indican un problema de seguridad subyacente. El monitoreo de la aplicación implementada puede contribuir en gran medida a identificar rápidamente y corregir posteriormente las vulnerabilidades.

En productos de software cuidadosamente diseñados, las excepciones deberían ser una ocurrencia bastante rara; por lo tanto, generalmente no introduce una sobrecarga significativa para monitorearlos.

Si tiene controles de compensación establecidos o cree que este problema no constituye un riesgo en sus circunstancias específicas, explique a continuación. Si está trabajando para abordar este problema, incluya una estimación de cuándo se resolverá:

Notas adicionales

Proporcione cualquier información adicional sobre la seguridad de su aplicación:

No se realizan Tests en la aplicación.
No se utilizan herramientas de CI/CD.

Contactos de seguridad

Enumere las direcciones de correo electrónico de las personas a las que debemos contactar sobre cualquier problema de seguridad en la aplicación:

Comentarios

¡Felicidades! Has llegado al final de este cuestionario. Si tiene otro minuto, háganos saber cómo podemos mejorarlo. Su retroalimentación es muy apreciada.

Estado:Borrador guardado

Descargar respuestas **Restablecer cuestionario**