Actividad 2.4 Análisis de Riesgos de Seguridad - Reto

Integración de seguridad informática en redes y sistemas de software

Grupo 402

José Antonio González Martínez - A01028517
Luis Daniel Filorio - A01028418
Gabriel Muñoz Luna - A01028774
Rodrigo Sosa Rojas - A01027913

Bajo la instrucción de

Carlos Enrique Vega Álvarez
Lizbeth Peralta Malváez
Osvaldo Cecilia Martínez
Edith Carolina Arias Serna

Fecha de entrega

06/10/2024

Índice

1. Objetivos

El objetivo de este documento es realizar un análisis profundo de los posibles riesgos que podemos afrontar dentro de nuestro sistema, realizando los análisis correspondientes para poder identificar los impactos que podrían tener dichos riesgos. De igual manera, encontrar las mejores opciones para el tratado de los mismos siendo estos la mitigación del riesgo, el transferirlo y/o evitarlo, esto con el fin de poder tener un sistema mucho más seguro en el cuál protejamos los datos de todos nuestros usuarios.

2. Descripción de los componentes de la solución de software y la interacción entre ellos

Componentes del sistema:

Frontend (React Admin):

> Página de Inicio de Sesión (Login)
> Página de Donaciones
> Dashboard de Recursos
> Dashboard de Usuarios (en desarrollo)

Backend (MongoDB):

> API para gestionar las donaciones y usuarios
> Manejo de autenticación
> Envío de correos (Nodemailer)
> Base de Datos (MongoDB):

Almacena información de usuarios y donaciones
Contraseñas almacenadas de forma segura mediante un hash

Seguridad:

> HTTPS para asegurar la transmisión de datos
> Autenticación y roles de usuario
> Prevención de ataques comunes

3. Identificación de componentes críticos de la solución de software

Dentro de nuestro sistema, nos encontramos con diferentes componentes entre los cuales los más importantes son:

- Inicio de sesión
- Manejo de datos personales
- Guardado de contraseñas
- Manejo de donaciones con datos del donador

En estos componentes es donde se manejan la mayor cantidad de datos sensibles de nuestros usuarios, por lo que se debe de tener un gran cuidado con el manejo de los mismos, así como protegerlos en caso de cualquier eventualidad.

4. Matriz CIA para la identificación de amenazas

Matriz CIA

Activo	C	I	A
Login	Robo de credenciales de usuario a través de ataque de fuerza bruta o phishing	Modificación de los datos introducidos (por ejemplo, monto de las donaciones) mediante XSS	Ataques DDoS que impidan el acceso al sitio
API	Exfiltración de datos a través de vulnerabilidades en la API	Inyección de comandos que alteren los datos de usuarios y donaciones	Ataques DDoS que afectan la disponibilidad del backend
MongoDB	Robo de información confidencial por acceso no autorizado	Modificación de datos almacenados	Bloqueo o corrupción de la base de datos por ransomware
Certificados	Robo de tokens JWT para acceder a áreas restringidas	Alteración de tokens para modificar permisos	Bloqueo del sistema de autenticación,

		o identidades de usuario	impidiendo que los usuarios inicien sesión
	Interceptación de datos durante la transmisión (si HTTPS no está bien configurado)		Desconfiguración de los certificados que impida la conexión segura

5. Descripción de las amenazas

1. Robo de Credenciales (Login y Donaciones)

Descripción: Un atacante podría intentar obtener credenciales de usuario mediante ataques de fuerza bruta, phishing o interceptación de datos, comprometiendo la confidencialidad de los usuarios.
Impacto: Si las credenciales de un administrador son robadas, podría permitir acceso a datos sensibles o funcionalidades críticas del sistema.
Probabilidad: Media, es la forma común de obtener información hoy en dia

2. Modificación de Datos de Usuarios o Donaciones

Descripción: Un atacante puede utilizar vulnerabilidades de Cross-Site Scripting para alterar los datos introducidos en el sistema
Impacto: Afecta la integridad de los datos, causando datos que no concuerden o sean falsos lo que podría impactar negativamente la operación y la confianza en la asociación.
Probabilidad: Media, ya que las aplicaciones web son frecuentemente objetivo de ataques de este tipo si las entradas no son correctamente validadas.

3. Ataques DDoS

Descripción: Un ataque distribuido de denegación de servicio (DDoS) puede sobrecargar los recursos del sistema, impidiendo que los usuarios accedan al frontend, backend o base de datos.
Impacto: La disponibilidad del sistema deja de funcionar, lo que podría afectar la operatividad del sitio web y generar pérdidas de usuarios y donaciones.
Probabilidad: Baja, pero puede pasar, especialmente si el sistema no tiene medidas de mitigación de DDoS.

4. Exfiltración de Datos a través de la API

Descripción: Los atacantes podrían explotar vulnerabilidades en la API para acceder a datos confidenciales, como información de usuarios y donaciones.
Impacto: Un acceso no autorizado podría comprometer datos personales y financieros, exponiendo a la organización a pérdidas económicas y problemas legales.

Probabilidad: Media, porque las API suelen ser atacadas frecuentes si no están bien protegidas con autenticación y autorización adecuada.

5. **Inyección NoSQL**

Descripción: Un atacante podría aprovechar vulnerabilidades de inyección NoSQL ya que la base de datos está en mongo para alterar datos almacenados, como montos de donaciones, roles de usuario o credenciales.
Impacto: La integridad de los datos almacenados en la base de datos se ve comprometida, lo que podría afectar operaciones críticas del sistema.
Probabilidad: Media, si no se implementan validaciones robustas en el backend.

6. **Ransomware**

Descripción: Un ataque de ransomware podría cifrar la base de datos, bloqueando el acceso a los datos de usuarios y donaciones hasta que se pague un rescate.
Impacto: La disponibilidad de la base de datos queda completamente comprometida, y la organización podría enfrentarse a la pérdida de datos críticos si no se paga el rescate.
Probabilidad: Baja, pero puede pasar si no se implementan medidas de seguridad robustas, como copias de seguridad encriptadas.

7. **Robo de Tokens**

Descripción: Un atacante podría interceptar o robar tokens, lo que le permitiría suplantar a un usuario legítimo y acceder a recursos o información.
Impacto: La confidencialidad y la integridad del sistema se ven comprometidas, ya que un atacante podría realizar acciones no autorizadas.
Probabilidad: Media, especialmente si los tokens no se protegen adecuadamente con medidas como expiración corta y almacenamiento seguro.

8. **Interceptación de Datos en Tránsito (HTTPS)**

Descripción: Si el HTTPS no está bien configurado o no se renueva los certificados, los datos en tránsito podrían ser interceptados, exponiendo información confidencial como credenciales y datos personales.
Impacto: Compromete la confidencialidad de la información transmitida entre el frontend y el backend.
Probabilidad: Baja, si se configuran adecuadamente HTTPS y los certificados de seguridad.

Amenaza	Probabilidad	Impacto
Robo de Credenciales	Medio	Alto
Modificación de Datos	Medio	Medio
Ataques DDo	Bajo	Alto
Exfiltración de Datos a través de la API	Medio	Alto
Inyección NoSQL	Medio	Alto
Ransomware	Bajo	Alto
Robo de Tokens	Medio	Medio
Interceptación de Datos en Tránsito	Bajo	Medio

6. Matriz de riesgos (heat map) para la estimación del nivel de riesgo

	Impacto			
Probabilidad		**Alto**	**Medio**	**Bajo**
	Alto			
	Medio	Robo de Credenciales, Exfiltración de Datos a través de la API Inyección NoSQL	Modificación de Datos Robo de Tokens	
	Bajo	Ataques DDo Ransomware	Interceptación de Datos en Tránsito	

7. Priorización de riesgos

1. Exfiltración de Datos a través de la API (Backend)
2. Robo de Credenciales
3. Inyección NoSQL
4. Robo de Tokens
5. Modificación de Datos
6. Ransomware
7. Ataques DDoS
8. Interceptación de Datos en Tránsito

Selección de los Riesgos Más Críticos

- Exfiltración de Datos a través de la API
- Robo de Credenciales
- Inyección NoSQL

8. Definición de tratamiento para los riesgos (evitar, transferir, mitigar, aceptar)

1. Exfiltración de Datos a través de la API

Tratamiento: Mitigar

Medidas de Mitigación:

- Implementar controles de acceso estrictos, asegurando que solo usuarios autorizados accedan a la API.
- Monitoreo constante de tráfico y actividades sospechosas en la API (implementación de herramientas de monitoreo).

2. Robo de Credenciales

Tratamiento: Mitigar

Medidas de Mitigación:

- Implementar autenticación multifactor (MFA) para mejorar la seguridad en el inicio de sesión.
- Limitar los intentos de inicio de sesión y aplicar medidas de bloqueo temporal ante intentos repetidos fallidos.
- Asegurar que las contraseñas estén correctamente hasheadas.
- Hacer uso de HTTPS en todas las comunicaciones entre el cliente y el servidor para evitar la interceptación de credenciales.

3. Inyección NoSQL

Tratamiento: Mitigar

Medidas de Mitigación:

- Usar operadores de consulta seguros para MongoDB en general, evitando la construcción dinámica de consultas con entradas no válidas.
- Implementar controles de acceso y monitoreo de la base de datos, con auditorías de cambios sospechosos.

4. Ransomware

Tratamiento: Mitigar

Medidas de Mitigación:

- Realizar copias de seguridad (backups) periódicas y almacenarlas en ubicaciones separadas de la infraestructura principal.
- Asegurar que las copias de seguridad estén encriptadas y sólo accesibles a personal autorizado.
- Implementar medidas de detección de malware y ransomware en el sistema, incluyendo alertas de situaciones extrañas.

5. Ataques DDoS

Tratamiento: Mitigar

Medidas de Mitigación:

- Usar un servicio de mitigación de DDoS en la infraestructura

6. Robo de Tokens

Tratamiento: Mitigar

Medidas de Mitigación:

- Implementar expiración corta para los tokens

7. Modificación de Datos

Tratamiento: Mitigar

Medidas de Mitigación:

- Validar y codificar correctamente la salida de datos para evitar que scripts maliciosos se ejecuten en los navegadores de los usuarios.

8. Interceptación de Datos en Tránsito (HTTPS)

Tratamiento: Mitigar

Medidas de Mitigación:

- Asegurarse de que HTTPS esté correctamente configurado.
- Renovar los certificados periódicamente y asegurarse de que estén configurados correctamente.

9. Selección de medidas de mitigación por cada riesgo y describir consideraciones para su implementación. Considerar al menos los siguientes controles:

- **Seguridad de datos en tránsito: https para backend y frontend**
  - Para el manejo de datos por la red, contamos con un certificado https que nos permite manejar los datos sensibles como las contraseñas de manera segura y que nos permite tener un tránsito de datos cifrados para poder proteger los contenidos de los paquetes que transitan.

- **Seguridad de datos en reposo: hasheo de secretos**

  Dentro de nuestra base de datos, contamos con un cifrado de contraseñas que resguardan las contraseñas de los usuarios de manera segura para evitar que puedan verse a simple vista.

- **Controles de autenticación: login y generación de tokens JWT**

  En el momento en el que un usuario inicia sesión para acceder a la página, se creará un token JWT a partir del usuario y el rol que tenga este que es firmado con una llave secreta. Este token dará acceso a donde tenga autorización el usuario.

- **Controles de autorización: roles de usuario y validación de de tokens JWT**

  En nuestro inicio de sesión y en general en el tránsito de usuarios, contamos con un sistema JWT que nos proporciona un token a partir del usuario y el rol que tiene el usuario para que así sólo los usuarios autorizados puedan acceder y poder realizar acciones a donde se les tenga permitido acceder, esto previniendo que usuarios comunes puedan a ver o acceder a componentes de administradores.

- **Prevención de vulnerabilidades comunes (XSS y SQL/NOSQL injection)**

  Para evitar que se sufra una inyección de NoSql contamos con el inicio de sesión por roles lo que nos permite dividir qué personas tienen acceso a la base de datos y quién no, esto evitando que a partir de cualquier cuenta se pueda hacer un ataque de este tipo.

  De igual manera, contamos con los certificados Https para poder evitar que se tenga acceso a los datos que navegan a través de la red y evitar un ataque XSS, esto a través del cifrado de los datos tanto enviados como recibidos.

10. Conclusiones

Los sistemas de páginas Web en combinación con Bases de Datos que contienen información sensible son sin lugar a dudas de los más vulnerables. Si no están bien protegidos la página Web, servidor, bus de datos, entre otros componentes, información como correos, nombres e incluso datos bancarios podrían estar afectados y expuestos a un ataque.

Nuestro sistema cubre una buena cantidad de estas vulnerabilidades a través de certificados y cifrado de datos. Fuimos capaces de mitigar las posibilidades de ataques con estos, manteniendo la información de los usuarios segura. Sin embargo, es importante mencionar que, como cualquier otro sistema, las vulnerabilidades podrían ser explotadas, por lo que un constante monitoreo y evolución de la integridad de las medidas de seguridad sería ideal para mantener la seguridad del sistema.