# Análisis de Amenazas para la Empresa de Zapatillas Online

**Etapas:**

**1. Definir los Objetivos Empresariales y de Seguridad**

- Cumplimiento PCI-DSS: Dado que la aplicación manejará transacciones financieras, es fundamental que cumpla con los estándares de PCI-DSS (Payment Card Industry Data Security Standard) para garantizar la seguridad de los datos de tarjetas de crédito.
- Autenticación y Autorización Seguras: Implemente autenticación multifactor (MFA) para reforzar la seguridad de los usuarios, especialmente durante los procesos de inicio de sesión y transacciones.
- Cifrado de Datos: Asegúrese de que todos los datos sensibles, como las credenciales y la información financiera, estén cifrados tanto en tránsito (utilizando TLS) como en reposo (utilizando AES-256).
- Control de Accesos: Aplique el principio de privilegio mínimo para asegurarse de que solo los usuarios autorizados tengan acceso a datos sensibles.

**2. Definir el Alcance Técnico**

- Revisión de APIs: Las APIs son esenciales para la interconexión entre clientes, socios y empleados. Deben ser evaluadas y priorizadas debido a los datos sensibles que manejan. Considere utilizar OAuth 2.0 y JWT (JSON Web Tokens) para autenticar y autorizar las solicitudes a la API.
- API Gateway: Utilice un API Gateway para gestionar el tráfico y proteger las APIs. Aplique medidas como limitación de velocidad (rate limiting) y autenticación para prevenir abusos.
- SHA-256: Asegúrese de que las contraseñas y los datos sensibles estén protegidos mediante SHA-256 y otras técnicas de cifrado robustas para garantizar la seguridad.
- PKI (Infraestructura de Clave Pública): Implemente PKI para asegurar la comunicación entre los usuarios y la aplicación, así como entre los sistemas internos.

**3. Descomponer la Aplicación**

- Diagrama de Flujo de Datos (DFD): Utilice un Diagrama de Flujo de Datos (DFD) detallado para ilustrar cómo circulan los datos dentro de la aplicación y para identificar puntos críticos que podrían ser vulnerables.
- SQL y Sentencias Preparadas: Asegúrese de que las consultas SQL utilicen sentencias preparadas para prevenir inyecciones SQL. Las entradas deben validarse de forma adecuada y evitar inyecciones maliciosas.
- Control de Entrada de Usuario: Implemente validación y saneado de entradas de usuario para prevenir vulnerabilidades como Cross-Site Scripting (XSS) o Cross-Site Request Forgery (CSRF).

**4. Análisis de Amenazas**

- Inyección SQL: Asegúrese de que la base de datos esté protegida contra ataques de inyección SQL, que son comunes en aplicaciones web que interactúan con bases de datos. Utilice sentencias preparadas y ORMs (Object Relational Mappers) para prevenir estas vulnerabilidades.
- Secuestro de Sesión: El secuestro de sesión es posible si las cookies no se manejan de manera segura. Implemente las banderas HTTPOnly y Secure para asegurar que las cookies no puedan ser accedidas por JavaScript ni transmitidas a través de canales inseguros.
- Cross-Site Scripting (XSS): Prevenga ataques XSS validando y escapando adecuadamente las entradas del usuario y utilizando cabeceras de seguridad como Content Security Policy (CSP).
- Ataques DDoS: La aplicación podría ser vulnerable a ataques de Denegación de Servicio Distribuida (DDoS). Utilice servicios de protección DDoS como Cloudflare o AWS Shield.

**5. Análisis de Vulnerabilidades**

- Inyección SQL por falta de Sentencias Preparadas: La falta de uso de sentencias preparadas puede abrir la puerta a inyecciones SQL. Asegúrese de que todas las consultas a la base de datos utilicen sentencias preparadas o procedimientos almacenados.
- Token de API Comprometido: Tokens de API deben ser generados de forma segura y nunca ser almacenados en texto claro. Implemente una política de rotación de tokens de API y use OAuth 2.0 o JWT para autenticar las solicitudes a las APIs.
- Manejo de Sesiones Inseguro: Si las cookies o tokens de sesión no se gestionan adecuadamente, un atacante podría secuestrar la sesión de un usuario. Revise las políticas de manejo de sesiones, implemente expiración automática y revocación de sesiones.

**6. Modelado de Ataques**

- Utilice MITRE ATT&CK y la Lista de CVE® para comprender las técnicas de ataque utilizadas por los ciberdelincuentes y adaptarlas al contexto de la aplicación.
- Implemente un análisis de riesgos de forma periódica para identificar nuevas amenazas, utilizando herramientas como OWASP ZAP y Burp Suite para realizar pruebas de penetración y identificar vulnerabilidades.

**7. Análisis de Riesgos e Impacto**

- SHA-256 para Cifrado Seguro: Aplique SHA-256 para el almacenamiento seguro de contraseñas y otros datos sensibles. Asegúrese de usar salts y hashing iterativo para proteger los datos en reposo.

- Procedimientos de Respuesta ante Incidentes: Implemente procedimientos de respuesta ante incidentes para detectar, contener y remediar rápidamente cualquier brecha de seguridad. Además, asegúrese de que el personal esté capacitado para responder a incidentes de manera efectiva.
- Política de Contraseñas: Establezca una política de contraseñas robustas que exija una longitud mínima y el uso de caracteres especiales, y que prohíba el uso de contraseñas comunes. Implemente un sistema de bloqueo de cuentas tras múltiples intentos fallidos.
- Monitorización de Seguridad: Implemente una solución de monitoreo de seguridad que pueda detectar actividad sospechosa en tiempo real, como intrusión de red o acceso no autorizado. Herramientas como Splunk y Elastic Stack son útiles para esto.

**Resumen de Mejoras y Próximos Pasos:**

1. Refinamiento de los Objetivos de Seguridad: Asegurarse de que los objetivos de seguridad estén claramente alineados con los objetivos comerciales y de negocio, y que se implementen controles adecuados desde el inicio.
2. Revisión Continua de las APIs y Componentes de Terceros: Las APIs deben ser continuamente evaluadas y auditadas para detectar nuevas vulnerabilidades.
3. Pruebas de Penetración Regulares: Realizar pruebas de penetración periódicas para identificar vulnerabilidades emergentes antes de que los atacantes puedan explotarlas.
4. Implementación de Controles de Seguridad: Aplicar controles adicionales como WAFs, API Gateways, y protección DDoS para fortalecer la postura de seguridad de la aplicación.