

**Definición de casos de prueba de seguridad**

### **Pasos para definir los casos de prueba de seguridad:**

#### **1. Identificación de riesgos y amenazas:**

El primer paso es identificar los riesgos específicos para este proyecto, como:

* **Acceso no autorizado a la información del cliente**.
* **Alteración o eliminación no autorizada de los registros de cobranza**.
* **Exfiltración de datos de pagos y métodos de cobro**.
* **Interrupciones en los servicios de cobranza debido a ataques como DoS**.

#### **2. Objetivos de seguridad para el sistema:**

Para el sistema de optimización de cobranza, los objetivos de seguridad incluyen:

* Garantizar la confidencialidad de los datos de los clientes y sus transacciones.
* Asegurar la integridad de los datos de cobranza (que no se alteren sin autorización).
* Implementar controles de acceso sólidos para diferentes roles de usuario (cobradores, administradores, clientes).
* Proteger contra ataques de inyección y denegación de servicio que afecten el funcionamiento del sistema.

#### **3. Definir los casos de prueba de seguridad:**

##### **Caso de Prueba 1: Autenticación y Autorización**

**Objetivo**: Validar que solo usuarios autorizados puedan acceder a diferentes funcionalidades del sistema.

* **Entradas**: Credenciales válidas e inválidas para diferentes roles de usuario (administradores, cobradores, supervisores, etc.).
* **Acciones**: Intentar iniciar sesión con credenciales correctas e incorrectas, y probar el acceso a diferentes módulos según el rol asignado.
* **Resultado esperado**: Los usuarios deben poder acceder únicamente a las funcionalidades permitidas para su rol, y los intentos de acceso no autorizado deben ser bloqueados y registrados en los logs de seguridad.

##### **Caso de Prueba 2: Protección de Datos Personales y de Transacciones (Cifrado)**

**Objetivo**: Asegurar que los datos sensibles de los clientes, como nombres, métodos de pago y detalles de transacciones, estén cifrados tanto en tránsito como en reposo.

* **Entradas**: Datos personales y de pago ingresados en formularios web o mediante APIs.
* **Acciones**: Inspeccionar el tráfico de red y la base de datos para verificar que los datos sensibles se cifren correctamente utilizando SSL/TLS y otros métodos de cifrado adecuados.
* **Resultado esperado**: Los datos deben estar cifrados y no legibles en texto plano durante su transmisión o almacenamiento.

##### **Caso de Prueba 3: Control de Acceso a los Registros de Cobranza**

**Objetivo**: Probar que los usuarios sin permisos no puedan acceder o modificar registros de cobro.

* **Entradas**: Intentos de acceder a registros de cobro como usuarios sin privilegios de edición.
* **Acciones**: Navegar al área de registros de cobranza y tratar de modificar o eliminar datos sin los permisos necesarios.
* **Resultado esperado**: Los usuarios sin permisos no deben poder modificar ni eliminar registros. Todos los intentos no autorizados deben ser registrados.

##### **Caso de Prueba 4: Pruebas de Inyección SQL**

**Objetivo**: Asegurar que el sistema es resistente a ataques de inyección SQL.

* **Entradas**: Cadenas de texto con código SQL malicioso en formularios de búsqueda o autenticación.
* **Acciones**: Intentar introducir código SQL en campos de entrada de datos (por ejemplo, en campos de búsqueda o formularios de autenticación).
* **Resultado esperado**: El sistema debe filtrar y bloquear cualquier intento de inyección SQL, y devolver un mensaje de error sin comprometer la base de datos.

##### **Caso de Prueba 5: Gestión de Sesiones y Protección contra Secuestro de Sesión**

**Objetivo**: Validar que las sesiones de los usuarios se gestionan correctamente, evitando secuestro de sesión.

* **Entradas**: Sesiones de usuario activas, intentos de acceder a sesiones mediante cookies robadas o manipuladas.
* **Acciones**: Iniciar sesión, intentar acceder a la sesión de otro usuario mediante manipulación de cookies, observar el comportamiento de la sesión después de inactividad prolongada.
* **Resultado esperado**: El sistema debe cerrar la sesión automáticamente tras un periodo de inactividad, invalidar las cookies manipuladas, y las sesiones deben ser seguras.

##### **Caso de Prueba 6: Pruebas de Denegación de Servicio (DoS)**

**Objetivo**: Probar la resistencia del sistema frente a ataques de denegación de servicio que intentan sobrecargar el sistema.

* **Entradas**: Alta frecuencia de peticiones simultáneas y carga excesiva en los servidores.
* **Acciones**: Enviar múltiples peticiones simultáneas al servidor para probar la respuesta del sistema bajo condiciones de carga.
* **Resultado esperado**: El sistema debe manejar la sobrecarga sin fallar, limitando el número de peticiones por usuario (mediante throttling o limitación de ancho de banda) y garantizar la continuidad de los servicios esenciales.

##### **Caso de Prueba 7: Registro y Monitoreo de Actividad**

**Objetivo**: Verificar que todas las acciones relacionadas con la seguridad son registradas adecuadamente.

* **Entradas**: Intentos de acceso no autorizado, cambios en los datos de los clientes, acceso a módulos críticos.
* **Acciones**: Ejecutar actividades que puedan ser consideradas riesgosas o no autorizadas dentro del sistema (como intentos fallidos de inicio de sesión o acceso a áreas restringidas).
* **Resultado esperado**: Todas las actividades deben ser registradas en los logs de seguridad, incluyendo detalles de la actividad (fecha, hora, usuario, dirección IP, etc.).

#### **4. Definir el entorno de pruebas:**

El entorno de pruebas debe replicar tanto las condiciones de producción como los posibles escenarios de ataque. Esto incluye:

* **Entorno de prueba con tráfico simulado**.
* **Pruebas desde redes externas y simulación de accesos maliciosos**.
* **Pruebas con diferentes niveles de permisos para usuarios internos y externos**.

#### **5. Revisión y refinamiento de los casos de prueba:**

Los casos de prueba deben ser revisados por el equipo de desarrollo y de seguridad para garantizar que cubren todos los aspectos relevantes y se ajusten a las políticas de seguridad de Alloxentric.

### **Consideraciones finales:**

Cada caso de prueba debe incluir pasos detallados para su ejecución y criterios claros de aceptación. Además, deben realizarse pruebas de penetración para identificar vulnerabilidades no cubiertas por los casos de prueba, y el sistema debe ser auditado regularmente para garantizar su seguridad a lo largo del tiempo.