

BANCO BOLIVARIANO

**Estándar de Seguridad
para Aplicaciones**




Banco Bolivariano

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 2 de 40	

Índice:

1. INTRODUCCIÓN	4
1.1. OBJETIVO	4
1.2. ÁMBITO DE APLICACIÓN	4
1.3. NORMATIVA MARCO (NORMATIVAS SUPERIOR DE REFERENCIA)	4
1.4. NORMATIVA DEROGADA	4
1.5. OTRAS NORMATIVAS ASOCIADAS	4
1.6. VIGENCIA	4
1.7. DISPOSICIONES GENERALES Y TRANSITORIAS	4
1.8. ROLES Y RESPONSABILIDADES	5
2. DE LAS GENERALIDADES	6
2.1. OBJETIVO	6
2.2. SEGURIDAD A NIVEL DE APLICACIONES	6
2.3. DOCUMENTACIÓN	6
3. CONSIDERACIONES BÁSICAS DE SEGURIDAD	7
3.1. OBJETIVO	7
3.2. BUENAS PRÁCTICAS DE CONTRASEÑAS	7
3.3. USO DE PROTOCOLOS SEGUROS	8
3.4. AUTENTICACIÓN	8
3.4.1. AUTENTICACIÓN CONTINUA	9
3.5. FACTORES DE AUTENTICACIÓN	9
3.6. MANEJO DE SESIONES	10
3.6.1. ADMINISTRACIÓN DE SESIONES	11
3.6.2. CROSS-SITE REQUEST FORGERY - CSRF	12
3.1. AUTORIZACIÓN	13
3.1.1. MANEJO DE REFERENCIAS DIRECTAS A OBJETOS	14
3.1.2. RESTRINGIR EL ACCESO A LA URL	14
3.2. MÍNIMO PRIVILEGIO	15
3.3. VALIDACIÓN DE DATOS DE ENTRADA	15
3.3.1. VALIDACIÓN: FALLAS POR INYECCIÓN	16
3.3.2. OBJECT-RELATIONAL MAPPING	17
3.3.3. EJECUCIÓN DE ARCHIVOS MALICIOSOS	17
3.1. CONTROL DE ACCESO	18
3.1.1. RE-DIRECCIONAMIENTO Y REENVÍO NO VALIDADO	19
3.2. MEJORES PRÁCTICAS EN LA IMPLEMENTACIÓN	20
3.3. COMPONENTES VULNERABLES DE TERCEROS	20
3.4. MANEJO DE ERRORES	21
3.5. COMENTARIOS Y DOCUMENTACIÓN	22
3.6. APLICACIONES QUE MANEJAN TARJETAS DE CRÉDITO	22
3.7. PROTECCIÓN DE DATOS	22
3.8. AUDITORIA	23
4. OWASP TOP 10 – 2021: RIESGO DE SEGURIDAD DE APLICACIONES	24
4.1. BROKEN ACCESS CONTROL	24
4.1.1. DIRECTORIO TRANSVERSAL	25
4.2. FALLAS DE CIFRADO (CRYPTOGRAPHIC FAILURES)	26
4.3. INYECCIÓN (INJECTION)	27
4.3.1. SQL INJECTION	28
4.3.2. XSS - CROSS-SITE SCRIPTING	28
4.4. INSECURE DESIGN	29


Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 2 de 40
--	---	--	--	-------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 3 de 40	

4.5	SECURITY MISCONFIGURATION.....	29
4.6	VULNERABLE AND OUTDATED COMPONENTS	30
4.7	IDENTIFICATION AND AUTHENTICATION FAILURES.....	31
4.8	IDENTIFICATION AND AUTHENTICATION FAILURES.....	32
4.9	SECURITY LOGGIN AND MONITORING FAILURES	33
4.10	SECURITY LOGGIN AND MONITORING FAILURES	34
5.	WAF.....	35
5.1.	CANAL DE COMUNICACIÓN	35
5.2.	MENSAJERÍA PARA CLIENTES	35
6.	PRINCIPIOS DE DISEÑO	36
7.	FUENTES DE VULNERABILIDADES EN EL SOFTWARE QUE SE DEBEN CONSIDERAR.....	37
6	CONTROL DE VERSIONES.....	40

CONFIDENCIAL

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 3 de 40
--	--	--	--	-----------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 4 de 40	

1. INTRODUCCIÓN

1.1. Objetivo

Definir las medidas necesarias para implementar el esquema de seguridad en aplicaciones, según las mejores prácticas de Seguridad de la Información.

1.2. Ámbito de Aplicación

Todas las aplicaciones de Banco Bolivariano que brinden servicio a cliente como las que pertenezcan al entorno de producción de la Entidad.

1.3. Normativa Marco (Normativas Superior de Referencia)

- PC.POL.1 - Política General de Seguridad de la Información

1.4. Normativa Derogada

- Ninguna.

1.5. Otras Normativas Asociadas

- Ninguna.

1.6. Vigencia


Este estándar de configuración entrará en vigencia a partir del 01 marzo de 2023.

1.7. Disposiciones Generales y Transitorias

Este estándar de configuración deberá ser revisado bianual o cuando se requieran por el Área de Seguridad de la Información de Banco Bolivariano. Los resultados de la revisión, y los cambios que se sucedan, serán reportados y comunicados a los involucrados antes de ser implementados.

La falta de cumplimiento de las definiciones descriptas en el presente estándar estará sujeta a las sanciones disciplinarias que amerite cada caso.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 4 de 40
--	---	--	--	-------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 5 de 40	

1.8. Roles y Responsabilidades

1.8.1 Seguridad de la Información:

Responsable de Seguridad de la Información: Tendrá a su cargo el mantenimiento del presente estándar, junto con las tareas de verificación del cumplimiento del mismo.

1.8.2 Gerencia de Desarrollo:


Gerente de Desarrollo: Deberá garantizar que los funcionarios del área de sistemas, implementen los estándares de configuración definidos en forma efectiva y oportuna.

Desarrolladores/Arquitectura: Serán encargados de implementar el presente estándar, siguiendo los lineamientos y tareas mencionadas en el mismo. Asimismo, deberán informar al Gerente de Tecnología sobre las configuraciones de seguridad que no puedan ser implementadas por restricciones técnicas y/o de negocio, las cuales deberán quedar adecuadamente documentadas

Los desarrolladores y arquitectura deberán considerar al grupo de Ingeniería de Seguridad para la implementación de nuevos sistemas o mantenimiento.

Infraestructura: Serán encargados de implementar el presente estándar, siguiendo los lineamientos de seguridad definidos en los estándares para las plataformas implementadas en Banco.

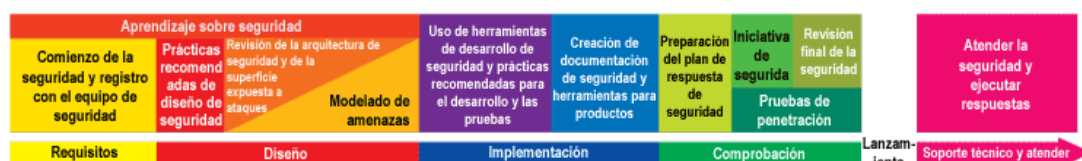
Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 5 de 40
--	---	--	--	-------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 6 de 40	

2. DE LAS GENERALIDADES

2.1 Objetivo

Definir las consideraciones de seguridad necesarias para el correcto desarrollo de software, disminuyendo las vulnerabilidades antes de salir a producción.



2.2 Seguridad a nivel de aplicaciones

Nunca se debe suponer que un producto de software de tamaño y complejidad no complicado está libre de vulnerabilidades de seguridad. Se debe aplicar los pasos necesarios para limitar el número de errores de codificación y reducir su repercusión práctica.


Las vulnerabilidades no solo pueden recaer en la aplicación sino también en la selección de tecnología inadecuada.

2.3 Documentación

Durante el diseño de una aplicación deben existir los siguientes entregables.

- Arquitectura completa que defina todo el alcance del proyecto.
 - Flujos de datos
 - Validación de los datos de entrada y de salida.
 - Control de Accesos
 - Manejo de sesiones.
 - Proceso de desarrollo seguro.
 - Análisis de vulnerabilidades.
 - Publicación de servicios web con certificados.
 - Manejo de Errores.
 - Mecanismos de Autenticación y autorización.
 - Mecanismos de cifrado.
 - Registro de eventos y Auditorias.
 - Reutilización de componentes confiables.
 - Identificación de usuarios, autenticación.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 6 de 40
---	---	---	---	-------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 7 de 40	

- Modelamiento de estructuras de seguridad a nivel físico.
- Protocolos de comunicaciones entre capas físicas y lógicas de información.
- Documentación de configuración WAF (Web Application Firewall)

3. CONSIDERACIONES BÁSICAS DE SEGURIDAD

3.1 Objetivo

Detallar las configuraciones básicas de seguridad en aplicaciones para asegurar la integridad y validez de las mismas.

3.2 Buenas prácticas de contraseñas

Para ayudar a los usuarios a elegir contraseñas seguras, se recomienda que los desarrolladores apliquen un **estándar de complejidad en la creación de contraseñas del Banco Bolivariano**.


Seguir estas reglas ayudará a aumentar la seguridad implícita en torno a las soluciones de autenticación de nombre de usuario y contraseña.

Directrices de contraseña de cadena:

- No permitir contraseña común.
- Se deben cumplir al menos 3 de las 4 reglas siguientes
 - Como mínimo, 10 caracteres de largo.
 - Como máximo, 18 caracteres de largo: Establecer un límite superior como este ayudará a proteger la aplicación contra vulnerabilidades relacionadas con entradas que son demasiado grandes, como desbordamientos de almacenamiento dinámico y búfer.
- Directrices de contraseña de cadena:
- No permitir contraseña común.
- Se deben cumplir las siguientes reglas:
 - Al menos un carácter en mayúscula (A-Z)
 - Al menos un carácter en minúscula (a-z)
 - Al menos un dígito (0-9)
 - Al menos un carácter especial (puntuación). No olvide tratar los caracteres como especiales y codificar la entrada.
- No más de dos caracteres idénticos seguidos. Por ejemplo, no se debe permitir '777'.

Las siguientes son prácticas recomendadas para permitir el uso de administradores de contraseñas:

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 7 de 40
--	---	--	--	-----------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 8 de 40	

- Utilice formularios HTML estándar y escriba atributos para los campos de nombre de usuario y contraseña.
- No limite artificialmente la longitud de una contraseña en la interfaz de usuario mientras mantiene un límite superior más grande en el back-end
- No prohibir el uso de la funcionalidad de copiar y pegar en el formulario de inicio de sesión.
- Evite el uso de páginas de inicio de sesión basadas en complementos.

3.3 Uso de protocolos seguros

- HTTP no es un protocolo intrínsecamente seguro, por lo que la información confidencial que se transmite de esta manera probablemente no sea segura.
- Las aplicaciones que necesitan transferir datos protegidos **deben implementar Transport Layer Security (TLS) versión 1.2**. La mayoría de los principales navegadores de Internet han declarado que dejarán de utilizar las versiones 1.0 y 1.1 de TLS.

3.4 Autenticación

Debe **eliminarse todo tipo de uso autenticación básica HTTP** si todavía están en uso y evitarse por completo para las nuevas aplicaciones.


La codificación Base64 no es un método de cifrado. Cualquiera que capture el tráfico de red que contiene la solicitud puede ejecutar fácilmente la decodificación Base64.

Las aplicaciones deben ofrecer la autenticación multifactor, donde esta sea bien implementada para ofrecer una mayor garantía de seguridad que las opciones estándar.

La autenticación por SMS, no se recomienda debido al aumento de los ataques de "SIM swapping", en los que una parte malintencionada intentará transferir el número de teléfono de la víctima a un dispositivo controlado por el atacante.

Los desarrolladores deben considerar situaciones como: **tokens de sesión robados, usuarios que no cierran sesión y cómo estos y otros problemas de autenticación** afectan el acceso potencial a la información confidencial del usuario.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 8 de 40
--	---	--	--	-------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 9 de 40	

Dependiendo de los tipos de datos que almacene su aplicación, puede ser importante **volver a autenticar a un usuario** que ya inició sesión.

Ejemplo: los usuarios que vuelvan a ingresar sus credenciales cuando intentan acceder o actualizar información personal, realizar una compra o restablecer su contraseña. También se recomienda encarecidamente finalizar las sesiones de un usuario después de que estén inactivos durante un período de tiempo predeterminado.

Las prácticas de re-autenticación brindan un mayor nivel de protección, ya que esto generalmente requiere que los atacantes tengan acceso a la contraseña, el correo electrónico, el teléfono u otros activos del usuario para poder usar con éxito la funcionalidad de la aplicación confidencial.

Del mismo modo, **las aplicaciones que envían notificaciones por correo electrónico o mensajes de texto** a los usuarios cuando se han realizado **transacciones importantes** con su cuenta no solo brindan a los usuarios un método de recurso para recuperar su cuenta en caso de que ocurra una infracción, sino que también resaltan el compromiso del equipo de la aplicación con la seguridad y la ayuda para mejorar la confianza del usuario.


3.4.1 Autenticación continua

- La autenticación continua podría requerir que el **usuario se vuelva a autenticar de otra manera**, alertar al usuario de actividad sospechosa a través de otros medios, como un correo electrónico. o prohibir la acción del usuario hasta que se hayan completado más desafíos de autenticación.
- Si es viable una aplicación podría tener una implementación de bajo nivel de uso de patrones de geolocalización, que **rastrean las ubicaciones** y los **tipos de dispositivos** que normalmente se usan para acceder a una cuenta y **alertan al usuario cuando ocurre una discrepancia**, como cuando se detecta un **dispositivo nuevo** o una **ubicación inusual**.

3.5 Factores de Autenticación

En los servicios de cajeros automáticos y banca electrónica se debe implementar como mínimo dos de los tres factores de autenticación para la ejecución de las transacciones de los clientes.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 9 de 40
--	---	--	--	-------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 10 de 40	

- algo que se sabe – Contraseña o PIN
- algo que se tiene – Tarjeta Inteligente
- algo que se es – Huellas dactilares, patrones de iris e impresiones de voz

Adicional en el servicio de banca electrónica uno de los factores debe ser dinámico y ser una clave de una sola vez (One time password) cada vez que realice una transacción.

3.6 Manejo de sesiones


Es fundamental contar con mecanismos que permitan mantener el estado de una sesión activa posibles casos de fraude o divulgación de información sensible.

- Limitar las sesiones debe ser único, no predecible y resistente a la ingeniería inversa.
- Limitar el tiempo de expiración.
- Regeneración de tokens para prevenir la captura de sesiones.
- Utilización de tokens robustos que no puedan ser vulnerados.
- Bloqueo de sesiones ante la detección de ataques por fuerza bruta o intentos falsificación de sesiones.
- Reutilización de sesiones para transmisión críticas.

Dado que el ID de sesión suele ser una prueba de la autenticación y proporciona acceso a la aplicación, es importante **contar con las protecciones de seguridad** pertinentes para evitar varios tipos de ataques de secuestro de sesión (**session hijacking attacks**).

- Los nombres de **ID de sesión deben ser genéricos** y evitar la divulgación de información técnica innecesaria.
- Las longitudes de **ID de sesión deben ser de 128 bits como mínimo**, para ayudar a prevenir ataques de fuerza bruta en el valor.
- Los **ID de sesión deben crearse utilizando generadores de números aleatorios comprobados con un alto nivel de entropía**.
- Los valores de **ID de sesión no deberían tener significado inherente**, ya que son accesibles en el lado del cliente del navegador. Cualquier información de uso o datos confidenciales asociados a la identificación deben almacenarse en el lado del servidor.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 10 de 40
--	---	--	--	------------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 11 de 40	

- Establecer el indicador de solo seguridad en el **valor de ID de sesión**, de modo que **solo se transporte a través de protocolos de comunicación HTTPS**. Esta práctica brinda seguridad adicional a través de una capa adicional de cifrado.


3.6.1 Administración de sesiones

La **prevención de ataques de secuestro de sesiones (session hijacking attacks)** requiere una sólida práctica de administración de sesiones. "Administración de sesiones" es el mecanismo utilizado para intercambiar continuamente tokens de sesión entre el usuario y la aplicación web.

Las cookies ofrecen a los desarrolladores control sobre una variedad de campos, muchos de los cuales son útiles por motivos de seguridad:

- Secure
 - httponly
 - Expires y Max-age.
- **Secure:** indica a un navegador que solo intercambie el valor a través de una conexión cifrada. Es obligatorio utilizar que este indicador se establezca en el token de ID de sesión para evitar que los ataques de intermediarios lo intercepten en una conexión sin cifrar.
 - *Sin embargo, esta no es una solución perfecta y es importante tener en cuenta que, si un servidor web establece el indicador de seguridad en una cookie, pero de todos modos la envía al navegador del usuario a través de una conexión insegura, la cookie seguirá estando en riesgo de ser interceptada.*
 - **Http-only:** no permite el acceso del lado del cliente a los datos de una cookie a través de herramientas de desarrollo basadas en navegador que mitigan un vector de ataque específico para secuencias de comandos entre sitios. Este indicador le indica al navegador que solo exponga los valores de las cookies a través de conexiones HTTP y HTTPS, lo que evita que los scripts del lado del cliente recuperen los datos de la cookie.
 - **Max-Age y Expires:** son dos pares clave-valor que se pueden configurar en una cookie para indicarle al navegador cuándo se debe eliminar la cookie. La variable expire establece una fecha y un tiempo para la eliminación, mientras que Max-Age establece un intervalo de tiempo activo, a partir de que el navegador recibió la cookie.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 11 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 12 de 40	

3.6.2 Falsificación de solicitud entre sitios - CSRF

Es un tipo de ataque en el que un actor malintencionado secuestra y autentica la sesión válida del usuario, lo que **obliga al usuario a realizar acciones no deseadas sin darse cuenta**.


Debido a que el atacante no puede recuperar la respuesta de la acción, los ataques **CSRF generalmente manipulan a un usuario para que realice una acción que resulte en un cambio de estado**, en lugar de intentar robar datos.

El impacto de los ataques CSRF exitosos puede ser extremadamente devastador según el nivel de acceso del usuario, la funcionalidad de la plataforma y los controles de autorización establecidos. **Los ataques exitosos pueden ser pequeños, como cambiar la dirección de correo electrónico del usuario en la cuenta, o grandes, como transferir fondos de una cuenta a otra**. Si el usuario tiene acceso administrativo, el ataque puede tener un impacto aún mayor debido a los privilegios elevados del usuario.

¿Cómo evitar?

- **El método recomendado para mitigar los ataques CSRF es un enfoque basado en tokens.** La aplicación asigna tokens específicos de transacciones o sesiones aleatorias criptográficamente fuertes que deben incluirse en la solicitud del formulario. Es importante hacer la distinción de que estos tokens suelen ser una parte oculta del campo de formulario y no un valor de cookie.
- **Considere el atributo de cookie de SameSite** para las cookies de sesión, pero tenga cuidado de NO configurar una cookie específicamente para un dominio, ya que eso introduciría una vulnerabilidad de seguridad que todos los subdominios de ese dominio comparten la cookie.
- No utilice solicitudes GET para operaciones de cambio de estado.
- una defensa alternativa es utilizar la técnica de cookies de envío doble. Esta técnica es fácil de implementar y no tiene estado. En esta técnica, enviamos un valor aleatorio tanto en una cookie como en un parámetro de solicitud, y el servidor verifica si el valor de la cookie y el valor de la solicitud coinciden. Cuando un usuario visita (incluso antes de autenticarse para evitar el inicio de sesión CSRF), el sitio debe generar un valor pseudoaleatorio (criptográficamente fuerte) y configurarlo como una cookie en la máquina del usuario separada del identificador de sesión. Luego, el sitio requiere que cada solicitud de transacción incluya este valor pseudoaleatorio como un valor de formulario oculto (o como un parámetro/encabezado de solicitud). Si ambos coinciden en el lado

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 12 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
	Código: PC.EST.1.23	Revisión: v. 5 - 2018
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Página 13 de 40	

del servidor, el servidor lo acepta como una solicitud legítima y, si no lo hacen,

3.1 Autorización

Es importante que las aplicaciones incluyan **métodos como roles y grupos de aplicaciones**, que categorizan y hacen cumplir los permisos de diferentes maneras según las acciones que pueden realizar los distintos tipos de usuarios, normalmente se conoce como **autorización programática** a través del control de acceso basado en roles (RBAC).

Existe otro tipo de **autorización** llamada **programática** usa reglas que se establecen y **aplican en el propio código de la aplicación**; esta técnica rara vez se usa por sí sola, generalmente solo en casos en los que pequeñas partes de la aplicación necesitan protección.


Las **funciones de autorización varían en su implementación** y el equipo de desarrollo es el responsable final de diseñar una solución segura. Afortunadamente, hay varios métodos posibles para implementar esta funcionalidad.

Para las aplicaciones cliente/servidor, incluidas las aplicaciones web, es crucial implementar cualquier funcionalidad de autorización en el lado del servidor.

- **Ocultar enlaces o bloquear el control y los botones en el lado del cliente** puede ocultar inicialmente la funcionalidad restringida, pero es fácil para los atacantes descubrir estas partes de una aplicación a través de parámetros de fuerza bruta, examinando solicitudes y respuestas, o simplemente conjeturas.
- La oscuridad no es seguridad y **la autorización debe aplicarse en el lado del servidor para que sea efectiva**.

La mayoría de los marcos web proporcionan mecanismos integrados y confiables para administrar la configuración de autorización y las funciones de los usuarios. Sin embargo, cada aplicación tendrá requisitos diferentes para las pautas de flujo de trabajo, los accesos a URL y otros problemas de autorización, por lo **que los equipos de desarrollo deben evaluar cuidadosamente las opciones disponibles para crear la solución más adecuada para las necesidades de sus aplicaciones**.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 13 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 14 de 40	

3.1.1 Manejo de referencias directas a objetos

En inglés, **Handling Direct Object References** hace que cualquier parámetro que se envíe a través del URL tiene un potencial riesgo a ser modificado. Si se proporcionan identificadores para características o datos de la aplicación en cualquier parte de la solicitud del usuario,

La aplicación debe verificar que el usuario tiene autorización para ver, usar o actualizar el recurso solicitado. Idealmente, el diseño de su aplicación evitará por completo la exposición de claves o identificadores de datos internos.

Ejemplo, podría considerar crear una tabla de referencia intermedia para la sesión de cada usuario. Los valores temporales expuesto al usuario están asociados con el identificador real del objeto en el lado del servidor, lo que evita que estos datos internos confidenciales estén expuestos al usuario.

3.1.2 Restringir el acceso a la URL

Los problemas de acceso a URL son un riesgo significativo orientado a la web, especialmente cuando las aplicaciones tienen diferentes niveles de acceso, como un grupo de usuarios administrativos que puede ver páginas y funciones restringidas, en contraste de usuarios finales. Estos ataques suelen constituir una violación de la confidencialidad.


Un entorno de aplicación también puede contener copias o copias de seguridad de varias páginas del servidor. **La presencia de dichos archivos indica prácticas de implementación deficientes**, ya que presenta la posibilidad de que un atacante puede obtener acceso a funciones obsoletas o interactuar con funciones destinadas únicamente a desarrolladores y personal de soporte.

En algunos casos, **particularmente cuando se encuentran extensiones de archivo inusuales, el servidor puede devolver el contenido del archivo**, lo que resulta en la divulgación del código fuente u otros detalles de configuraciones confidenciales.

Es posible que una parte malintencionada use fuerza bruta para ubicar archivos o rutas de aplicaciones con herramientas automatizadas disponibles gratuitamente.

Las prácticas de implementación deben garantizar que los archivos confidenciales nunca se coloquen en directorios web públicos. Esto incluye los scripts de compilación o las utilidades en las que se basa la

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 14 de 40
---	---	---	---	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 15 de 40	

aplicación, así como los repositorios de código en desuso o los archivos de copia de seguridad.

3.2 Mínimo privilegio

Es un principio rector crítico para la configuración de la aplicación, que dicta que los procesos de **la aplicación solo deben tener la autoridad mínima necesaria cuando interactuaran con otros activos o sistemas**.

La intención del privilegio mínimo es tanto **prevenir la explotación de vulnerabilidades de aplicaciones como minimizar el impacto de las explotaciones exitosas cuando ocurran**.

El mínimo privilegio **debe aplicarse a todas las interacciones de una aplicación con otras entidades**, que comúnmente incluyen base de datos y sistemas de archivos.

3.3 Validación de datos de entrada.

En todas las aplicaciones se deben **realizar validaciones de ingreso de datos o parámetros tanto del lado del cliente como del lado del servidor**.


Se debe asegurar la integridad de los datos proporcionados por los usuarios, otros sistemas o almacenes de datos. Se debe **asegurar que cualquier dato que ingrese a un sistema coincida con el tiempo, tamaño y formato esperado**.

Cualquier dato que ingrese a una aplicación a través de un límite de confianza tiene el potencial de ser malicioso y debe analizarse como un posible ataque.

Una mejor estrategia de validación de entrada, **es tener una lista de entrada permitidas**, es más fácil de implementar, captura con mayor precisión la funcionalidad prevista y requiere menos gastos generales. Una lista de permitidos implicar **verificar que los datos proporcionados por el usuario se ajusten a los límites de los que se esperan para un campo de entrada**.

Ejemplo: Si la entrada esperada es el apellido de una persona, la validación verificaría que cada carácter sea alfabético, un apóstrofe o algún carácter especial la aplicación debería rechazar los valores. Es importante que los valores ingresados sean limpiados y no devuelva dichos atributos ingresados, pero podría generar una vulnerabilidad adicional.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 15 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 16 de 40	

Los datos no deben validarse contra especificaciones “negativas”, dado que va a ser difícil filtrar los casos que se presenten, la validación es recomendable realizarse de la siguiente manera.

- Tipos de datos.
- Conjunto de caracteres permitidos.
- Longitudes mínimas y máximas.
- Validación de datos nulos, requeridos o repetidos.
- Rango numérico.
- Valores y patrones específicos permitidos.

La falta de validación puede generar los siguientes inconvenientes:

1. La aplicación espera recibir un ID numérico y en su lugar recibe un ID alfanumérico. Al no validarse el tipo de dato se genera un error de ejecución que es mostrado al usuario.
2. Las comunicaciones pueden ser interceptadas, permitiendo alterar la información transmitida entre el usuario y la aplicación.
3. Nunca se debe almacenar información crítica en las cookies.
4. Almacenar las cookies de la aplicación web de forma encriptada.
5. Definir tiempo de expiración de las cookies de 10 días.

3.3.1 Validación: Fallas por inyección


Los problemas de inyección ***pueden ocurrir siempre que una aplicación incluya una entrada*** que no sea de confianza en una comunicación enviada a otro interprete.

Ejemplo: un atacante podría inyectar código arbitrario en solicitud LDAP, comandos de SHELL del sistema operativo, analizadores XML entre otros. La aplicación de la desinfección adecuada para todos los datos no confiables enviados a un intérprete ayudara a aumentar la postura de seguridad de una aplicación contra los ataques de inyección.

Mitigación de fallas de inyección

- Asegúrese de que las ***entradas se traten como datos***, en lugar de ejecutarse o interpretarse como código.
- ***Los valores que no son de confianza deben codificarse correctamente*** para el intérprete que recibe los datos.
- ***Utilizar procedimientos almacenados y declaraciones preparadas.***

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 16 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 17 de 40	

- **Especifique el código de la declaración SQL que se ejecutará**, así como las variables que se deben codificar antes de pasar al intérprete SQL.
- En última instancia, se necesita la codificación, ya que siempre habrá casos en los que las entradas contengan caracteres válidos.
- La **validación de entrada es una técnica de defensa valiosa**, pero debe combinarse con la codificación de salida adecuada para que sea eficaz.

3.3.2 Object-Relational Mapping

El mapeo relacional de objetos, o ORM, **es una práctica popular de codificación de base de datos** que vincula fuentes de datos de back-end con objetos de datos en la aplicación.

Permiten que la aplicación recupere y actualice datos manipulados los objetos asignados, en lugar de crear y ejecutar consulta SQL directas.


Esta funcionalidad **proporciona codificación implícita a los caracteres SQL persisten a través de estos objetos**. Es importante tener en cuenta que, si bien las soluciones de mapeo relacional de objetos **pueden ayudar a aumentar su perfil de seguridad**, una solución mal implementada aún puede dejar la base de datos vulnerable a los ataques.

3.3.3 Ejecución de archivos maliciosos

Ocurre cuando una aplicación acepta o ejecuta ciegamente archivos o nombres de archivo proporcionados por el usuario. Esto puede **permitir que un atacante ejecute código arbitrario** que se ejecuta con los mismos niveles de permisos que la aplicación o el servidor web.

Es común, aunque peligroso, que las aplicaciones coloquen el archivo cargado en un directorio accesible desde la web para que los usuarios accedan más tarde a través de un enlace directo. **Si el atacante carga un archivo malicioso con una extensión que el servidor puede procesar directamente**, solicitar el archivo puede hacer que la aplicación web lo trate

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 17 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST. 1.23	Revisión: v. 5 - 2018
	Página 18 de 40	

como una página del servidor u otro contenido ejecutable y lo ejecute sin validar la legitimidad del recurso.

En los casos más graves, ***el atacante puede cargar archivos como shells web u otros scripts maliciosos para tomar el control del servidor.***

La práctica común de usar referencias de enlaces directos al contenido cargado puede presentar riesgos de seguridad adicionales para su aplicación web. Si se hace referencia a los archivos por su nombres o ubicaciones reales, puede abrir la puerta para que los usuarios maliciosos accedan a materiales que no deben ver o descargar.


También puede ser una práctica beneficiosa validar las características de cualquier contenido proporcionado por el usuario contra una lista de atributos permitidos.

Si su aplicación debe manejar contenido de archivo, ***asegúrese de que cualquier funcionalidad de desempaquetado este configurada para que las extensiones de archivo***, los tamaños y otros atributos peligrosos se validen contra la lista de caracteres permitidos.

3.1 Control de Acceso

- Permite a la aplicación determinar quién puede acceder a un recurso o función y quien no, estas validaciones deben realizarse en la fase de diseño y siguiendo los estándares y políticas de control de acceso de la Entidad.
 - Id seguros claves robustas, caracteres especiales, etc).
 - Los Id's deben ser validados cada vez que sean utilizados.
- Las aplicaciones no deben depender del secreto de ningún ID, los mismos deben ser validados cada vez que sean utilizados.
 - Las verificaciones de seguridad insertadas en la aplicación no puedan ser evitadas.
 - Se debe publicar archivos que deban ser accedidos por los usuarios y que los mismos cuenten con los permisos adecuados.
 - Toda aplicación cuando va a realizar su pase a producción debe contar con los ID de usuarios debidamente renombrados y cambiadas las claves por default.
 - No deben existir en producción usuarios ni claves por default
 - Las claves de ingreso a los canales electrónicos y aplicaciones internas deben ser generadas y validadas con herramientas para este propósito.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 18 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 19 de 40	

- Las claves de ingreso al servicio de cajeros automáticos no se deben ser almacenadas.
- Se debe presentar enmascarada el ingreso de las claves en los canales electrónicos y aplicaciones internas.
- Se debe bloquear la cuenta del usuario cuando ingrese erróneamente máximo 3 veces la clave en medios de canales electrónicos y aplicaciones internas.
- Se debe solicitar al usuario la renovación de clave de acceso a los canales electrónicos y aplicaciones internas por lo menos una vez al año.

Las deficiencias en los mecanismos de control de acceso podrían ser utilizados por un atacante o un usuario malicioso, para utilizar identificadores de usuarios de terceros, acceder a información sensible, etc.

1. El atacante ingresa a la aplicación con un usuario válido, luego altera la URL y se hace pasar un tercero.
2. Debido a que la aplicación no maneja adecuadamente el control de accesos, el atacante podrá imperdonarse en una cuenta de usuario que lo pertenece.


3.1.1 Re-direccionamiento y reenvío no validado

Las vulnerabilidades de re-dirección y reenvío no validados ocurren cuando una aplicación usa redirecciones para enviar a los usuarios a una URL diferentes o reenvía para evitar a los usuarios a otra sección del mismo sitio.

Si los parámetros de destino no están validados, es posible que **el atacante engañe a un usuario para que visite un sitio malicioso o que el atacante eluda los controles de acceso** dentro de una aplicación y obtenga acceso a la funcionalidad que no está autorizado a ver.

Los atacantes pueden crear direcciones URL que eluden los permisos de autenticación adecuados, otorgándoles acceso a página de administración, registros de usuarios y otros puntos finales confidenciales dentro de la aplicación. Estos ataques tienen éxito debido a que la aplicación no verifica que el parámetro de destino proporcionada sea válido o confirma que el usuario tiene autorización adecuada para solicitar ese recurso.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 19 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 20 de 40	

3.2 Mejores prácticas en la implementación

Se deben mantener entornos independientes configurados de forma idéntica para el desarrollo, las pruebas y la producción. Cada uno de estos entornos debe tener diferentes credenciales de acceso, para evitar que los demás estén en riesgo en caso de que uno de ellos sea violado.

El entorno de prueba solo debe usar datos ficticios, nunca información de usuario en vivo extraída del sistema de producción. Cualquier **mensaje de error que muestre la aplicación debe generalizarse para evitar que los seguimientos** de la pila y otros detalles de depuración se revelen a los usuarios, ya que esta información puede divulgar datos confidenciales del sistema que el atacante puede usar para crear vulnerabilidades específicas.


Los registros deben revisarse periódicamente en busca de eventos o comportamientos inusuales e, idealmente, el sistema debe tener alguna capacidad para alertar al equipo cuando se detecta actividad sospechosa.

3.3 Componentes vulnerables de terceros

Para reducir los riesgos que plantean los componentes vulnerables de terceros, se **deben tomar las siguientes medidas**:

- **Inventario:** Investigue y catalogue las dependencias para todas las aplicaciones personalizadas. Debe *incluir detalles de la versión, parches y fechas de actualización, información sobre la licencia* y notas sobre cualquier vulnerabilidad conocida asociada con el software del componente. Realice pruebas de seguridad en los proyectos.
- **Analizar:** Desarrolle un proceso de investigación para agregar nuevas bibliotecas o marcos a un producto o proyecto. Antes de seleccionar nuevos componentes de terceros investigue la información, y divulgaciones de vulnerabilidades.
- **Control:** Cree un repositorio de estos marcos y bibliotecas verificados, así como un proceso para que los desarrolladores descargue estos paquetes de software aprobados.
- **Monitor:** Asigne recursos para administrar las actualizaciones del marco y la biblioteca. Asegurase de que esto incluya tiempo adicional para re factorizar el código para tener en cuenta las actualizaciones de software de terceros, eliminar la deuda técnica y contribuir a la calidad del código.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 20 de 40
--	---	--	--	------------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 21 de 40	

La organización debe tener un proceso bien definido para mantener configuraciones seguras, un plan proactivo para administrar parches futuros para todos los componentes y subsistemas de la aplicación.

Las bibliotecas deben estar actualizadas con las últimas actualizaciones y parches en el momento en que se implementa la aplicación.

Mitigar el uso de frameworks, bibliotecas y otros componentes potencialmente vulnerables. Control, marcos aprobados y bibliotecas mediante el uso de repositorios locales. Haga un inventario de todas las dependencias de la aplicación e investigue sus últimos parches y conozca las vulnerabilidades.

3.4 Manejo de errores

Un manejo inadecuado de los errores de una aplicación puede brindar información valiosa para iniciar un ataque.

Las aplicaciones deberán **limitar la posibilidad de exposición de la información** no intencional.


Los equipos de desarrollo deben tener en cuenta el manejo de errores al largo de las fases de diseño, codificación y control de calidad para garantizar que las aplicaciones respondan de manera consistente y seguras a las condiciones de falla.

Ejemplo: Un riesgo de manejo de información ocurre cuando una aplicación informa "Nombre de usuario no válido" o "Contraseña no válida" tras un intento fallido de inicio de sesión en lugar del mucho más genérico "Nombre de usuario o contraseña no válido". Si bien es muy básico, especificar que variable es incorrecta ilustra la importancia de devolver mensajes de error explícitamente agnósticos en torno a la funcionalidad sensible.

Generalmente se observa que los errores de las capas de datos, aplicaciones y presentación no son adecuadamente manejados y en consecuencia terminan siendo presentados al usuario final, por ejemplo:

- Volcado de pilas (stacks)
- Errores de base de datos
- Errores del servidor Web.
- Código de errores propios de la aplicación.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 21 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 22 de 40	

Los mensajes funcionales que se presentan a los usuarios muchas veces exponen información sensible de la aplicación.

Los mensajes de error deben informar al usuario sin revelar información del funcionamiento de la aplicación.

3.5 Comentarios y documentación

Los comentarios pueden presentar un riesgo de divulgación de información, en particular los comentarios HTML, los comentarios JavaScript u otro contenido descriptivo enviado a los usuarios como parte de una respuesta.

En los aplicativos con back-end no se debe permitir el registro de comentarios. Los atacantes examinan rutinariamente los comentarios HTML en busca de cualquier información que pueda ayudar en los ataques, como credenciales de cuentas de prueba o suposiciones explotables sobre la lógica de la aplicación.

No crear archivos de respaldo dentro de servidores de producción. Los atacantes pueden buscar archivos de respaldo implementados u otros activos que no son esenciales para la funcionalidad estándar de la aplicación.

3.6 Aplicaciones que manejan Tarjetas de Crédito

Deben incluir en la definición de aplicaciones el número de PAN no se debe presentar ni almacenar de forma clara, ni se deben generar reportes o archivos (por ejemplo, en la pantalla, en recibos en papel), excepto en los casos en que existe una necesidad comercial de negocio avalada para visualizarlo así.


3.7 Protección de datos

Es obligatorio aplicar controles PCI-DSS cuando la aplicación manipule o almacene datos de tarjeta crédito y otra información confidencial.

Las aplicaciones **no deberán utilizar algoritmos de cifrado con debilidades conocidos o algoritmos personalizados** por desarrolladores internamente para proteger los datos.

Para la protección de datos se **debe utilizar bibliotecas criptográficas públicas bien examinadas** creadas específicamente para estos fines.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 22 de 40
---	---	---	---	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 23 de 40	

Las aplicaciones **deben evitar ser susceptibles a prácticas operativas deficientes como dejar claves codificadas en el código fuente** o almacenar claves en ubicaciones desprotegidas, como archivos de configuración.

Cuando una aplicación deba retener la información de pago para que los usuarios no tengan que volver a ingresar su tarjeta de crédito cada vez que realicen una compra, **se debe implementar dentro de la aplicación el cifrado simétrico** para el almacenamiento de los datos.

Las aplicaciones deberán tener implementado el algoritmo simétrico más confiable y de uso común como AES.

Adicionalmente, es necesario que **las aplicaciones legacy que tenga implementado algoritmos como DES, 3DES, Blowfish, RC2, RC4 y RC5, estos algoritmos deben eliminarse gradualmente** en la base de código donde aparecen. Para los nuevos proyectos los equipos de desarrollo nunca deben usar estos algoritmos obsoletos.

Las aplicaciones no deberán tener implementado algoritmos hash como MD5, RMD160 y SHA-1, se consideran obsoletos. Los algoritmos recomendados son scrypt, argon2, bcrypt y PBKDF2.

Los desarrolladores deben analizar todos los datos procesados por sus aplicaciones y protegerlos de acuerdo con los estándares determinados por la Política de Seguridad.

Las claves criptográficas deben almacenarse de manera segura, separada del código de la aplicación. De ser posible se puede optar por implementar Key Management System (KMS).


3.8 Auditoria

Se debe **contar con mecanismos de registro de eventos de** seguridad que permitan analizar las acciones realizadas por los usuarios en los sistemas.

Deben existir registros con fecha hora y acción realizada.

La aplicación debe contar con un LOG de accesos y con un LOG que registre las actividades sensibles llevadas a cabo por los usuarios en la aplicación. Asimismo, dichos LOGs de seguridad deben estar adecuadamente protegidos.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 23 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 24 de 40	

Debe garantizarse que **no existan funcionalidades que puedan ser accedidas en forma directa sin media autenticación**, aunque las mismas no permitan operar al usuario.

No se debe de almacenar en los logs información de TC en claro.

El entorno de producción debe estar libre de archivos de resguardo (backups) y códigos fuente de prueba.

Los registros de auditoria de aplicaciones adecuados **deben permitir que el equipo de soporte del aplicativo reconstruya la secuencia de eventos** que llevaron a un compromiso, error o interrupción.

En caso de compromiso, **los registros de auditoría pueden brindar detalles sobre que vulnerabilidades** o métodos se usaron para infiltrarse en los sistemas.

4. **OWASP Top 10 – 2021: Riesgo de seguridad de aplicaciones**

4.1 Pérdida de control de acceso

Los problemas de control de acceso ocurren cuando los usuarios, autenticados o no, **actúan fuera de los permisos previstos** de su función de cuenta.


Si los usuarios **pueden eludir las comprobaciones de control de acceso modificando los parámetros de URL, el contenido HTML, los estados de sesión de la aplicación interna** o mediante otros mecanismos, su aplicación tiene una vulnerabilidad de control de acceso.

Su aplicación es vulnerable **si los usuarios pueden actuar con privilegios elevados**, lo que significa funcionar como un usuario válido cuando no está autenticado o actuar como administrador cuando está autenticado como usuario estándar.

¿Cómo prevenir?

Denegar el acceso de forma predeterminada al configurar el modelo de permiso de su aplicación, excepto para los recursos públicos.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 24 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 25 de 40	

Deshabilite las listas de directorios en su servidor web y asegúrese de que los archivos confidenciales, como los metadatos de la aplicación y las copias de seguridad, no se almacenen en los directorios raíz.

Asegúrese que todos los token web y de sesión de invaliden correctamente en el servidor después del cierre de sesión de los usuarios.

Registre las fallas de control de acceso de forma predeterminada y tenga un mecanismo para alertar a los administradores en caso de actividades sospechosas, como fallas de inicio de sesión repetidas, acceso a API de gran volumen y otras detecciones similares.

La mejor manera de detectar un control de acceso faltante o ineficaz es a través de pruebas manuales cuidadosas y exhaustivas. Por lo tanto, es importante incluir pruebas unitarias y pruebas de integración que verifiquen el control de acceso funcional.

4.1.1 Directorio transversal

Directory traversal, o **path traversal attacks** ocurren cuando una parte malintencionada puede usar entradas especializadas para escapar fuera de las ubicaciones restringidas del servidor o del sistema de archivos, lo que les permite acceder a archivos o directorios que no deberían poder ver.


El error de tener un **Directory Transversal** pueden ser el resultado de malas prácticas de codificación de aplicaciones o una configuración incorrecta del sistema, pero sin importar cuál sea su origen, estas vulnerabilidades tienen consecuencias graves, como la divulgación de contraseñas de usuarios e información personal, o la exposición de datos confidenciales como el sistema. detalles de configuración y secretos comerciales.

El problema ocurre cuando las aplicaciones dependen de la información proporcionada por el usuario para construir nombres de ruta de solicitud a los recursos locales sin desinfectar adecuadamente los caracteres especiales utilizados para el acceso al sistema de archivos.

¿Cómo prevenir?

- Se recomienda que los documentos de cara al público se alojen en un servidor de archivos separado o en una ubicación de almacenamiento en la nube, lejos del material confidencial dentro del sistema.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 25 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 26 de 40	

- Si la aplicación necesita manejar un gran volumen de documentos, también podría **considerar usar un sistema de administración de contenido de terceros** en lugar de crear flujos de trabajo complejos de carga, indexación y publicación desde cero.
- Implementar fuertes políticas de control de acceso para **restringir los directorios de los que un usuario puede recuperar archivos** o guardar archivos.
- **Garantizar que los procesos del servidor se ejecuten solo con los niveles de privilegio mínimos** necesarios para la tarea requerida.
- Asegúrese de que su sistema **reciba mantenimiento y parches con las últimas actualizaciones de seguridad** y estabilidad tan pronto como sea práctico desde el punto de vista organizativo.
- Almacene y recupere archivos utilizando un ID de archivo aleatorio generado de forma segura
- **Verifique los nombres de archivo en una lista blanca** de recursos conocidos que los usuarios pueden solicitar.
- Siempre que sea posible, **evite el uso de la entrada del usuario por completo al interactuar con el sistema de archivos** y los recursos del sistema.

4.2 Fallas de criptográficas


Se trata cuando **la aplicación usa cifrado con algoritmos débiles** o depender de métodos de generación de claves defectuosos.

Las aplicaciones que transmiten datos sin cifrar, ya sea de forma interna o externa, o aquellas que almacenan datos en texto claro pueden estar en riesgo.

¿Cómo prevenir?

- **Confirme que todos los datos confidenciales están cifrados** mientras en están en reposo y que **los datos en tránsito utilicen protocolos**

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 26 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 27 de 40	

seguros como TLS, con cifrados de confidencialidad directos perfecto y parámetros seguros.

- **Aplique** el cifrado con directas **HTTP Stric Transport Security (HSTS)** o mecanismos similares.

4.3 Inyección


Sucedan cuando un intérprete acepta retos que no son de confianza como parte de una consulta o comando de búsqueda. Esto **puede permitir que los atacantes vean los datos a los datos que no tienen autorización** para acceder, hasta ejecutar comandos arbitrarios en el sistema, ejecución remota de código y otras acciones igualmente peligrosas.

Las vulnerabilidades de inyección ocurren cuando los datos se concatenan directamente o funcionalidad del sistema, como cadenas consulta SQL o comandos del sistema operativo, sin desinfección o filtrado previsto.

¿Cómo prevenir?

- **Se debe garantizar que los datos externos se validen o limpien antes de que se utilicen en consultas internas** o con otra funcionalidad crítica. El método ideal para proteger su aplicación de fallas de inyección es usar una API segura que proporcione una interfaz parametrizada y tal vez incluso evite por completo el uso de un intérprete.
- **Se puede usar la validación de entrada contra una lista de permitidos del lado del servidor.**
- **Si debe usar consultas dinámicas**, los caracteres especiales deben escaparse correctamente de acuerdo con las convenciones del intérprete relevante.
- **Se debe tener presente que incluso en procedimiento almacenados parametrizados** si la consultas o los datos se concatenan directamente en el procedimiento.
- **Revisar el código fuente de su aplicación** es el mejor método para descubrir vulnerabilidades de inyección.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 27 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 28 de 40	

4.3.1 SQL injection

La técnica de inyección de código **permite a un atacante a transmitir código malicioso a otro sistema través de una aplicación web**. Estos ataques incluyen llamadas al sistema operativo, el uso de programas externos vía interfaz de comandos y llamadas a servidores de base de datos mediante la utilización de comandos SQL.

Dado que muchas aplicaciones utilizan características del sistema operativo y programas externos para realizar sus funciones, la validación inadecuada de los parámetros de entrada puede resultar en la ejecución arbitraria de comandos por parte de personas no autorizadas.

Se debe evitar el acceso a programas externos para realizar funciones específicas de la aplicación.

Se recomienda el uso de “stored procedures” y “prepared statements” que permitan asegurar que los datos introducidos son tomados como datos precisamente y no como instrucciones a ejecutar

4.3.2 XSS - Cross-Site Scripting

El XSS es una vulnerabilidad propia del sitio web y deriva de la inadecuada validación ser confiable y no reconocer que se trata de un ataque.

La mitigación de XSS deben abordarse mediante la validación adecuada de los datos que ingresan a la aplicación y a la codificación de los datos que salen de la aplicación.

Los ataques del tipo XSS generalmente se presentan en la forma de JavaScript incrustado. Sin embargo, cualquier contenido activo incrustado es una fuente potencial de daño incluyendo: ActiveX (OLE), VBScript, Shockwave, Flash, etc.


Recomendaciones:

La mejor forma de proteger una aplicación es asegurar que la misma valida todos los encabezados, cookies, peticiones, campos de formularios y campos ocultos contra un estándar riguroso de datos válidos.

Codificar la cadena de caracteres a filtrar utilizando “HtmlEncode”.

Es recomendable aplicar una política de validación “positiva” a los datos de entrada.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 28 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 29 de 40	

Se puede **implementar una lista entradas validadas y permitidas** para proporcionar una postura de seguridad defensiva a profundidad al evitar que se acepten entradas inesperadas en el sistema.

4.4 Diseño Inseguro

Si el diseño es inseguro, la implementación perfecta no ayudará porque, por definición, nunca se establecieron las medidas de seguridad necesarias para defenderse de los ataques.

Algunas debilidades se pueden identificar con bastante anticipación, antes de la etapa de implementación.

¿Cómo prevenir?

- Consulte a los profesionales de AppSec y utilice un ciclo de vida de desarrollo seguro para evaluar y diseñar controles relacionados con la seguridad y la privacidad.
- Realizar pruebas unitarias y de integración.
- Limite el consumo de recursos por usuario o servicio.

4.5 Configuración de Seguridad incorrecta


Estas vulnerabilidades a menudo se pueden aprovechar para obtener acceso no autorizado a datos y funciones privados e incluso, en ocasiones, para comprometer todo el sistema.

La gravedad de este ataque dependerá de los detalles específicos de como se ha configurado incorrectamente su aplicación y los tipos de datos que intentan proteger.

La vulnerabilidad puede estar presentes cuando:

- Deje habilitado los puertos y funciones innecesarios.
- Deje los permisos de cuenta predeterminados en su lugar.
- Proporcionar mensajes de error demasiados detallados.
- Configuración incorrecta u olvido habilitar las funciones de seguridad.
- Usar software obsoleto o software con vulnerabilidades conocidas
- No se puede usar encabezados de seguridad

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 29 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 30 de 40	

¿Cómo prevenir?

- **Implemente proceso de configuración y fortalecimiento** repetibles e idealmente automatizados para implementar nuevos sistemas.
- **Sus entornos de producción, desarrollo y testing deben configurarse de manera idéntica**, pero deben usar credenciales de inicio de sesión diferentes en cada entorno.
- Trate de manera su plataforma lo mas mínima posible. **Elimine los programas no utilizados y otros componentes** y evite instalar software innecesario.
- Realice una **revisión periódica de la administración de parches** para verificar las actualizaciones relevantes, los boletines de seguridad y los ajustes de configuración actuales.
- Las cuentas de usuario **deben utilizar permisos granulares** con un modelo de privilegios mínimos.
- **Asegúrese de actualizar y parchear regularmente** todos los procesadores y bibliotecas XML en sus aplicaciones y sistemas operativos host.
- **Deshabilite el procesamiento de definición de tipo de documento y entidad externa XML.**


4.6 Componentes Vulnerables y Desactualizados

La funcionalidad de terceros puede incluir componentes vulnerables. Los equipos de desarrollo no siempre conocen todos los componentes utilizados en su aplicación o API, y mucho menos los boletines de seguridad relevantes y las actualizaciones disponibles para estas partes dispares.

¿Cómo prevenir?

- Implementar un sólido proceso de administración de parches. Esto debe incluir la eliminación de dependencias y características no utilizadas, y parches y actualizaciones de software obsoletos.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 30 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 31 de 40	

- Asegurarse de que haya un esfuerzo continuo para detectar, clasificar y actualizar componentes obsoletos o configuraciones de infraestructura para todos los ciclos de vida del sistema.
- Solo obtengan componentes a través de enlaces seguros de fuente oficiales.
- Realice un inventario continuo de las versiones de todos los componentes y dependencias que utiliza supervise fuentes como MITRE o la base de datos nacional de vulnerabilidades para detectar problemas relevantes.
- Verifique las bibliotecas y los componentes utilizados en su aplicación para ver cuales ya no se admiten actualizaciones de seguridad.

4.7 Fallas por identificación y Autenticación

Los problemas de autenticación se **pueden aprovechar para comprometer contraseña y tokens de sesión**, o asumir las identidades de usuarios válidos y acceder a información personal confidencial.

Se **debe verificar como almacenan internamente las credenciales**, ya que las aplicaciones que almacenan contraseña en texto sin formato o cifradas con algoritmos débiles tiene un alto riesgo de compromiso.


Las aplicaciones de software personalizadas, en lugar de método validados por la industria, para generar tokens de sesión, y otra información, que debe ser verdaderamente aleatoria o criptográficamente sólida.

Si aplicación expone los ID de sesión en la URL, no rota los ID de sesión después de un inicio de sesión correcto o no invalida correctamente las sesiones de usuario o los tokens autenticación después de que un usuario cierra sesión.

¿Cómo prevenir?

- Asegúrese de **cambiar todas las credenciales predeterminadas**, especialmente para usuarios privilegiados como administradores o cuenta de los sistemas.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 31 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 32 de 40	

- **Limite los intentos de inicio de sesión fallidos** o retrase cada vez más los intentos de inicio de sesión después de una cierta cantidad de fallas.
- **Configure su sistema de monitoreo para alertar a los administradores** si se detectan intentos de acceso automático y asegúrese de registrar todos los intentos de acceso fallidos para una revisión posterior.
- **Refuerce los mecanismos de recuperación de credenciales** de su aplicación, el registro de cuentas y otras vías API relevantes al proporcionar un mensaje de error relevante, pero sin detalles para cada resultado.
- **Implementar la funcionalidad de generación y gestión de sesiones.** La administración de sesión debe estar integrada en el lado del servidor, y las nuevas ID de sesión de alta entropía deben generarse después de un inicio de sesión exitoso.
- **No cree sus propias soluciones personalizadas para generar tokens** de sesión. Es mejor confiar en bibliotecas y métodos bien conocidos.

4.8 Identification and authentication failures

Se refiere a protección insuficiente y problemas relaciones con errores de código e infraestructura. Se base en suposiciones relaciones con actualizaciones de software, datos críticos y canalizaciones de CI/CD que se implementan sin verificación.

Hay varias aplicaciones con funcionalidad de actualización automática, que descargan las actualizaciones sin una verificación suficiente de la aplicación de confianza.


¿Cómo prevenir?

Siempre debe asegurarse de que el software o los datos que usa provienen de la fuente confiable que espera y no se ha modificado. Esto se puede lograr implementando firmas digitales o usando técnicas similares.

Verificar las bibliotecas y las dependencias, como **npm** o **maven**, utilicen repositorio de confianza.

Verificar que los componentes no concatenan vulnerabilidades conocidas utilizando una herramienta de suministros, como **OWASP dependency check** u **Owasp cycloneDX**.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 32 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 33 de 40	

Asegurarse de que se establezca un proceso de revisión de cambios de código y configuración.

Utilizar verificaciones de integridad, como el uso de una firma digital, para detectar manipulación o reproducir ataques de los datos serializados y para garantizar que los datos serializados sin firmar o sin cifrar no se envíen a clientes que no sean de confianza.

4.9 Security Login and Monitoring Failures


Examinar los registros de la aplicación de manera oportuna, o incluso implementar protocolos de registro en primer lugar. La falla al detectar actividad maliciosa en tiempo real a menudo es aprovechada por los atacantes para lograr sus objetivos.

Se debe registrar registra eventos como inicio de sesión, inicios de sesión fallidos, transacciones que actualizan información personal o financiera y otras actividades similares de alto riesgo.

¿Cómo prevenir?

- Asegúrese de que su aplicación **registre todas las acciones de inicio de sesión**, las fallas de control de acceso y las fallas de validación de entrada del lado del servidor.
- **Se debe almacenar registros en otra localidad diferente que solo en el sistema local** es unos síntomas de problemas, ya que los atacantes pueden modificar o eliminar registros almacenadas localmente durante una violación del sistema.
- **Estos registros deben tener suficiente detalles** y contexto para permitir que los analistas identifiquen fácilmente conductas o cuentas sospechosas.
- Sus registros **deben ser fácilmente analizados por un sistema de administración de registro** por separado del sistema en cuestión. Estos registros deben conservarse durante el tiempo suficiente para permitir un examen forense retrasado o un análisis posterior.
- **Implemente un sistema de monitoreo y alerta para garantizar que las actividades sospechosas se detecten y responda de manera**

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 33 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 34 de 40	

oportuna, y establezca un plan de respuesta y recuperación de incidentes.

4.10 Security Login and Monitoring Failures


La falsificación de solicitudes del lado del servidor (SSRF) es una de las técnicas utilizadas en muchos ataques de ciberseguridad recientes. Los ataques SSRF están dirigidos a aplicaciones vulnerables, donde **los sistemas internos están ocultos detrás de firewall desde la red externa**.

Para este tipo de aplicaciones, **un atacante explota el servidor back-end para crear y enviar solicitudes al servidor interno**.

¿Cómo prevenir?

- Desinfecte y autorice los datos de entrada proporcionado por el cliente.
- Implemente el esquema de URL, el puerto y una lista de permitidos.
- No envíe ninguna respuesta sin procesar a los clientes.
- Deshabilite las redirecciones HTTP.
- Garantice las consistencias de URL para evitar ataques como el reenlace de DNS y las condiciones de carrera.
- Deshabilitar esquemas de URL no utilizados prohibirá a los atacantes usar su aplicación web para realizar solicitudes utilizando esquemas potencialmente peligrosos como file, ftp, entre otros.

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 34 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 35 de 40	

5. WAF

Toda aplicación web debe ser flexible para el manejo de herramientas de protección como es el WAF (Web Application Firewall), el WAF está compuesto por al menos las siguientes firmas de protección:

- StartURL.
- Denny URL
- Cookie Consistency
- Buffer Overflow
- Credit Card
- Safe Object
- Form Field Consistency
- Field Formats
- HTML Cross-Site Scripting
- HTML SQL Injection

La configuración para las aplicaciones web se basa en tres pasos fundamentales.

1. Crear Perfil
2. Crear Políticas
3. Enlazar o vincular las políticas de manera global o por cada servidor virtual.

El F5 Application Firewall protege las aplicaciones Web contra los diferentes ataques de la capa de aplicaciones web, capaz de proteger los servidores Web sin degradar los rendimientos o los tiempos de respuestas de las aplicaciones; evitando cualquier tipo de intrusión o ejecución de códigos remotos.


5.1. Canal de Comunicación

La comunicación entre el canal electrónico y la institución deberá estar encriptado de acordes con los estándares internacionales vigentes

5.2. Mensajería para clientes

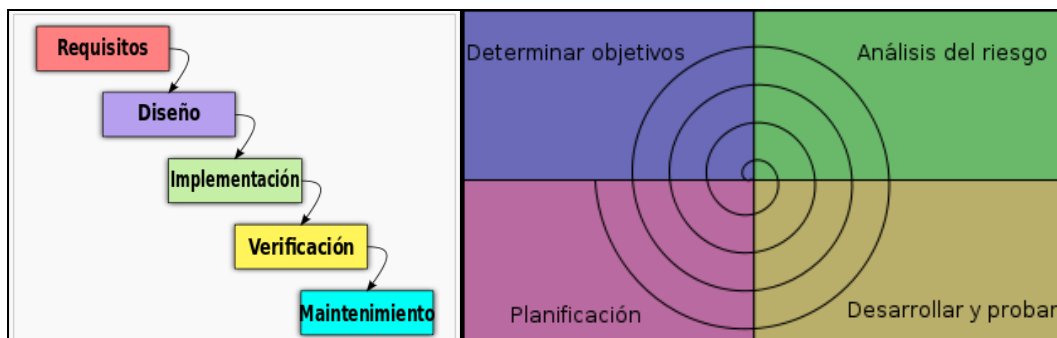
Se debe enviar notificación a los clientes a través de mensajería móvil o correo electrónico cuando el cliente ingrese y realice transacciones en los canales electrónicos

<i>Fecha de Elaboración (dd/mm/aa) 2018</i>	<i>Aprobado por (Area) Oficial de Seguridad Información</i>	<i>Fecha de Actualización (dd/mm/aa) 02/28/2023</i>	<i>Aprobado por (Cargo) Gerente de Desarrollo</i>	<i>Página 35 de 40</i>
---	---	---	---	----------------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 36 de 40	


6. PRINCIPIOS DE DISEÑO

Todo código debe cumplir con las recomendaciones de programación segura definido por las mejores prácticas (referencia: www.owasp.org para aplicaciones web, securecoding.cert.org), desarrollo en cascada, desarrollo en espiral, etc. En la que se definan las etapas del proceso para el desarrollo de software.



- **Análisis de Requisitos**
Es la fase donde se analizan las necesidades de los usuarios finales del software para determinar los objetivos que se deben cumplir.
- **Diseño de Sistema**
Organiza el sistema en elementos que puedan elaborarse por separado.
- **Diseño del Programa**
Se deben realizar algoritmos necesarios para el cumplimiento de los requerimientos del usuario.
- **Codificación**
Implementación del código fuente.
- **Pruebas**
Pruebas de la funcionalidad correcta y el cumplimiento.
- **Verificación**
El usuario final debe ejecutar el sistema.
- **Mantenimiento**
Correcciones o necesidades nuevas del software.
- **Menor Privilegio**
Este principio establece que un sujeto sólo debe dar a un objeto los privilegios que necesita para completar sus tareas asignadas.
- **Economía y simplificación de mecanismos de seguridad**

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 36 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 37 de 40	

Este principio establece que los mecanismos de seguridad que se establezcan deben ser tan sencillos como sea posible.

- **Configuraciones por defecto seguras**

Este principio comenta que a menos que un sujeto haya otorgado acceso explícito a un objeto, éste no debería tenerlo. Es decir, todo lo que no está estrictamente permitido es prohibido.

- **Mediación completa**

Este principio afirma que todos los accesos a un objeto(s) deben ser verificados para asegurarse de que cuentan con el permiso para hacerlo.

- **Diseño Abierto**

Este principio establece que la seguridad de un mecanismo no debería depender del secreto o confidencialidad de su diseño o implementación.

- **Privilegios Condicionados**

Este principio dice que se deben mantener los privilegios necesarios en diferentes momentos, en diferentes rutinas o programas. Es decir, los privilegios no deben ser estáticos para los programas o rutinas en el tiempo y en ejecución.

- **Menor mecanismo común**

Este principio comenta que deben existir el menor número de recursos compartidos entre sujetos u objetos.

- **Aceptación psicológica**


Este principio comenta que el mecanismo de seguridad que se establezca para un objeto no debe sugerir mayor dificultad a la que si el mecanismo no estuviese presente. En otras palabras, el mecanismo de seguridad deber ser fácil de usar.

7. FUENTES DE VULNERABILIDADES EN EL SOFTWARE QUE SE DEBEN CONSIDERAR.

- **Cambios en el ambiente de ejecución**

Los parches, los cambios en la configuración y variables de entorno alrededor de las aplicaciones son elementos críticos para mantener una ejecución adecuada y controlada de las rutinas y acciones previstas en el software. Al descuidar este aspecto, es probable involucrar efectos de borde o condiciones de excepción no previstas que comprometan no solamente un módulo de la aplicación sino el sistema de información mismo.

<i>Fecha de Elaboración (dd/mm/aa) 2018</i>	<i>Aprobado por (Area) Oficial de Seguridad Información</i>	<i>Fecha de Actualización (dd/mm/aa) 02/28/2023</i>	<i>Aprobado por (Cargo) Gerente de Desarrollo</i>	<i>Página 37 de 40</i>
---	---	---	---	----------------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 38 de 40	

- **Desbordamientos y chequeos de sintaxis**

Dos elementos importantes en la revisión y evaluación de software. Por un lado, la evaluación de los desbordamientos bien sea de memoria o de variables específicas dentro de un programa y por otro lado, la verificación de buen uso de los comandos o palabras reservadas en el lenguaje de programación, que permitan al programador un uso adecuado y eficiente de las estructuras. Si este aspecto no se considera con el rigor necesario, se estará comprometiendo la integridad del ambiente de ejecución de la aplicación.

- **Convenientes pero peligrosas características del diseño del software**

Esta fuente de vulnerabilidad nos presenta funcionalidades que son deseables en el software para aumentar la versatilidad de uso de las aplicaciones. Entre estas tenemos herramientas de depuración o debugging, conexiones remotas en puertos especiales, entre otras, las cuales ofrecen importantes elementos a los programadores y usuarios, pero que generalmente abren posibilidades de ingresos no autorizados que comprometen la integridad de sistemas y socavan la confianza del usuario frente a la aplicación.


- **Invocaciones no controladas**

En este aparte hacemos referencia a un inadecuado manejo de errores o excepciones en las aplicaciones o exceso de privilegios de ejecución, los cuales se manifiestan en comportamientos inesperados del software que generalmente ofrecen mayores privilegios o accesos adicionales a la información del sistema. En este sentido, el control adecuado de interrupciones, mensajes de error y entorno de ejecución de los programas se vuelve crítico al ser éstos elementos los que definen la interacción del software con el usuario final y su relación con el entorno de ejecución.

- **Bypass a bajo nivel**

Las implicaciones de esta fuente de vulnerabilidades hacen referencia al aseguramiento que la aplicación debe tener al ser invocada o ejecutada en un ambiente computacional seguro. El programador debe fortalecer y asegurar una manera autorizada de ingreso a la aplicación por parte del usuario, estableciendo mecanismos de monitoreo y control que velen porque esto se cumpla. El sobrepasar un control de acceso a un objeto,

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 38 de 80
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC.EST.1.23	Revisión: v. 5 - 2018
	Página 39 de 40	

bien sea a través de permisos deficientemente otorgados, artificios que interrumpen la normal ejecución (contraseñas de BIOS) o por la manipulación de la memoria de ejecución de la aplicación constituye un atentado directo contra la confiabilidad e integridad del software.

- **Fallas en la implementación de protocolos**

Los elementos de seguridad mencionados en este apartado, hacen referencia a las medidas de seguridad en redes. Si bien, los protocolos utilizados para transmisión y control de datos, presentan múltiples fallas, éstas con frecuencia no son consideradas dentro del proceso de implementación de una aplicación. En este sentido, sabemos que las aplicaciones que se ejecutan sobre TCP/IP tienen inherentes las fallas de éste conjunto de protocolos, por tanto, es menester del programador establecer junto con el encargado de seguridad de la información, analizar los posibles requerimientos de seguridad necesarios para que la aplicación funcione sobre un ambiente de red que brinde mayores niveles de seguridad y control de tráfico.


- **Fallas en software de base**

Todas las aplicaciones finalmente se ejecutan bajo la supervisión de un software de base o sistema operacional. Generalmente cuando se desarrollan aplicaciones, las condiciones o aseguramiento del software de base, no es condición para la adecuada ejecución de aplicaciones. Nada ganamos con aplicar y efectuar un amplio espectro de pruebas y controles, cuando el ambiente de ejecución o el software base no ha pasado por una valoración y afinamiento necesario para asegurar un ambiente de ejecución estable y seguro. En este punto, se llama la atención tanto a proveedores como a programadores, donde el trabajo conjunto debe ser una constante para incrementar los niveles de seguridad y disminuir las vulnerabilidades frecuentes inherentes al arte y la ciencia de programar.

- **Uso de componentes con vulnerabilidades conocidas**

En el proceso de diseño y desarrollo de aplicaciones no se deben considerar versiones de componentes, librerías, DLLs, entre otros recursos que contengan vulnerabilidades la cuales estuvieren publicadas en fuentes abiertas como CVE “Common Vulnerabilities and Exposures” (<https://cve.mitre.org/>)

Fecha de Elaboración (dd/mm/aa) 2018	Aprobado por (Area) Oficial de Seguridad Información	Fecha de Actualización (dd/mm/aa) 02/28/2023	Aprobado por (Cargo) Gerente de Desarrollo	Página 39 de 40
--	---	--	--	--------------------

 Banco Bolivariano	Tipo de Documento: ESTÁNDAR	
	Propietario: SEGURIDAD DE LA INFORMACIÓN	
Título: ESTÁNDAR DE SEGURIDAD PARA APLICACIONES	Código: PC. EST.1.23	Revisión: v. 5 - 2018
	Página 40 de 40	

6 CONTROL DE VERSIONES

Fecha	Revisión	Observaciones
9-sep-2010	Esquema de seguridad	Creación del Estándar de seguridad
28-02-2023	Esquema de seguridad	Creación del Estándar de seguridad

CONFIDENCIAL

<i>Fecha de Elaboración (dd/mm/aa) 2018</i>	<i>Aprobado por (Area) Oficial de Seguridad Información</i>	<i>Fecha de Actualización (dd/mm/aa) 02/28/2023</i>	<i>Aprobado por (Cargo) Gerente de Desarrollo</i>	<i>Página 40 de 40</i>
---	---	---	---	----------------------------