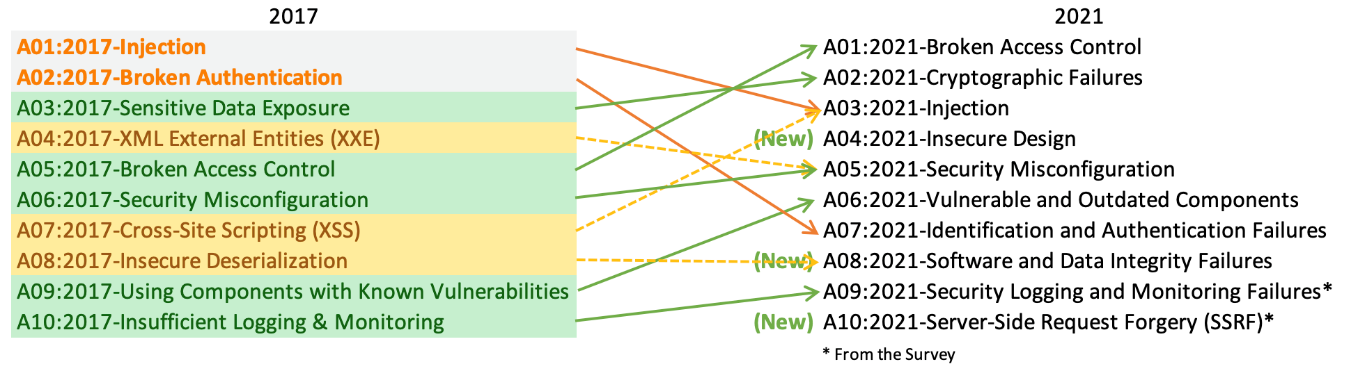
**Top 10 owasp**

Hay tres categorías nuevas, cuatro categorías con cambios de nomenclatura y alcance, y cierta consolidación en el Top 10 para 2021.

***#NOMENCLATURA conjunto de términos o palabras propias utilizadas en una ciencia, técnica, o especialidad, o por un autor.***

****

**A01: 2021-Broken Access Control** Pasando de la quinta posición, el 94% de las aplicaciones se probaron para detectar algún tipo de control de acceso defectuoso. Los CWE notables incluidos son CWE-200: Exposición de información sensible a un actor no autorizado , CWE-201: Exposición de información sensible a través de datos enviados y CWE-352: Falsificación de solicitudes entre sitios.

***#CWE (Common Weakness Enumeration) es una lista de tipos de debilidades de software dirigida a desarrolladores y profesionales de la seguridad. Se puede ver como un catálogo de debilidades documentadas que se suelen cometer programando, y que podría derivar en vulnerabilidades.***

El control de acceso hace cumplir la política de modo que los usuarios no pueden actuar fuera de sus permisos previstos. Las fallas generalmente conducen a la divulgación de información no autorizada, la modificación o la destrucción de todos los datos o la realización de una función comercial fuera de los límites del usuario.

**SOLUCIÓN.**

El control de acceso solo es efectivo en código del lado del servidor confiable o API sin servidor, donde el atacante no puede modificar la verificación de control de acceso o los metadatos.

* A excepción de los recursos públicos, denegar por defecto.
* Implemente mecanismos de control de acceso una vez y reutilícelos en toda la aplicación, incluida la minimización del uso de CORS.
* Los controles de acceso del modelo deben imponer la propiedad de los registros en lugar de aceptar que el usuario pueda crear, leer, actualizar o eliminar cualquier registro.
* Los modelos de dominio deben hacer cumplir los requisitos de límite de negocio de aplicaciones únicas.
* Desactive la lista de directorios del servidor web y asegúrese de que los metadatos del archivo (por ejemplo, .git) y los archivos de respaldo no estén presentes en las raíces web.
* Registre las fallas de control de acceso, avise a los administradores cuando sea apropiado (por ejemplo, fallas repetidas).
* Limite el acceso a la API y al controlador para minimizar el daño de las herramientas de ataque automatizadas.
* Los tokens JWT deben invalidarse en el servidor después de cerrar la sesión.

La página cuenta con la falencia de administrar datos a vista de todo público, sobre todo mostrando correos, que podrían usar para otros fines.



**A02: 2021-Fallos criptográficos** sube una posición al # 2, anteriormente conocido como Exposición de datos sensibles, que era un síntoma amplio en lugar de una causa raíz. El enfoque renovado aquí está en las fallas relacionadas con la criptografía que a menudo conduce a la exposición de datos confidenciales o al compromiso del sistema.

**SOLUCIÓN.**

Haga lo siguiente, como mínimo, y consulte las referencias:

* Clasifique los datos procesados, almacenados o transmitidos por una aplicación. Identifique qué datos son confidenciales de acuerdo con las leyes de privacidad, los requisitos reglamentarios o las necesidades comerciales.
* Aplicar controles según la clasificación.
* No almacene datos sensibles innecesariamente. Deséchelo lo antes posible o utilice la tokenización compatible con PCI DSS o incluso el truncamiento. Los datos que no se conservan no se pueden robar.
* Asegúrese de cifrar todos los datos confidenciales en reposo.
* Garantizar la implementación de algoritmos, protocolos y claves estándar sólidos y actualizados; Utilice una gestión de claves adecuada.
* Cifre todos los datos en tránsito con protocolos seguros como TLS con cifrado de secreto directo perfecto (PFS), priorización de cifrado por parte del servidor y parámetros seguros. Aplique el cifrado mediante directivas como HTTP Strict Transport Security (HSTS).
* Deshabilite el almacenamiento en caché para respuestas que contengan datos confidenciales.
* Almacene las contraseñas utilizando fuertes funciones de hash adaptativas y saladas con un factor de trabajo (factor de retraso), como Argon2, scrypt, bcrypt o PBKDF2.
* Verifique de forma independiente la eficacia de la configuración y los ajustes.

**A03: 2021-** La inyección se desliza hasta la tercera posición. El 94% de las aplicaciones fueron probadas para alguna forma de inyección. Los CWE notables incluidos son CWE-79: Cross-site Scripting , CWE-89: SQL Injection y CWE-73: Control externo de nombre de archivo o ruta .

***#Cross-site Consiste en modificar valores que la aplicación web usa para pasar variables entre dos páginas. Un clásico ejemplo de esto es hacer que a través de un buscador se ejecute un mensaje de alerta en JavaScript. Con XSS reflejado, el atacante podría robar las*cookies*para luego robar la identidad, pero para esto, debe lograr que su víctima ejecute un determinado comando dentro de su dirección web.***

* La aplicación no valida, filtra ni desinfecta los datos proporcionados por el usuario.
* Las consultas dinámicas o las llamadas no parametrizadas sin escape sensible al contexto se utilizan directamente en el intérprete.
* Los datos hostiles se utilizan dentro de los parámetros de búsqueda de mapeo relacional de objetos (ORM) para extraer registros confidenciales adicionales.
* Los datos hostiles se utilizan o concatenan directamente. El SQL o comando contiene la estructura y los datos maliciosos en consultas dinámicas, comandos o procedimientos almacenados.

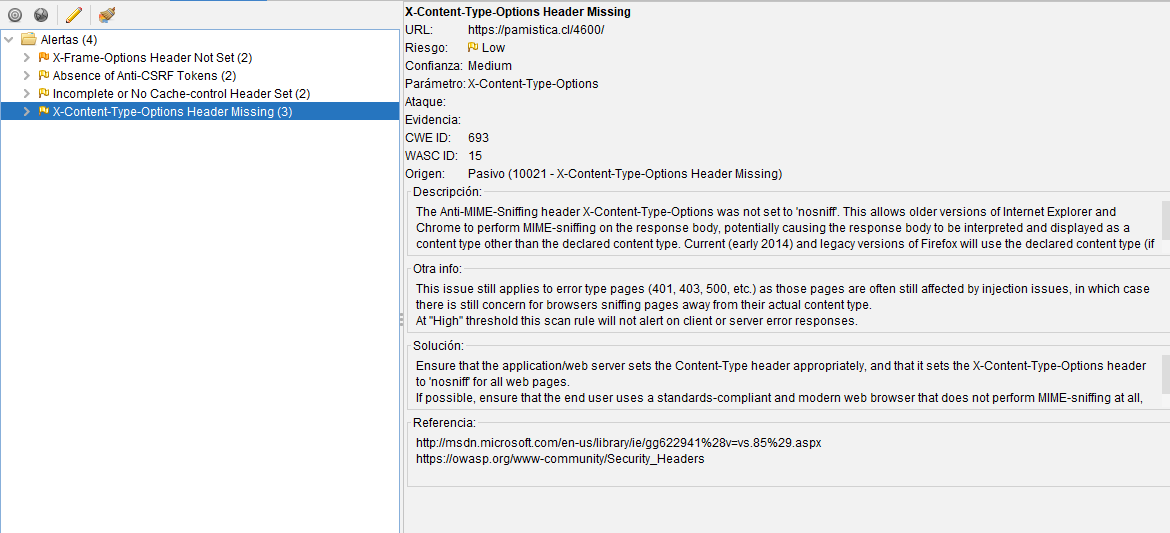
**SOLUCIÓN.**

* La prevención de la inyección requiere mantener los datos separados de los comandos y las consultas.
* La opción preferida es utilizar una API segura, que evita el uso del intérprete por completo, proporciona una interfaz parametrizada o migra a Object Relational Mapping Tools (ORM).

Nota: Incluso cuando están parametrizados, los procedimientos almacenados pueden introducir inyección SQL si PL / SQL o T-SQL concatenan consultas y datos o ejecutan datos hostiles con EXECUTE IMMEDIATE o exec ().

* Utilice la validación de entrada del lado del servidor positiva o de "lista blanca". Esta no es una defensa completa, ya que muchas aplicaciones requieren caracteres especiales, como áreas de texto o API para aplicaciones móviles.
* Para cualquier consulta dinámica residual, escape caracteres especiales utilizando la sintaxis de escape específica para ese intérprete.
* Nota: las estructuras SQL como nombres de tablas, nombres de columnas, etc. no se pueden escapar y, por lo tanto, los nombres de estructuras proporcionados por el usuario son peligrosos. Este es un problema común en el software de redacción de informes.
* Utilice LIMIT y otros controles SQL dentro de las consultas para evitar la divulgación masiva de registros en caso de inyección SQL.

Haciendo inyección, Los ataques más peligrosos podrían modificar o eliminar datos o incluso invocar procedimientos almacenados

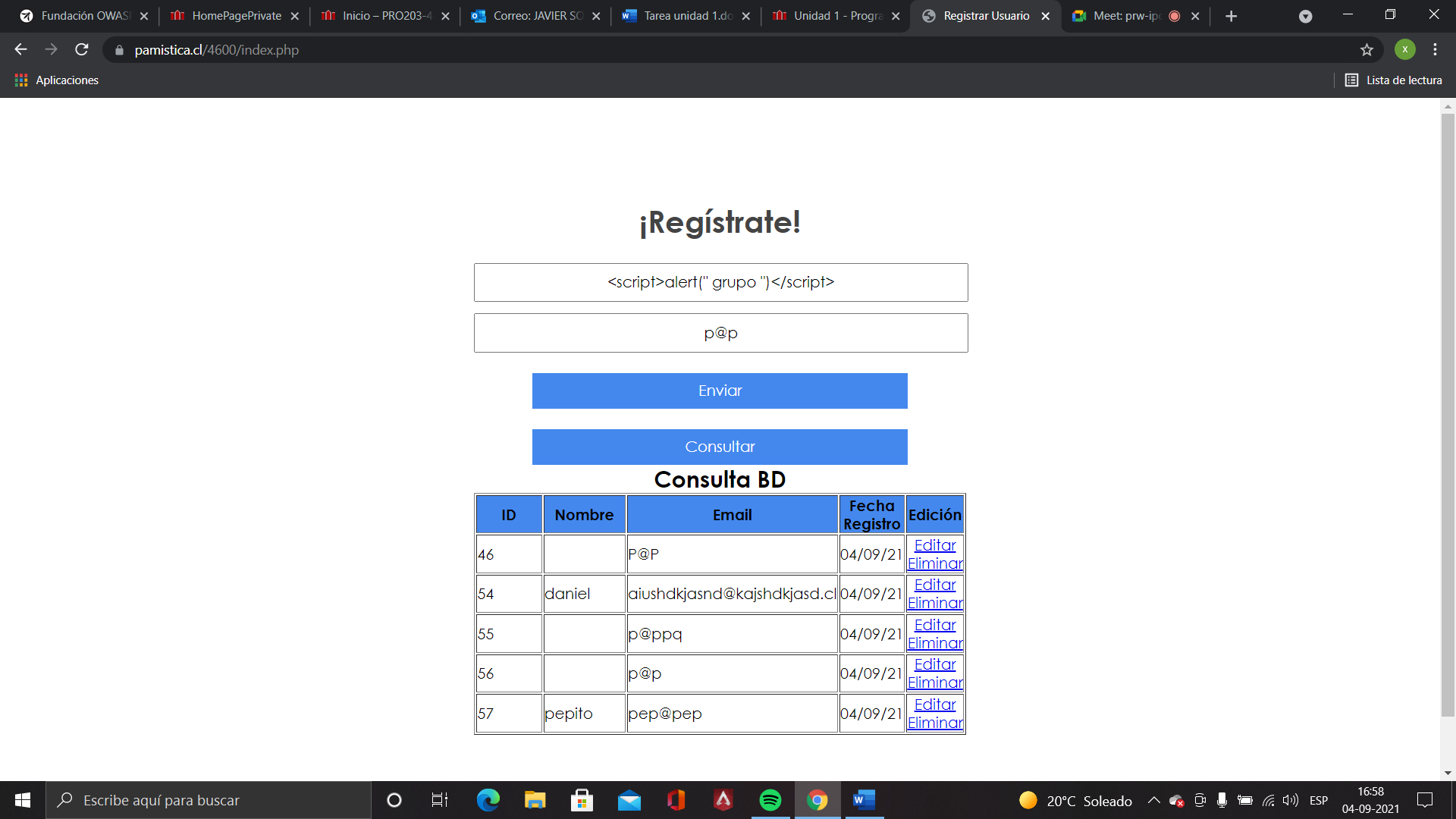


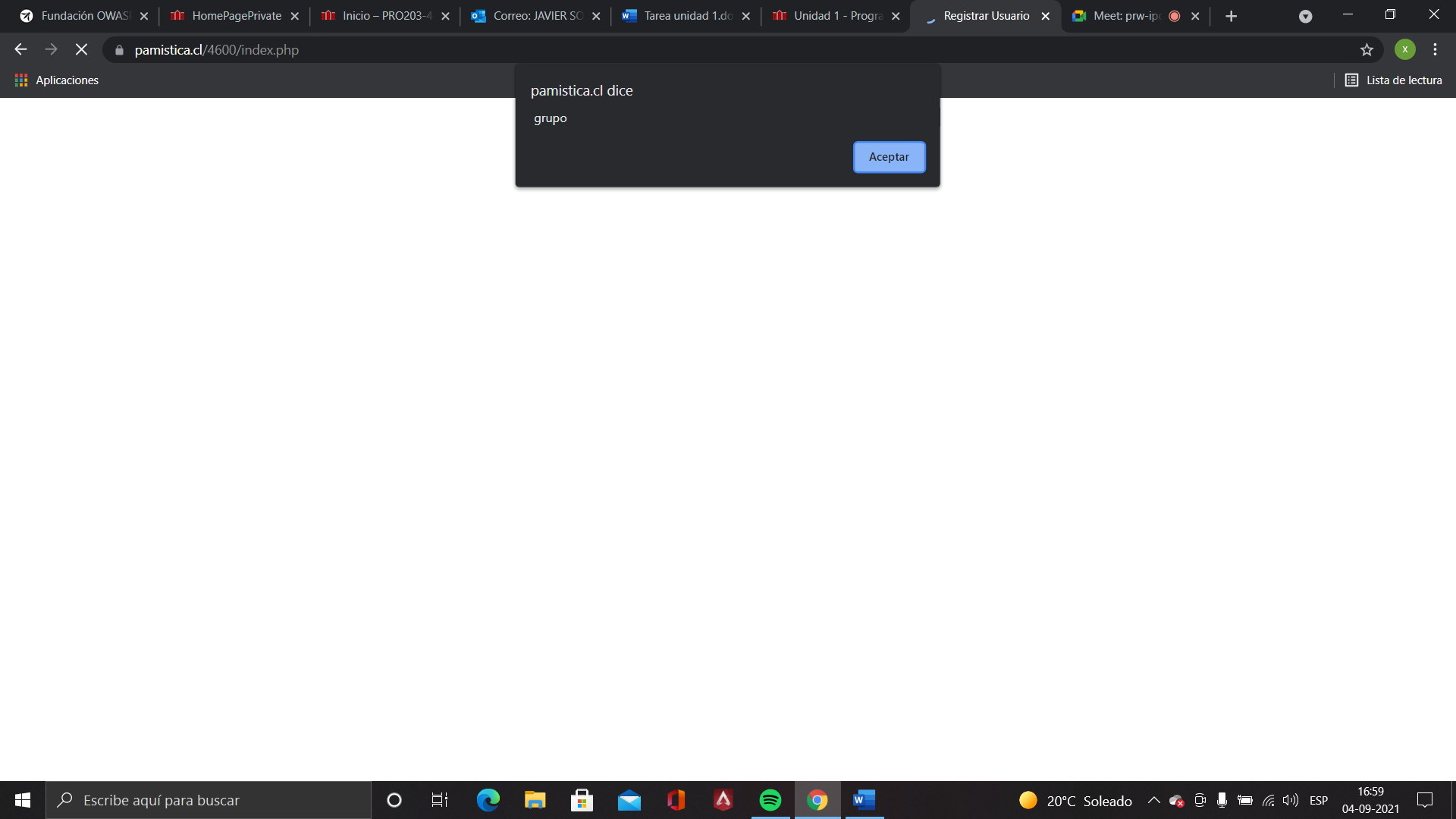
*\*resultado análisis owasp.*

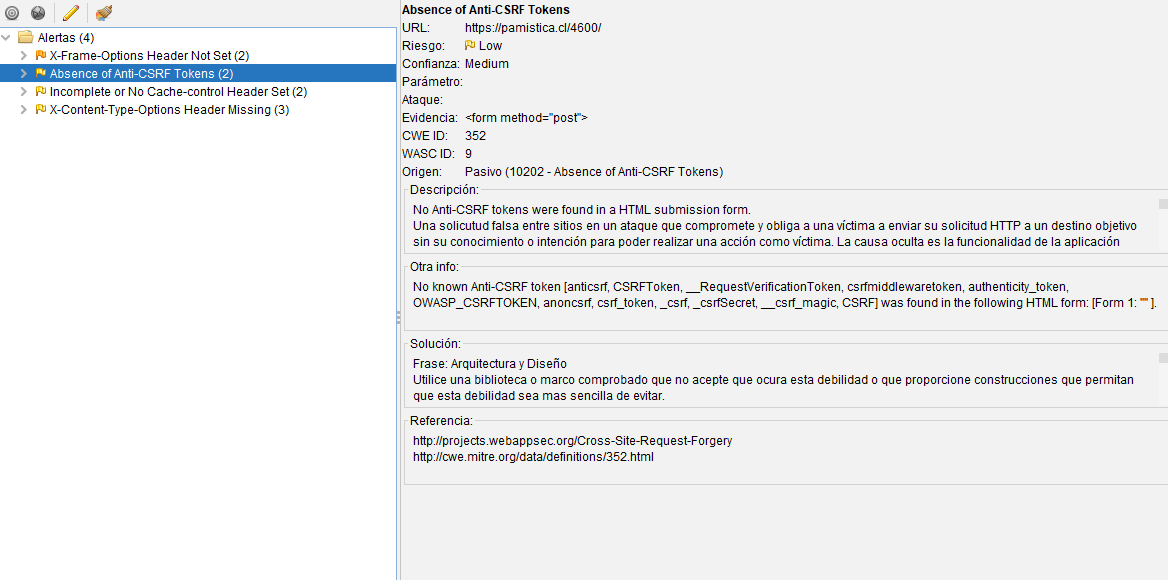
*El encabezado de respuesta HTTP 'X-Content-Type-Options' evita que el navegador MIME rastree una respuesta fuera del tipo de contenido declarado.  
El servidor no devolvió un encabezado 'X-Content-Type-Options' correcto, lo que significa que este sitio web podría estar en riesgo de un ataque de Cross-Site Scripting (XSS).*

*#El encabezado HTTP de respuesta****X-Content-Type-Options****es un marcador utilizado por el servidor para indicar que los*[*tipos MIME*](https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types)*anunciados en los encabezados*[*Content-Type*](https://developer.mozilla.org/es/docs/Web/HTTP/Headers/Content-Type)*no se deben cambiar ni seguir*

*#MIME Types (Multipurpose Internet Mail Extensions) son la manera standard de mandar contenido a través de la red.*







*\*resultado análisis owasp.*

*La técnica llamada falsificación de petición en sitios cruzados, proviene de su nombre en inglés****Cross Site Request Forgery****(CSRF o XSRF). Este ataque fuerza al navegador web de su víctima, validado en algún servicio (como por ejemplo correo o home banking) a enviar una petición a una aplicación web vulnerable.*

*Esta aplicación se encarga de realizar la acción elegida a través de la víctima, debido que la actividad maliciosa será procesada en nombre del usuario logueado. Al contrario de los ataques conocidos como Cross Site Scripting (su traducción sería ordenes en sitios cruzados – XSS) los cuales explotan la confianza del usuario para con un sitio particular; el Cross Site Request Forgery****explota la confianza****que un sitio web tiene en un usuario particular.*

**A04: 2021-Diseño inseguro** Una nueva categoría para 2021 se centra en los riesgos relacionados con el diseño y las fallas arquitectónicas, con un llamado a un mayor uso del modelado de amenazas, patrones de diseño seguros y arquitecturas de referencia. Los CWE notables incluyen CWE-209: Generación de mensaje de error que contiene información confidencial , CWE-256: Almacenamiento desprotegido de credenciales , CWE-501: Infracción de límites de confianza y CWE-522: Credenciales insuficientemente protegidas .

Por ejemplo, imagine un código que debería cifrar datos confidenciales, pero no existe un método. El diseño inseguro ineficaz es donde se puede realizar una amenaza, pero una validación de la lógica de dominio (empresarial) insuficiente impide la acción. Por ejemplo, imagine una lógica de dominio que se supone procesa la desgravación fiscal pandémica en función de los tramos de ingresos, pero que no valida que todas las entradas estén correctamente firmadas y proporcione un beneficio de desgravación mucho más significativo de lo que debería concederse.

**SOLUCIÓN.**

* Establecer y utilizar un ciclo de vida de desarrollo seguro con los profesionales de AppSec para ayudar a evaluar y diseñar controles relacionados con la seguridad y la privacidad.
* Establecer y utilizar una biblioteca de patrones de diseño seguros o componentes de carreteras pavimentadas listas para usar
* Utilice el modelado de amenazas para la autenticación crítica, el control de acceso, la lógica empresarial y los flujos de claves.
* Escriba pruebas unitarias y de integración para validar que todos los flujos críticos sean resistentes al modelo de amenaza.

**A05: 2021-La configuración incorrecta de seguridad** Pasando del puesto 6 en la edición anterior, el 90% de las aplicaciones se probaron para detectar algún tipo de configuración incorrecta. Con más cambios en software altamente configurable, no es sorprendente ver que esta categoría asciende. Los CWE notables incluidos son Configuración CWE-16 y CWE-611 Restricción incorrecta de la referencia de entidad externa XML .

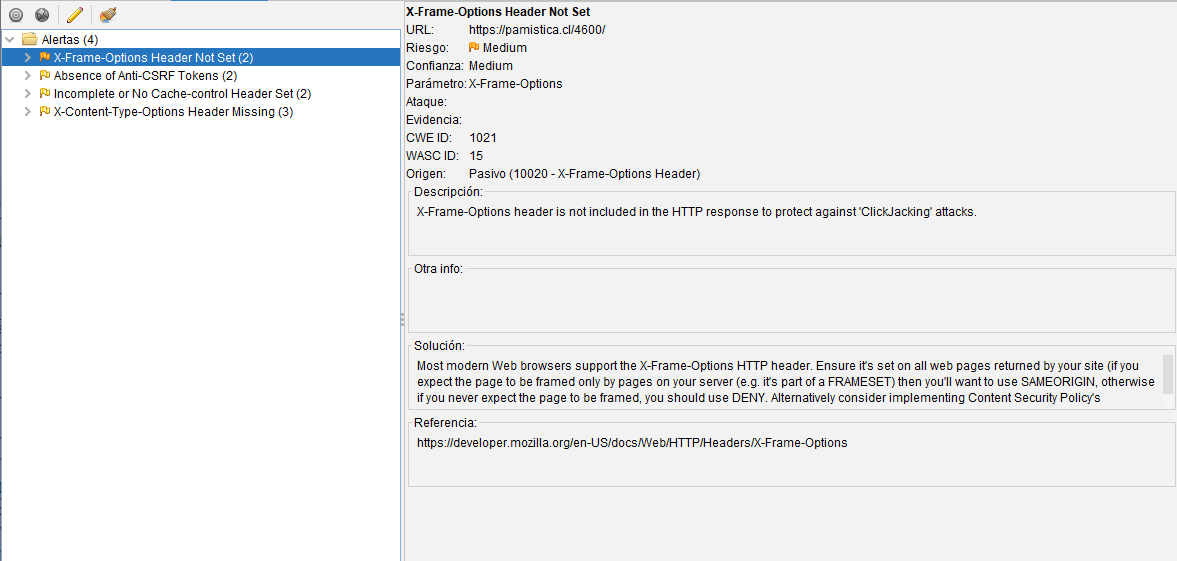
La aplicación puede ser vulnerable si la aplicación es:

* Falta el refuerzo de seguridad adecuado en cualquier parte de la pila de aplicaciones o permisos configurados incorrectamente en los servicios en la nube.
* Se habilitan o instalan funciones innecesarias (por ejemplo, puertos, servicios, páginas, cuentas o privilegios innecesarios).
* Las cuentas predeterminadas y sus contraseñas aún están habilitadas y sin cambios.
* El manejo de errores revela a los usuarios rastros de pila u otros mensajes de error demasiado informativos.
* Para los sistemas actualizados, las últimas funciones de seguridad están deshabilitadas o no configuradas de forma segura.
* Las configuraciones de seguridad en los servidores de aplicaciones, marcos de aplicaciones (por ejemplo, Struts, Spring, ASP.NET), bibliotecas, bases de datos, etc., no están configuradas para valores seguros.
* El servidor no envía encabezados o directivas de seguridad, o no están configurados para valores seguros.
* El software está desactualizado o es vulnerable (consulte A06: 2021-Componentes vulnerables y desactualizados).

**SOLUCIÓN.**

Se deben implementar procesos de instalación seguros, que incluyen:

* Un proceso de endurecimiento repetible agiliza y facilita la implementación de otro entorno que esté adecuadamente bloqueado. Los entornos de desarrollo, control de calidad y producción deben configurarse de forma idéntica, con diferentes credenciales utilizadas en cada entorno. Este proceso debe automatizarse para minimizar el esfuerzo necesario para configurar un nuevo entorno seguro.
* Una plataforma mínima sin funciones, componentes, documentación ni ejemplos innecesarios. Elimine o no instale funciones y marcos no utilizados.
* Una tarea para revisar y actualizar las configuraciones apropiadas para todas las notas de seguridad, actualizaciones y parches como parte del proceso de administración de parches (consulte A06: 2021-Componentes vulnerables y obsoletos). Revise los permisos de almacenamiento en la nube (p. Ej., Permisos de depósito de S3).
* Una arquitectura de aplicación segmentada proporciona una separación efectiva y segura entre componentes o inquilinos, con segmentación, contenedorización o grupos de seguridad en la nube (ACL).
* Envío de directivas de seguridad a los clientes, por ejemplo, encabezados de seguridad.
* Un proceso automatizado para verificar la efectividad de las configuraciones y ajustes en todos los entornos.



***El encabezado X-Frame-Options es una medida de seguridad que evita que la consola web Qlik NPrinting y NewsStand se incrusten en un <frame> o <iframe>. Habilitar los encabezados de respuesta HTTP de X-Frame-Options protege contra Cross-Frame Scripting (XFS), clickjacking y otras formas de ataque.***

***El encabezado de respuesta***[***HTTP***](https://developer.mozilla.org/en-US/docs/Web/HTTP)***X-Frame-Options puede ser usado para indicar si debería permitir un navegador renderizar una página en un***[***<frame>***](https://developer.mozilla.org/es/docs/Web/HTML/Element/frame)***,***[***<iframe>***](https://developer.mozilla.org/es/docs/Web/HTML/Element/iframe)***o***[***<object>***](https://developer.mozilla.org/es/docs/Web/HTML/Element/object)***. Las páginas webs pueden usarlo para evitar ataques de*** [***clickjacking***](https://es.wikipedia.org/wiki/Clickjacking)***, asegurándose que su contenido no es embebido en otros sitios.***

*#****Clickjacking,*** *también conocido como "ataque de compensación de UI", es cuando un atacante usa varias capas transparentes u opacas para engañar a un usuario para que haga clic en un botón o enlace en otra página cuando intenta hacer clic en la página del nivel superior. Por lo tanto, el atacante está "secuestrando" los clics destinados a su página y enrutando a otra página, muy probablemente propiedad de otra aplicación, dominio o ambos. Tipo pop up.*

**A06: 2021-Componentes vulnerables y obsoletos** Los componentes vulnerables son un problema conocido que nos cuesta probar y evaluar el riesgo y es la única categoría que no tiene ningún CVE asignado a los CWE incluidos, por lo que se utiliza un peso de impacto / exploits predeterminado de 5.0. Los CWE notables incluidos son CWE-1104: Uso de componentes de terceros no mantenidos y los dos CWE del Top 10 2013 y 2017.

Probablemente seas vulnerable:

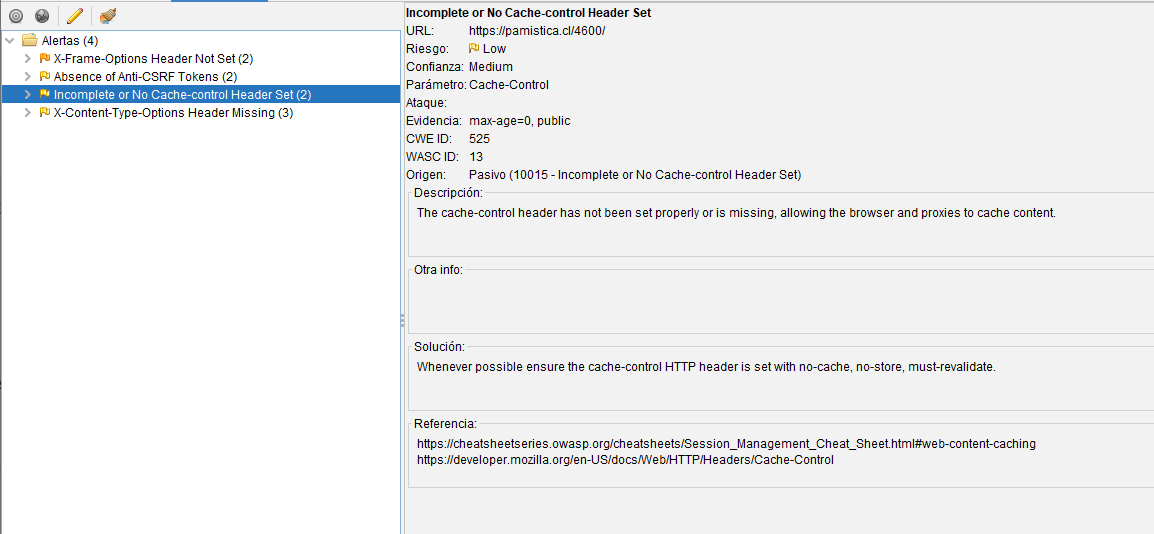
* Si no conoce las versiones de todos los componentes que utiliza (tanto del lado del cliente como del lado del servidor). Esto incluye los componentes que usa directamente, así como las dependencias anidadas.
* Si el software es vulnerable, no es compatible o no está actualizado. Esto incluye el sistema operativo, el servidor web / de aplicaciones, el sistema de administración de bases de datos (DBMS), las aplicaciones, las API y todos los componentes, los entornos de ejecución y las bibliotecas.
* Si no busca vulnerabilidades con regularidad y se suscribe a los boletines de seguridad relacionados con los componentes que utiliza.
* Si no repara o actualiza la plataforma subyacente, los marcos y las dependencias de manera oportuna y basada en el riesgo. Esto suele ocurrir en entornos en los que la aplicación de parches es una tarea mensual o trimestral bajo control de cambios, lo que deja a las organizaciones abiertas a días o meses de exposición innecesaria a vulnerabilidades reparadas.
* Si los desarrolladores de software no prueban la compatibilidad de las bibliotecas actualizadas, actualizadas o parcheadas.
* Si no asegura las configuraciones de los componentes (consulte A05: 2021-Configuración incorrecta de seguridad).

**SOLUCIÓN.**

Debe existir un proceso de administración de parches para:

* Elimine las dependencias no utilizadas, las funciones, los componentes, los archivos y la documentación innecesarios.
* Realice un inventario continuo de las versiones de los componentes del lado del cliente y del lado del servidor (por ejemplo, marcos, bibliotecas) y sus dependencias utilizando herramientas como versiones, OWASP Dependency Check, retire.js, etc. componentes. Utilice herramientas de análisis de composición de software para automatizar el proceso. Suscríbase para recibir alertas por correo electrónico sobre vulnerabilidades de seguridad relacionadas con los componentes que utiliza.
* Solo obtenga componentes de fuentes oficiales a través de enlaces seguros. Prefiera los paquetes firmados para reducir la posibilidad de incluir un componente malicioso modificado (consulte A08: 2021-Fallos de integridad de datos y software).
* Supervise las bibliotecas y los componentes que no se mantienen o no crean parches de seguridad para versiones anteriores. Si no es posible aplicar un parche, considere implementar un parche virtual para monitorear, detectar o protegerse contra el problema descubierto.

Cada organización debe garantizar un plan continuo para monitorear, clasificar y aplicar actualizaciones o cambios de configuración durante la vida útil de la aplicación o cartera.

*\*resultado análisis owasp.*

***El encabezado de control de caché no se ha configurado correctamente o falta, lo que permite que el navegador y los proxies almacenen contenido en caché.***

**A07: 2021-Identification and Authentication Failures** anteriormente era una autenticación rota y se está deslizando hacia abajo desde la segunda posición, y ahora incluye CWE que están más relacionados con fallas de identificación. Esta categoría sigue siendo una parte integral del Top 10, pero la mayor disponibilidad de marcos estandarizados parece estar ayudando. La confirmación de la identidad, la autenticación y la administración de sesiones del usuario es fundamental para proteger contra los ataques relacionados con la autenticación. Si la aplicación:

* Permite ataques automatizados como el relleno de credenciales, donde el atacante tiene una lista de nombres de usuario y contraseñas válidos.
* Permite la fuerza bruta u otros ataques automatizados.
* Permite contraseñas predeterminadas, débiles o conocidas, como "Contraseña1" o "admin / admin".
* Utiliza procesos de recuperación de credenciales débiles o ineficaces y de olvido de contraseñas, como "respuestas basadas en el conocimiento", que no pueden protegerse.
* Utiliza contraseñas de texto sin formato, cifradas o con un hash débil (consulte A3: Exposición de datos confidenciales de 2017).
* Tiene autenticación multifactor falta o ineficaz.
* Expone los ID de sesión en la URL (por ejemplo, reescritura de URL).
* No gire los ID de sesión después de iniciar sesión correctamente.
* No invalida correctamente los ID de sesión. Las sesiones de usuario o los tokens de autenticación (principalmente los tokens de inicio de sesión único (SSO)) no se invalidan correctamente durante el cierre de sesión o un período de inactividad.

**SOLUCIÓN.**

* Siempre que sea posible, implemente la autenticación multifactor para evitar el relleno automatizado de credenciales, la fuerza bruta y los ataques de reutilización de credenciales robadas.
* No envíe ni implemente con credenciales predeterminadas, especialmente para usuarios administradores.
* Implemente comprobaciones de contraseñas débiles, como probar contraseñas nuevas o cambiadas en la lista de las 10.000 peores contraseñas.
* Alinee las políticas de longitud, complejidad y rotación de contraseñas con las pautas de NIST 800-63b en la sección 5.1.1 para Secretos memorizados u otras políticas de contraseñas modernas basadas en evidencia.
* Asegúrese de que el registro, la recuperación de credenciales y las rutas de API estén reforzadas contra los ataques de enumeración de cuentas mediante el uso de los mismos mensajes para todos los resultados.
* Limite o retrase cada vez más los intentos fallidos de inicio de sesión. Registre todas las fallas y avise a los administradores cuando se detecten ataques de relleno de credenciales, fuerza bruta u otros ataques.
* Utilice un administrador de sesiones integrado, seguro y del lado del servidor que genere una nueva ID de sesión aleatoria con alta entropía después de iniciar sesión. Los ID de sesión no deben estar en la URL, almacenarse de forma segura e invalidarse después del cierre de sesión, inactivo y tiempos de espera absolutos.

**A08: 2021-Software and Data Integrity Failures** Una nueva categoría para 2021 se enfoca en hacer suposiciones relacionadas con actualizaciones de software, datos críticos y canalizaciones de CI / CD sin verificar la integridad. Uno de los impactos más ponderados de los datos CVE / CVSS. Los CWE notables incluyen CWE-502: deserialización de datos que no son de confianza , CWE-829: inclusión de funcionalidad de una esfera de control que no es de confianza y CWE-494: descarga de código sin verificación de integridad. \*Las fallas en la integridad del software y los datos se relacionan con el código y la infraestructura que no protegen contra las violaciones de la integridad. Por ejemplo, cuando los objetos o los datos se codifican en una estructura que un atacante puede ver y modificar, es vulnerable a una deserialización insegura. Otra forma de esto es cuando una aplicación se basa en complementos, bibliotecas o módulos de fuentes, repositorios y redes de entrega de contenido (CDN) que no son de confianza. Una canalización de CI / CD insegura puede presentar la posibilidad de acceso no autorizado, código malicioso o compromiso del sistema. Por último, muchas aplicaciones ahora incluyen la funcionalidad de actualización automática, donde las actualizaciones se descargan sin una verificación de integridad suficiente y se aplican a la aplicación de confianza anterior. Los atacantes podrían cargar sus propias actualizaciones para distribuirlas y ejecutarlas en todas las instalaciones.

**SOLUCIÓN.**

* Asegúrese de que los datos serializados no firmados o no cifrados no se envíen a clientes que no sean de confianza sin algún tipo de verificación de integridad o firma digital para detectar alteraciones o reproducción de los datos serializados.
* Verifique que el software o los datos provengan de la fuente esperada mediante la firma o mecanismos similares
* Asegúrese de que las bibliotecas y dependencias, como npm o Maven, consuman repositorios confiables
* Asegúrese de que se utilice una herramienta de seguridad de la cadena de suministro de software, como OWASP Dependency Check o OWASP CycloneDX, para verificar que los componentes no contengan vulnerabilidades conocidas.
* Asegúrese de que su canalización de CI / CD tenga la configuración y el control de acceso adecuados para garantizar la integridad del código que fluye a través de los procesos de construcción e implementación.

**A09: Las fallas de registro y monitoreo de seguridad de 2021** eran anteriormente Registro y monitoreo insuficientes y se agrega de la encuesta de la industria (n. ° 3), avanzando desde el n. ° 10 anterior. El registro y el monitoreo pueden ser difíciles de probar, a menudo implican entrevistas o preguntar si se detectaron ataques durante una prueba de penetración. No hay muchos datos CVE / CVSS para esta categoría, pero detectar y responder a las infracciones es fundamental. Aún así, puede ser muy impactante para la visibilidad, las alertas de incidentes y el análisis forense. Esta categoría se expande más allá de CWE-778 Registro insuficiente para incluir CWE-117 Neutralización de salida incorrecta para registros , CWE-223 Omisión de información relevante para la seguridad y CWE-532 Inserción de información confidencial en el archivo de registro . \*Sin registro y monitoreo, las infracciones no se pueden detectar. En cualquier momento se produce un registro, detección, supervisión y respuesta activa insuficientes:

* Los eventos auditables, como inicios de sesión, inicios de sesión fallidos y transacciones de alto valor, no se registran.
* Las advertencias y los errores generan mensajes de registro inexistentes, inadecuados o poco claros.
* Los registros de aplicaciones y API no se controlan para detectar actividades sospechosas.
* Los registros solo se almacenan localmente.
* Los umbrales de alerta apropiados y los procesos de escalamiento de la respuesta no están establecidos o no son efectivos.
* Las pruebas de penetración y los escaneos mediante herramientas DAST (como OWASP ZAP) no activan alertas.
* La aplicación no puede detectar, escalar ni alertar sobre ataques activos en tiempo real o casi en tiempo real.

**\*Es vulnerable a la fuga de información al hacer que los eventos de registro y alerta sean visibles para un usuario o un atacante\***

**SOLUCIÓN.**

Los desarrolladores deben implementar algunos o todos los siguientes controles, dependiendo del riesgo de la aplicación:

* Asegúrese de que todas las fallas de validación de entrada, control de acceso y del lado del servidor puedan registrarse con suficiente contexto de usuario para identificar cuentas sospechosas o maliciosas y mantenerse durante el tiempo suficiente para permitir un análisis forense retrasado.
* Asegúrese de que los registros se generen en un formato que las soluciones de administración de registros puedan consumir fácilmente.
* Asegúrese de que los datos de registro estén codificados correctamente para evitar inyecciones o ataques en los sistemas de registro o monitoreo.
* Asegúrese de que las transacciones de alto valor tengan una pista de auditoría con controles de integridad para evitar alteraciones o eliminaciones, como tablas de bases de datos de solo anexión o similares.
* Los equipos de DevSecOps deben establecer un monitoreo y alertas efectivos para que las actividades sospechosas se detecten y se responda rápidamente.
* Establezca o adopte un plan de recuperación y respuesta a incidentes, como NIST 800-61r2 o posterior.

**R10: La falsificación de solicitudes del lado del servidor 2021** se agrega de la encuesta de la industria (n. ° 1). Los datos muestran una tasa de incidencia relativamente baja con una cobertura de pruebas por encima del promedio, junto con calificaciones por encima del promedio para el potencial de Explotación e Impacto. Esta categoría representa el escenario en el que los profesionales de la industria nos dicen que esto es importante, aunque no está ilustrado en los datos en este momento. Las fallas de SSRF ocurren cuando una aplicación web está obteniendo un recurso remoto sin validar la URL proporcionada por el usuario. Permite que un atacante coaccione a la aplicación para que envíe una solicitud diseñada a un destino inesperado, incluso cuando está protegido por un firewall, VPN u otro tipo de ACL de red. Dado que las aplicaciones web modernas brindan a los usuarios finales funciones convenientes, la búsqueda de una URL se convierte en un escenario común. Como resultado, la incidencia de SSRF está aumentando. Además, la gravedad de SSRF es cada vez mayor debido a los servicios en la nube y la complejidad de las arquitecturas.

***#SSRF Esta vulnerabilidad ocurre cuando una aplicación web permite hacer consultas HTTP del lado del servidor hacia un dominio arbitrario elegido por el atacante. Esto le permite a un atacante hacer conexión con servicios de la infraestructura interna donde se aloja la web y exfiltrar información sensible.***

**SOLUCIÓN.**

**Desde la capa de red.**

* Segmentar la funcionalidad de acceso a recursos remotos en redes separadas para reducir el impacto de SSRF
* Hacer cumplir las políticas de firewall "denegar por defecto" o las reglas de control de acceso a la red para bloquear todo el tráfico de la intranet excepto el esencial

**Desde la capa de aplicación.**

* Desinfecte y valide todos los datos de entrada proporcionados por el cliente
* Haga cumplir el esquema de URL, el puerto y el destino con una lista de permitidos positiva
* No envíe respuestas sin procesar a los clientes
* Deshabilitar las redirecciones HTTP
* Tenga en cuenta la coherencia de la URL para evitar ataques como el reenlace de DNS y las condiciones de carrera de "tiempo de verificación, tiempo de uso" (TOCTOU)

No mitigue SSRF mediante el uso de una lista de denegación o una expresión regular. Los atacantes tienen listas de carga útil, herramientas y habilidades para eludir las listas de denegación.

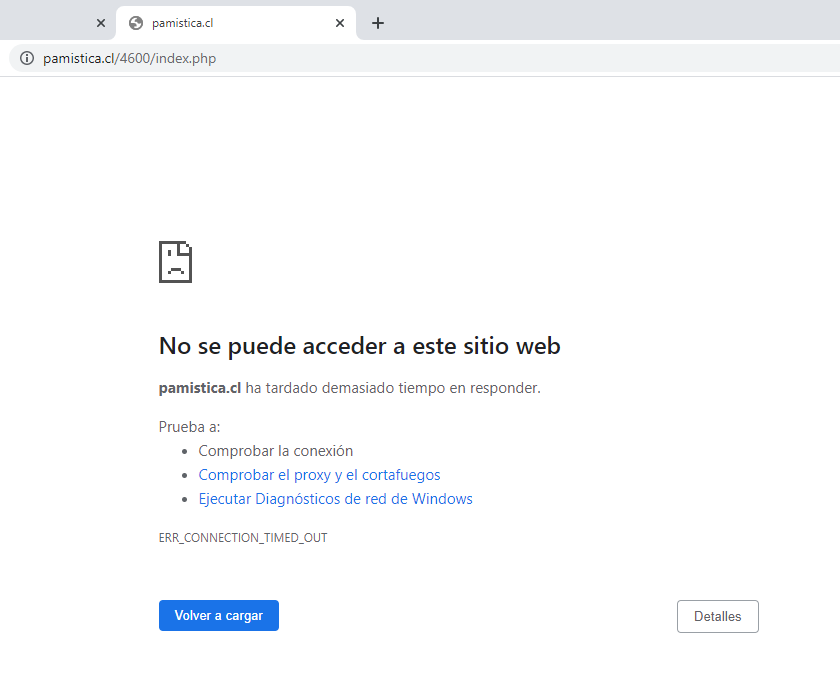
Solucion.

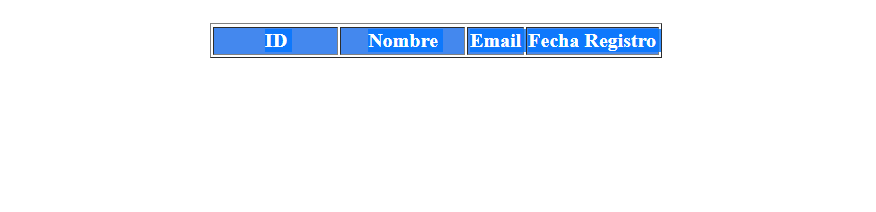
#### **Solución**

Configure su servidor web para incluir un encabezado 'X-Content-Type-Options' con un valor de 'nosniff'.

El servidor pamistica.cl bloquea IP tras recibir constantes ataques:

Ip publica bloqueada: 186.107.15.182





<script>alert(" marcianeke ")</script>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente