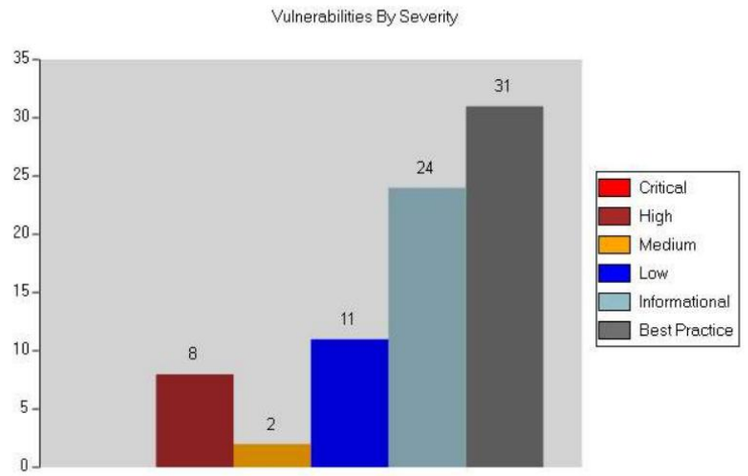
Fortalecer WebInspect

Vulnerabilidad (heredada)

Informe de evaluación de aplicaciones web

Nombre del escaneo: Flujo de trabajo: volver a probar http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec			
Política:	Estándar	Sesiones de rastreo:	855
Fecha de escaneo:	05/01/2026 17:36:19	Vulnerabilidades:	21
Versión escaneada:	24.4.0.70	Duración del escaneo:	3 horas: 19 minutos
Tipo de escaneo:	Sitio	Cliente:	Costumbre

Servidor: http://172.16.21.39:7001



Alto

Transporte inseguro

Resumen:

Cualquier área de una aplicación web que posiblemente contenga información confidencial o acceso a funcionalidades privilegiadas como la administración de sitios remotos debe utilizar SSL u otra forma de cifrado para evitar que la información de inicio de sesión sea interceptada o robada. http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec no ha cumplido con esta política.
Las recomendaciones incluyen garantizar que las áreas sensibles de su aplicación web tengan protocolos de cifrado adecuados para evitar que la información de inicio de sesión y otros datos que podrían ser útiles para un atacante sean interceptados.

Implicación:

Un atacante que aproveche esta vulnerabilidad de diseño podría utilizar la información para escalar su método de ataque, posiblemente conduciendo a la suplantación de un usuario legítimo, el robo de datos confidenciales o la ejecución de acciones no previstas por los desarrolladores de la aplicación.

Arreglar:

Para operaciones de seguridad:
Asegúrese de que las áreas sensibles de su aplicación web tengan implementados protocolos de cifrado adecuados para evitar que la información de inicio de sesión y otros datos que podrían ser útiles para un atacante sean interceptados.

Para el desarrollo:
Asegúrese de que las áreas sensibles de su aplicación web tengan implementados protocolos de cifrado adecuados para evitar que la información de inicio de sesión y otros datos que podrían ser útiles para un atacante sean interceptados.

Para control de calidad:
Pruebe la aplicación no sólo desde la perspectiva de un usuario normal, sino también desde la perspectiva de uno malintencionado.

Nombres de archivos:

- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/inicioErr.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/

Alto

A menudo mal utilizado: Iniciar sesión

Resumen:

Se ha descubierto un formulario de inicio de sesión sin cifrar. Cualquier área de una aplicación web que contenga información confidencial o acceso a funciones privilegiadas, como la administración remota de sitios, debe utilizar SSL u otra forma de cifrado para evitar que la información de inicio de sesión sea interceptada o robada. Si el formulario de inicio de sesión se entrega mediante SSL, es imprescindible acceder a la página a la que se envía mediante SSL. Todos los enlaces o URL presentes en esa página (no solo la acción del formulario) deben entregarse mediante HTTPS. Esto evitará ataques de intermediario (Man-in-the-Middle) en el formulario de inicio de sesión.

Solo la acción del formulario debe enviarse mediante HTTPS. Esto evitará ataques de intermediario (Man-in-the-Middle) en el formulario de inicio de sesión.
Las recomendaciones incluyen garantizar que las áreas sensibles de su aplicación web tengan protocolos de cifrado adecuados para evitar que la información de inicio de sesión y otros datos que podrían ser útiles para un atacante sean interceptados.

Implicación:

Un atacante que aproveche esta vulnerabilidad de diseño podría utilizar la información para escalar su método de ataque, posiblemente conduciendo a la suplantación de un usuario legítimo, el robo de datos confidenciales o la ejecución de acciones no previstas por los desarrolladores de la aplicación.

Arreglar:

Asegúrese de que las áreas sensibles de su aplicación web tengan implementados protocolos de cifrado adecuados para evitar que la información de inicio de sesión y otros datos que podrían ser útiles para un atacante sean interceptados.

Referencia:

Aviso: http://www.kb.cert.org/vuls/id/466433

Nombres de archivos:

- http://172.16.21.39:7001/SisAdhereVerFirmas/
- http://172.16.21.39:7001/SisAdhereVerFirmas/inicioErr.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec

Alto — Scripting entre marcos

Resumen:

Una vulnerabilidad de secuencias de comandos entre marcos (XFS) puede permitir a un atacante cargar la aplicación vulnerable dentro de una etiqueta iframe HTML en una página maliciosa. El atacante podría usar esta vulnerabilidad para diseñar un ataque de clickjacking y realizar ataques de phishing, rastreo de frames, ingeniería social o falsificación de solicitudes entre sitios.

Secuestro de clics
El objetivo de un ataque de clickjacking es engañar a la víctima (usuario) para que interactúe con los elementos de la interfaz de usuario que elija el atacante en el sitio web objetivo sin su conocimiento y, posteriormente, ejecute funciones privilegiadas en su nombre. Para lograrlo, el atacante debe explotar la vulnerabilidad XFS para cargar el objetivo del ataque dentro de una etiqueta iframe, ocultarlo mediante hojas de estilo en cascada (CSS) y superponer el contenido de phishing en la página maliciosa. Al colocar los elementos de la interfaz de usuario en la página de phishing de forma que se superpongan con los de la página objetivo del ataque, el atacante puede asegurarse de que la víctima interactúe con los elementos de la interfaz de usuario de la página objetivo que no son visibles para ella.

WebInspect ha detectado una página que potencialmente maneja información confidencial mediante un formulario HTML con un campo de ingreso de contraseña y no tiene protección XFS.

Un La técnica eficaz para romper el marco fue no observado mientras se carga esta página dentro a marco.

Ejecución:

Cree una página de prueba que contenga una etiqueta iframe HTML cuyo atributo src esté establecido en http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec. Si la página de destino se enmarca correctamente, indica que la aplicación es susceptible a XFS.

Tenga en cuenta que WebInspect solo reportará una instancia de esta comprobación en cada host dentro del alcance del análisis. Sin embargo, las demás páginas visibles del sitio también pueden ser vulnerables a XFS y, por lo tanto, deben protegerse con una solución adecuada.

Implicación:

Una vulnerabilidad de scripting entre marcos podría permitir a un atacante incrustar la aplicación vulnerable dentro de un iframe. La explotación de esta vulnerabilidad podría resultar en:

Secuestro de eventos de usuario como pulsaciones de teclas
Robo de información sensible
Ejecución de funcionalidad privilegiada mediante la combinación con ataques de falsificación de solicitud entre sitios

Arreglar:

La directiva frame-ancestors de la Política de Seguridad de Contenido (CSP) deja obsoleto el encabezado X-Frame-Options. Ambas proporcionan una técnica de mitigación basada en políticas contra vulnerabilidades de scripting entre marcos. La diferencia radica en que, mientras que la técnica X-Frame-Options solo verifica la ubicación del documento de nivel superior, el encabezado frame-ancestors de la CSP verifica la conformidad de todos los ancestros.

Si tanto los encabezados CSP frame-ancestors como los X-Frame-Options están presentes y son compatibles, prevalecerá la directiva CSP.
WebInspect recomienda utilizar los encabezados CSP frame-ancestors y X-Frame-Options ya que Internet Explorer y muchas versiones anteriores de otros navegadores no admiten CSP.

Además, los desarrolladores también deben usar JavaScript para romper marcos del lado del cliente como protección contra XFS. Esto permitirá que los usuarios de navegadores antiguos que no admiten el encabezado X-Frame-Options también estén protegidos contra ataques de clickjacking.

Opciones de X-Frame
Los desarrolladores pueden usar este encabezado para indicar al navegador las acciones apropiadas a realizar si su sitio está incluido dentro de un

Los desarrolladores pueden usar este encabezado para indicar al navegador las acciones apropiadas a realizar si su sitio está incluido dentro de un iframe. Los desarrolladores deben configurar el encabezado X-Frame-Options en uno de los siguientes valores permitidos:

DENEGAR

Denegar todos los intentos de enmarcar la página

MISMO ORIGEN

La página puede ser enmarcada por otra página sólo si pertenece al mismo origen que la página que está siendo enmarcada.

Política de seguridad del contenido: ancestros del marco

Los desarrolladores pueden usar el encabezado CSP con la directiva frame-ancestors, que reemplaza al encabezado X-Frame-Options, para indicar al navegador qué acciones debe realizar si su sitio está incluido dentro de un iframe. Los desarrolladores pueden establecer el atributo frame-ancestors con uno de los siguientes valores permitidos:

'ninguno'

Equivalente a “DENY”: denegar todos los intentos de enmarcar la página

'ser'

Equivalente a “SAMEORIGIN”: la página puede estar enmarcada por otra página solo si pertenece al mismo origen que la página siendo incriminado

<fuente del esquema>

Los desarrolladores también pueden especificar un esquema como http: o https: que pueda enmarcar la página.

Referencia:

Rompiendo marcos:
Destruyendo Frame Busting: Un estudio de vulnerabilidades de clickjacking en sitios populares
OWASP: Rompiendo marcos Rompiendo marcos

OWASP:
Secuestro de clics

Política de seguridad de contenido (CSP)
CSP: ancestros de trama

Especificación:
Política de seguridad de contenido Nivel 2
Borrador del IETF sobre opciones de X-Frame

Configuración del servidor:
Apache, nginx

Informe de ciberseguridad de HP 2012
El encabezado X-Frame-Options: un error al iniciar

Nombres de archivos: ● http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec

---

Alto — A menudo mal utilizado: Carga de archivos

Resumen:

Permitir que los usuarios carguen archivos puede permitir que los atacantes inyecten contenido peligroso o código malicioso para ejecutarlo en el servidor.

WebInspect ha cargado exitosamente un archivo al servidor.

WebInspect ha cargado correctamente WebInspect_11503_File_8.exe al servidor en:

http://172.16.21.39:7001/SisAdhereVerFirmas/lote/

NOTAS:

El archivo cargado es un archivo ejecutable.
archivo fue subido al servidor por WebInspect durante la verificación de carga de archivos peligrosos. Por favor, busque el archivo una vez finalizada la prueba.
Eliminar y

El cheque acepta tres entradas de cheque de la aplicación. Estas entradas de verificación son del usuario. Las entradas de verificación permiten personalizar la verificación de la configurable. aplicación y ayudan a mejorar el rendimiento. Si este resultado parece ser falso positivo (o se sospecha que hay falsos negativos para un análisis de las entradas de comprobar precisión. verificación disponibles), esta vulnerabilidad) considere configurar uno para obtener resultados o más Llegar más precisos.

Ejecución:

Navegar a:
http://172.16.21.39:7001/SisAdhereVerFirmas/lote/navegarCargarArchivo.reniec
y cargar un archivo del tipo reportado.

Implicación:

Independientemente del lenguaje de programación, los ataques más devastadores suelen implicar la ejecución remota de código, donde un atacante logra ejecutar código malicioso en el contexto del programa. Si se permite a los atacantes subir archivos a un directorio accesible desde la web y hacer que estos se transmitan a un intérprete de código (p. ej., JSP/ASPX/PHP), pueden provocar que el código malicioso contenido en estos archivos se ejecute en el servidor.

Las implicaciones exactas dependen de la naturaleza de los archivos que un atacante pueda subir. Estas incluyen desde la publicación de contenido no autorizado hasta la facilitación de ataques de phishing, e incluso la vulneración total del servidor web.

Incluso si un programa almacena los archivos subidos en un directorio inaccesible desde la web, los atacantes podrían aprovechar la capacidad de introducir contenido malicioso en el servidor para lanzar otros ataques. Si el programa es susceptible a la manipulación de rutas, la inyección de comandos o vulnerabilidades de inclusión de archivos peligrosos, un atacante podría subir un archivo con contenido malicioso y hacer que el programa lo lea o lo ejecute aprovechando otra vulnerabilidad.

Arreglar:

No acepte archivos adjuntos si puede evitarlos. Si un programa debe aceptar archivos adjuntos, restrinja la capacidad de un atacante para proporcionar contenido malicioso aceptando únicamente los tipos específicos de contenido que el programa espera. La mayoría de los ataques que se basan en el contenido subido requieren que los atacantes puedan proporcionar el contenido que elijan. Imponer restricciones al contenido que el programa aceptará limitará considerablemente el alcance de posibles ataques. Compruebe los nombres, las extensiones y el contenido de los archivos para asegurarse de que sean los esperados y aceptables para la aplicación. Dificulte al atacante la determinación del nombre y la ubicación de los archivos subidos. Estas soluciones suelen ser específicas del programa y varían desde almacenar los archivos subidos en un directorio con un nombre generado a partir de un valor aleatorio fuerte al inicializar el programa, hasta asignar a cada archivo subido un nombre aleatorio y rastrearlos mediante entradas en una base de datos.

Asegúrese de que se realicen los siguientes pasos para desinfectar el archivo que se recibe:

Limite los tipos de archivos que se pueden subir. Por ejemplo, en una página de subida de imágenes, se debe rechazar cualquier archivo que no sea .jpg.

Asegúrese de que el usuario web no tenga ningún control sobre el nombre y la ubicación del archivo cargado en el servidor.
Nunca utilice el nombre que el usuario le asigna.
Nunca derive el nombre del archivo del nombre de usuario o del ID de sesión del usuario web.
No coloque el archivo en un directorio accesible para usuarios web. Es preferible que esta ubicación esté fuera de la raíz web.
Asegúrese de que se establezcan permisos estrictos tanto para el archivo cargado como para el directorio en el que se encuentra.
No permita permisos de ejecución en los archivos subidos. Si es posible, deniegue todos los permisos a todos los usuarios excepto al usuario de la aplicación web.

Verifique que el archivo subido contenga el contenido adecuado. Por ejemplo, un archivo JPEG subido debe tener un encabezado JPEG estándar.

Referencia:

Carga segura de archivos en aplicaciones web PHP

Mapeo de estándares - Enumeración de debilidades comunes - (CWE) CWE ID 434

Carga de archivos sin restricciones de OWASP

Nombres de archivos: ● http://172.16.21.39:7001/SisAdhereVerFirmas/lote/navegarCargarArchivo.reniec

Medio — Manejo deficiente de errores: Excepción no controlada

Resumen:

Se encontró un mensaje de error en tiempo de ejecución de Java que indica que la evaluación generó una excepción no controlada en el código de su aplicación web. Las excepciones no controladas son circunstancias en las que la aplicación recibe una entrada de usuario (o parámetros) inesperada y no sabe cómo gestionarlos. En muchos casos, un atacante puede aprovechar las condiciones que causan estos errores para obtener acceso no autorizado al sistema. Se recomienda proporcionar lógica para gestionar cualquier situación en la que un parámetro no se pase o se pase incorrectamente.

Implicación:

El mensaje de error puede contener la ubicación del archivo donde se encuentra la función infractora. Esto puede revelar la ruta absoluta del directorio raíz web, así como proporcionar al atacante la ubicación de los archivos de inclusión de la aplicación o la información de configuración. Incluso puede revelar la parte del código que falló.

Nota: Esta vulnerabilidad puede ser un falso positivo si la página en la que está marcada es documentación técnica.

Arreglar:

Para operaciones de seguridad:

En última instancia, este problema requerirá un cambio en el código de la aplicación. Sin embargo, siga estas recomendaciones para garantizar la seguridad de su aplicación web:

Utilice códigos de error uniformes: Asegúrese de no proporcionar información inadvertidamente a un atacante mediante mensajes de error incoherentes o contradictorios. Por ejemplo, no revele información no deseada mediante mensajes de error como "Acceso denegado", que también le permitirán al atacante saber que el archivo que busca existe. Utilice códigos de error uniformes.

Como "Acceso denegado", que también permite al atacante saber que el archivo que busca existe. Use una terminología coherente para archivos y carpetas que existen, que no existen y que tienen acceso de lectura denegado.
Mensajes de error informativos: Asegúrese de que los mensajes de error no revelen demasiada información. Nunca se deben revelar al usuario final rutas completas o parciales, nombres de variables y archivos, nombres de filas y columnas en tablas, ni errores específicos de bases de datos. Recuerde que un atacante recopilará la mayor cantidad de información posible y luego combinará fragmentos de información aparentemente inofensiva para diseñar un ataque.

Manejo adecuado de errores: Utilice páginas de error genéricas y lógica de manejo de errores para informar a los usuarios finales sobre posibles problemas. No proporcione información del sistema ni otros datos que un atacante pueda usar al orquestar un ataque.

Para el desarrollo:

Este problema surge de la validación incorrecta de los caracteres aceptados por la aplicación. Al pasar un parámetro a una página web generada dinámicamente, se debe asumir que los datos podrían tener un formato incorrecto. La aplicación debe contener la lógica suficiente para gestionar cualquier situación en la que un parámetro no se pase o se pase incorrectamente. Tenga en cuenta cómo se envían los datos, como resultado de un GET o un POST. Además, para desarrollar código seguro y estable, trate las cookies como parámetros. Las siguientes recomendaciones le ayudarán a garantizar la seguridad de sus aplicaciones web.

Defina estrictamente el tipo de dato: Defina estrictamente el tipo de dato (una cadena, un carácter alfanumérico, etc.) que aceptará la aplicación. Valide la entrada para detectar caracteres incorrectos. Adopte la filosofía de usar lo bueno en lugar de lo malo. Defina el conjunto de caracteres permitido. Por ejemplo, si un campo debe recibir un número, permita que solo acepte números. Defina las longitudes máxima y mínima de datos que aceptará la aplicación.

Verificar que se esté pasando el parámetro: Si se omite un parámetro que se espera que se pase a una página web dinámica, la aplicación debe mostrar un mensaje de error aceptable al usuario. Además, nunca utilice un parámetro hasta haber verificado que se ha pasado a la aplicación.

Verifique el formato correcto: Nunca asuma que un parámetro tiene un formato válido. Esto es especialmente cierto si el parámetro se pasa a una base de datos SQL. Cualquier cadena que se pase directamente a una base de datos sin verificar previamente su formato correcto puede representar un riesgo importante para la seguridad. Además, aunque un parámetro se proporcione normalmente mediante un cuadro combinado o un campo oculto, no asuma que el formato es correcto. Un hacker intentará primero alterar estos parámetros al intentar acceder a su sitio web.

Verificar los nombres de archivo que se pasan mediante un parámetro: Si se utiliza un parámetro para determinar qué archivo procesar, nunca utilice el nombre del archivo antes de verificar su validez. En concreto, compruebe la existencia de caracteres que indiquen un recorrido de directorio, como .../, c:\ y /.

No almacene datos críticos en parámetros ocultos: Muchos programadores cometen el error de almacenar datos críticos en un parámetro oculto o una cookie. Suponen que, como el usuario no los ve, es un buen lugar para almacenar datos como el precio, el número de pedido, etc. Tanto los parámetros ocultos como las cookies pueden manipularse y devolverse al servidor, así que nunca asuma que el cliente devolvió lo que envió mediante un parámetro oculto o una cookie.

Para control de calidad:

Unas pruebas sencillas suelen determinar cómo reaccionará su aplicación web a diferentes errores de entrada. Se requieren pruebas más exhaustivas para detectar errores internos y evaluar la reacción del sitio. Esta evaluación realiza ambas tareas.

Asegúrese de que el esquema de gestión de errores sea coherente y no revele información privada sobre su aplicación web. Un dato aparentemente inocuo puede proporcionar a un atacante los medios para descubrir información adicional que pueda utilizar para llevar a cabo un ataque. Haga las siguientes observaciones:

¿Recibe el mismo tipo de error para archivos existentes y no existentes?
¿El error incluye frases (como "Permiso denegado") que podrían revelar la existencia de un archivo?

Nombres de archivos: ● http://172.16.21.39:7001/SisAdhereVerFirmas/recursos

Medio — Scripting entre marcos

Resumen:

Una vulnerabilidad de secuencias de comandos entre marcos (XFS) puede permitir a un atacante cargar la aplicación vulnerable dentro de una etiqueta iframe HTML en una página maliciosa. El atacante podría usar esta vulnerabilidad para diseñar un ataque de clickjacking y realizar ataques de phishing, rastreo de frames, ingeniería social o falsificación de solicitudes entre sitios.

Secuestro de clics
El objetivo de un ataque de clickjacking es engañar a la víctima (usuario) para que interactúe con los elementos de la interfaz de usuario que elija el atacante en el sitio web objetivo sin su conocimiento y, posteriormente, ejecute funciones privilegiadas en su nombre. Para lograrlo, el atacante debe explotar la vulnerabilidad XFS para cargar el objetivo del ataque dentro de una etiqueta iframe, ocultarlo mediante hojas de estilo en cascada (CSS) y superponer el contenido de phishing en la página maliciosa. Al colocar los elementos de la interfaz de usuario en la página de phishing de forma que se superpongan con los de la página objetivo del ataque, el atacante puede asegurarse de que la víctima interactúe con los elementos de la interfaz de usuario de la página objetivo que no son visibles para ella.

WebInspect ha detectado una respuesta que contiene uno o más formularios que aceptan la entrada del usuario pero carece de protección XFS.

Una técnica eficaz para romper el marco fue no observado mientras se carga esta página dentro a marco.

Ejecución:

Cree una página de prueba que contenga una etiqueta iframe HTML cuyo atributo src esté establecido en http://172.16.21.39:7001/SisAdhereVerFirmas/listaExpresiones.reniec. Si la página de destino se enmarca correctamente, indica que la aplicación es susceptible a XFS.

Tenga en cuenta que WebInspect solo reportará una instancia de esta comprobación en cada host dentro del alcance del análisis. Sin embargo, las demás páginas visibles del sitio también pueden ser vulnerables a XFS y, por lo tanto, deben protegerse con una solución adecuada.

Implicación:

Una vulnerabilidad de scripting entre marcos podría permitir a un atacante incrustar la aplicación vulnerable dentro de un iframe. La explotación de esta vulnerabilidad podría resultar en:

Secuestro de eventos de usuario como pulsaciones de teclas
Robo de información sensible
Ejecución de funcionalidad privilegiada mediante la combinación con ataques de falsificación de solicitud entre sitios

Arreglar:

La directiva frame-ancestors de la Política de Seguridad de Contenido (CSP) deja obsoleto el encabezado X-Frame-Options. Ambas proporcionan una técnica de mitigación basada en políticas contra vulnerabilidades de scripting entre marcos. La diferencia radica en que, mientras que la técnica X-Frame-Options solo verifica la ubicación del documento de nivel superior, el encabezado frame-ancestors de la CSP verifica la conformidad de todos los ancestros.

Si tanto los encabezados CSP frame-ancestors como los X-Frame-Options están presentes y son compatibles, prevalecerá la directiva CSP.
WebInspect recomienda utilizar los encabezados CSP frame-ancestors y X-Frame-Options ya que Internet Explorer y muchas versiones anteriores de otros navegadores no admiten CSP.

Además, los desarrolladores también deben usar JavaScript para romper marcos del lado del cliente como protección contra XFS. Esto permitirá que los usuarios de navegadores antiguos que no admiten el encabezado X-Frame-Options también estén protegidos contra ataques de clickjacking.

Opciones de X-Frame
Los desarrolladores pueden usar este encabezado para indicar al navegador las acciones apropiadas a realizar si su sitio está incluido dentro de un iframe. Los desarrolladores deben configurar el encabezado X-Frame-Options en uno de los siguientes valores permitidos:

DENEGAR
Denegar todos los intentos de enmarcar la página
MISMO ORIGEN
La página puede ser enmarcada por otra página sólo si pertenece al mismo origen que la página que está siendo enmarcada.

Política de seguridad del contenido: ancestros del marco

Los desarrolladores pueden usar el encabezado CSP con la directiva frame-ancestors, que reemplaza al encabezado X-Frame-Options, para indicar al navegador qué acciones debe realizar si su sitio está incluido dentro de un iframe. Los desarrolladores pueden establecer el atributo frame-ancestors con uno de los siguientes valores permitidos:

'ninguno'
Equivalente a “DENY”: denegar todos los intentos de enmarcar la página
'ser'
Equivalente a “SAMEORIGIN”: la página puede estar enmarcada por otra página solo si pertenece al mismo origen que la página
siendo incriminado
<fuente del esquema>
Los desarrolladores también pueden especificar un esquema como http: o https: que pueda enmarcar la página.

Referencia:

Rompiendo marcos:
Destruyendo Frame Busting: Un estudio de vulnerabilidades de clickjacking en sitios populares
OWASP: Rompiendo marcos Rompiendo marcos

OWASP:
Secuestro de clics

Política de seguridad de contenido (CSP)
CSP: ancestros de trama

Especificación:
Política de seguridad de contenido Nivel 2
Borrador del IETF sobre opciones de X-Frame

Configuración del servidor:

Apache, nginx

Informe de ciberseguridad de HP 2012

El encabezado X-Frame-Options: un error al iniciar

Nombres de archivos:

- http://172.16.21.39:7001/SisAdhereVerFirmas/listaExpresiones.reniec

Bajo

A menudo mal utilizado: Carga de archivos

Resumen:

Se encontró un indicador de capacidad de carga de archivos. Esta capacidad permite a un usuario web enviar un archivo desde su ordenador al servidor web.
Si la aplicación web que recibe el archivo no lo examina cuidadosamente en busca de contenido malicioso, un atacante podría usar la carga de archivos para ejecutar comandos arbitrarios en el servidor. Se recomienda adoptar una política estricta de carga de archivos que impida la carga de material malicioso mediante la limpieza y el filtrado.

Implicación:

Las implicaciones exactas dependen de la naturaleza de los archivos que un atacante pueda subir. Estas incluyen desde la publicación de contenido no autorizado hasta la facilitación de ataques de phishing, e incluso la vulneración total del servidor web.

Arreglar:

Para operaciones de seguridad:
Esta comprobación forma parte de las pruebas de aplicaciones desconocidas. Estas pruebas buscan descubrir nuevas vulnerabilidades tanto en software personalizado como comercial. Por ello, no existen parches ni descripciones específicas para este problema. Si la página no muestra ninguna función de carga de archivos, esta comprobación puede ignorarse sin problemas. Puede configurar el escáner para que ignore esta vulnerabilidad haciendo clic derecho en el nodo de vulnerabilidad en el árbol de resultados y seleccionando "Ignorar vulnerabilidad".

Para control de calidad:
Este problema deberá resolverse en el código de producción. Notifique al desarrollador correspondiente.

Para el desarrollo:
Asegúrese de que se realicen los siguientes pasos para desinfectar el archivo que se recibe:

Limite los tipos de archivos que se pueden subir. Por ejemplo, en una página de subida de imágenes, se debe rechazar cualquier archivo que no sea .jpg.

Asegúrese de que el usuario web no tenga ningún control sobre el nombre y la ubicación del archivo cargado en el servidor.
Nunca utilice el nombre que el usuario le asigna.
Nunca derive el nombre del archivo del nombre de usuario o del ID de sesión del usuario web.
No coloque el archivo en un directorio accesible para usuarios web. Es preferible que esta ubicación esté fuera de la raíz web.
Asegúrese de que se establezcan permisos estrictos tanto para el archivo cargado como para el directorio en el que se encuentra.
No permita permisos de ejecución en los archivos subidos. Si es posible, deniegue todos los permisos a todos los usuarios excepto a la aplicación web.
usuario.
Verifique que el archivo subido contenga el contenido adecuado. Por ejemplo, un archivo JPEG subido debe tener un encabezado JPEG estándar.

Nombres de archivos:

- http://172.16.21.39:7001/SisAdhereVerFirmas/js/jquery/jquery.form.js
- http://172.16.21.39:7001/SisAdhereVerFirmas/lote/navegarCargarArchivo.reniec

Bajo

Fuga de información del sistema: IP interna

Resumen:

Se descubrió una cadena que coincide con un rango de direcciones IPv4 o IPv6 interno/reservado. Esto podría revelar información sobre el esquema de direccionamiento IP de la red interna y ser valioso para los atacantes. Los rangos internos de IPv4/IPv6 son:
10.xxx
172.16.xx a 172.31.xx
192.168.xx
fd00::x
Si no es parte de la documentación técnica, las recomendaciones incluyen eliminar la cadena del servidor de producción.

Arreglar:

Este problema puede ocurrir por varias razones. La más común es que el mensaje de error de la aplicación o del servidor web revele la dirección IP. Esto se puede solucionar determinando dónde desactivar los mensajes de error detallados en la aplicación o el servidor web.
Otra razón común se debe a un comentario ubicado en el código fuente de la página web. Este puede eliminarse fácilmente.

Nombres de archivos: ● http://172.16.21.39:7001/SisAdhereVerFirmas/configuracion/accionConsultarIpAcceso.reniec

Bajo Configuración incorrecta del servidor web: mensaje de error del servidor

Resumen:

Se detectó una respuesta de error del servidor. El servidor podría estar experimentando errores debido a un mal funcionamiento de la aplicación, una configuración incorrecta o un valor malicioso enviado durante el proceso de auditoría. Si bien las respuestas de error en sí mismas no son peligrosas, permiten a los atacantes comprender cómo la aplicación gestiona las condiciones de error. Los errores que un atacante puede activar remotamente también pueden provocar un ataque de denegación de servicio u otra vulnerabilidad más grave.
Las recomendaciones incluyen diseñar y agregar mecanismos de manejo de errores consistentes que sean capaces de manejar cualquier entrada del usuario a su aplicación web, brindar detalles significativos a los usuarios finales y evitar que se muestren mensajes de error que podrían proporcionar información útil a un atacante.

Implicación:

El servidor ha emitido una respuesta de error 500. Si bien el contenido de la página de error puede no revelar información sobre el error técnico, el código de estado 500 confirma que se produjo un error. Saber si ciertas entradas desencadenan un error del servidor puede ayudar o informar a un atacante sobre posibles vulnerabilidades.

Arreglar:

Para operaciones de seguridad:

Los mensajes de error del servidor, como "Archivo protegido contra acceso", suelen revelar más información de la prevista. Por ejemplo, un atacante que recibe este mensaje puede estar relativamente seguro de la existencia del archivo, lo que podría proporcionarle la información necesaria para seguir otras pistas o para ejecutar un exploit. Las siguientes recomendaciones ayudarán a garantizar que un posible atacante no obtenga información valiosa de ningún mensaje de error del servidor que se presente.

Códigos de error uniformes: Asegúrese de no proporcionar información inadvertidamente a un atacante mediante mensajes de error incoherentes o contradictorios. Por ejemplo, no revele información no deseada utilizando mensajes de error como "Acceso denegado", que también le permitirán al atacante saber que el archivo que busca existe. Utilice una terminología uniforme para los archivos y carpetas que existen, los que no existen y los que tienen acceso de lectura denegado.
Mensajes de error informativos: Asegúrese de que los mensajes de error no revelen demasiada información. Nunca se deben revelar al usuario final rutas completas o parciales, nombres de variables y archivos, nombres de filas y columnas en tablas, ni errores específicos de bases de datos. Recuerde que un atacante recopilará la mayor cantidad de información posible y luego combinará fragmentos de información aparentemente inofensiva para diseñar un método de ataque.
Manejo adecuado de errores: Utilice páginas de error genéricas y lógica de manejo de errores para informar a los usuarios finales sobre posibles problemas. No proporcione información del sistema ni otros datos que un atacante pueda utilizar al orquestar un ataque.

Eliminar mensajes de error detallados

Encuentre instrucciones para desactivar la mensajería de error detallada en IIS en este enlace:

http://support.microsoft.com/kb/294807

Para el desarrollo:

Desde una perspectiva de desarrollo, la mejor manera de prevenir problemas derivados de mensajes de error del servidor es adoptar técnicas de programación segura que eviten los problemas que podrían surgir si un atacante descubre demasiada información sobre la arquitectura y el diseño de su aplicación web. Las siguientes recomendaciones pueden servir de base para ello.

Defina estrictamente el tipo de datos (por ejemplo, una cadena, un carácter alfanumérico, etc.) que aceptará la aplicación.
Usa lo bueno en lugar de lo malo. Valida la entrada para detectar caracteres incorrectos.
No muestre mensajes de error al usuario final que proporcionen información (como nombres de tablas) que pueda utilizarse para orquestar un ataque.

Define el conjunto de caracteres permitido. Por ejemplo, si un campo debe recibir un número, permite que solo acepte números.
Define las longitudes de datos máximas y mínimas que aceptará la aplicación.
Especifique rangos numéricos aceptables para la entrada.

Para control de calidad:

La mejor medida que pueden tomar los asociados de control de calidad es garantizar la coherencia del esquema de gestión de errores. ¿Recibe un tipo de error diferente para un archivo inexistente que para uno existente? ¿Se utilizan frases como "Permiso denegado" que podrían revelar la existencia de un archivo a un atacante? Los métodos inconsistentes para gestionar errores ofrecen a un atacante una forma muy eficaz de recopilar información sobre su aplicación web.

Referencia:

Apache:
Consejos de seguridad para la configuración del servidor
Protección de documentos confidenciales en su sitio
Protección de Apache: control de acceso

Microsoft:
Cómo configurar los permisos NTFS y los derechos de usuario necesarios para un servidor web IIS 5.0
Permisos y derechos de usuario predeterminados para IIS 6.0
Descripción de los códigos de estado de Microsoft Internet Information Services (IIS) 5.0 y 6.0

Nombres de archivos:

- http://172.16.21.39:7001/SisAdhereVerFirmas/usr/share/zoneinfo/GMT%3ffakestp.html
- http://172.16.21.39:7001/SisAdhereVerFirmas/recursos
- http://172.16.21.39:7001/SisAdhereVerFirmas/GET

Bajo — Configuración incorrecta del servidor web: configuración de tipo de contenido insegura

Resumen:

La falta de un encabezado Content-Type en la respuesta HTTP podría exponer la aplicación a vulnerabilidades de secuencias de comandos entre sitios a través de:

Falta de coincidencia en el rastreo de
contenido No especificar explícitamente el tipo de contenido servido por el recurso solicitado puede permitir a los atacantes realizar ataques de secuencias de comandos entre sitios explotando las inconsistencias en las técnicas de rastreo de contenido empleadas por los navegadores.
El encabezado Content-Type es utilizado por:

El servidor web determina cómo el agente de usuario interpreta el recurso solicitado. En ausencia de este encabezado, el navegador depende de algoritmos de rastreo de contenido para determinar el tipo de contenido y procesarlo o interpretarlo en consecuencia.
Filtros de carga de archivos para descartar tipos de archivo no permitidos por la aplicación. Si no hay un encabezado Content-Type, el filtro de carga de archivos se basa en la extensión o el contenido del archivo para detectar y almacenar un tipo MIME adecuado para el archivo cargado.

La falta de una especificación explícita del tipo de contenido puede permitir a los atacantes aprovechar la discrepancia entre el algoritmo de rastreo MIME utilizado por el navegador y el filtro de subida. Al subir archivos con extensiones inocuas (como .jpg), un atacante puede eludir fácilmente el filtro de subida y subir archivos con contenido HTML malicioso. Sin embargo, el algoritmo de rastreo de contenido del navegador lo renderizará como HTML según el contenido del archivo, ejecutando así cualquier script malicioso incrustado en el contenido HTML.

Desajuste del conjunto de caracteres

La especificación del conjunto de caracteres forma parte del encabezado Content-Type. La ausencia de esta especificación podría permitir a los atacantes eludir los filtros de validación de entrada o la funcionalidad de escape de entidades HTML y realizar ataques de secuencias de comandos entre sitios (CSE) contra la aplicación objetivo. Si no se especifica el conjunto de caracteres, los navegadores intentarán determinar cuál es el más adecuado. Esto podría provocar una discrepancia entre el conjunto de caracteres asumido por la aplicación durante la generación del contenido y el que utiliza el navegador durante el análisis e interpretación del mismo. Un atacante puede aprovechar esta inconsistencia para codificar ataques con un conjunto de caracteres que oculte las cargas maliciosas de los filtros de validación y los mecanismos de escape implementados por la aplicación, pero que, al mismo tiempo, sea interpretado por el navegador como una entidad ejecutable válida.

Ejecución:

Los siguientes escenarios de ejemplo demuestran la explotación de la debilidad:

Desajuste de rastreo de contenido

El atacante sube un archivo con extensión .jpg sin especificar el tipo de contenido. El archivo contiene HTML y JavaScript maliciosos incrustados.

En ausencia del encabezado Content-Type, la aplicación guarda el archivo cargado junto con el tipo MIME del .jpg.

El atacante utiliza ingeniería social para persuadir al objetivo deseado a acceder al archivo cargado.

Al recibir el archivo solicitado sin el encabezado Content-Type, el navegador del objetivo asume que el tipo de contenido es HTML basándose en el contenido HTML y JavaScript que contiene y renderiza el archivo, lo que provoca la ejecución de la carga útil JavaScript del atacante.

Desajuste del conjunto de caracteres

0. El atacante convierte la carga útil deseada de <script>alert(document.location)</script> en una cadena codificada en UTF-7 +ADw-script+AD4-alert(document.location)+ADw-/script+AD4 y la envía a la aplicación vulnerable.

Una aplicación que utilice el conjunto de caracteres ISO-8859-1 para filtrar o escapar caracteres especiales no detectará los caracteres '<' y ">' como peligrosos.

La ausencia de la especificación del conjunto de caracteres debido a la ausencia del encabezado Content-Type obligará al navegador a adivinar el conjunto de caracteres que se usará para representar la respuesta de la aplicación que contiene la carga útil del atacante. Si el navegador adivina correctamente la codificación UTF-7, la carga útil inyectada se ejecutará correctamente.

Implicación:

La aplicación no logra imponer restricciones al análisis e interpretación del contenido de la respuesta, lo que permite a los atacantes eludir los filtros de validación o la funcionalidad de escape e introducir scripts maliciosos y obligar al navegador a ejecutar la carga útil deseada.

Arreglar:

Configure el servidor para enviar el tipo de contenido y la información del conjunto de caracteres adecuados para el recurso solicitado.

Referencia:

Configuración del servidor
Tipos MIME en IIS 7
Negociación de contenido - Servidor HTTP Apache

Detección de contenido:
Estándar de olfateo de mimos
Firmas de rastreo de contenido
Detección segura de contenido para navegadores web [PDF]

OWASP:
Apéndice D de la Guía de pruebas de OWASP: Inyección codificada

Nombres de archivos:
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/accionConsultaReversion.reniec

---

Bajo

HTML5: Política de seguridad de contenido mal configurada

Resumen:

La Política de Seguridad de Contenido (CSP) es un encabezado de seguridad declarativo que permite a los desarrolladores determinar desde qué dominios el sitio puede cargar contenido o iniciar conexiones al visualizarse en el navegador web. Proporciona una capa adicional de seguridad contra vulnerabilidades críticas como secuencias de comandos entre sitios, secuestro de clics, acceso entre orígenes, etc., además de la validación de entrada y la inclusión de una lista de permitidos en el código. Sin embargo, un encabezado mal configurado no proporciona esta capa adicional de seguridad. La política se define mediante quince directivas, ocho de las cuales controlan el acceso a los recursos:

script-src
img-src
objeto-src
estilo_origen
fuente-src
media-src
marco-src
conectar-src

Cada uno de estos toma una lista de fuentes como un valor que especifica los dominios a los que el sitio puede acceder para la función cubierta por esa directiva. Los desarrolladores pueden usar el comodín * para indicar todo o parte de la fuente. Ninguna de las directivas es obligatoria. Los navegadores permitirán todas las fuentes para la directiva no listada o derivarán su valor de la directiva opcional default-src. Además, la especificación para este encabezado ha evolucionado con el tiempo. Se implementó como X-Content-Security-Policy en Firefox hasta la versión 23 y en IE hasta la versión 10, y se implementó como X-Webkit-CSP en Chrome hasta la versión 25. Ambos nombres están obsoletos en favor del nombre ahora estándar Content Security Policy. Dado el número de directivas, dos nombres alternativos obsoletos y la forma en que se tratan las ocurrencias múltiples del mismo encabezado y la directiva repeat en un solo encabezado, existe una alta propensión a que el desarrollador pueda configurar incorrectamente este encabezado.

Propensión a que el desarrollador configure mal este encabezado.

Considere los siguientes escenarios de configuración incorrecta:

Puede ser una política demasiado permisiva si default-src no está configurado o está configurado como comodín y/o otras directivas están configuradas como comodín.
Se permiten varias instancias de este encabezado en la misma respuesta. Tanto el equipo de desarrollo como el de seguridad podrían configurar el encabezado, pero con políticas diferentes, o bien podrían usar los encabezados X-Content-Security-Poilcy o X-Webkit-CSP, ahora obsoletos. Las versiones obsoletas se ignoran si el encabezado Content-Security-Policy estándar está presente.
Si una directiva se repite dentro de la misma instancia del encabezado, se ignoran todas las ocurrencias posteriores.

Se detectaron las siguientes configuraciones incorrectas en http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec:

Política de seguridad de contenido excesivamente permisiva.

Política de seguridad de contenido: origen predeterminado, origen del script, origen del objeto, origen de la fuente, origen del estilo, origen de la imagen, origen del marco, origen de la conexión, origen del medio están configurados para comodín.

Ejecución:

Acceda al enlace http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec a través del proxy y observe el encabezado de política de seguridad de contenido mal configurado en los encabezados de respuesta.

Implicación:

El encabezado de la política de seguridad de contenido proporciona una capa adicional de seguridad para el sitio contra ataques del lado del cliente, como secuencias de comandos entre sitios. En ausencia de este encabezado, un atacante puede explotar vulnerabilidades del lado del cliente, como secuencias de comandos entre sitios, clickjacking y falsificación de solicitudes entre sitios.

Arreglar:

Elimine los valores comodín de source-list para limitar el alcance del acceso entre orígenes desde el sitio. Asegúrese de que se use el nombre Política de seguridad de contenido canónico para especificar la política. Los nombres X-Content-Security-Policy y X-Webkit-CSP están obsoletos. Cualquier referencia a estos encabezados solo es útil si se desea compatibilidad con navegadores anteriores. Se dice que la presencia de estos encabezados, además de Content-Security-Policy, causa comportamientos inesperados en ciertas versiones de navegadores.
Compatibilidad con el estándar Content-Security-Policy versión 2:

Edge: Compatible con Edge 15-18, con un error de nonce. Compatible a partir de la versión 75.
Chrome: en Chrome 36-38 faltan las directivas plugin-types, child-src, frame-ancestors, base-uri y form-action.
En Chrome 39 faltan las directivas plugin-types, child-src, base-uri y form-action. A partir de Chrome 40, son totalmente compatibles.

Firefox; Firefox 31-34 carece de las directivas plugin-types, child-src, frame-ancestors, base-uri y form-action. Firefox 35 carece de las directivas plugin-types, child-src, frame-ancestors y form-action. Firefox 36-44 carece de las directivas plugin-types y child-src. Firefox 45+ carece de la directiva plugin-types.

Además, la directiva report-uri se puede configurar para recibir informes de intentos de incumplimiento de la política. Estos informes
Puede utilizarse como una indicación temprana de problemas de seguridad en el sitio, así como para optimizar la política.

Referencia:

http://www.w3.org/TR/CSP/
https://owasp.org/www-community/controls/Política_de_Seguridad_del_Contenido
https://developer.mozilla.org/es-ES/docs/Web/Security/CSP/Introducción_a_la_Política_de_Seguridad_del_Contenido
https://politica-de-seguridad-de-contenido.com

Nombres de archivos:
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec

Bajo

Seguridad de cookies: falta el atributo SameSite

Resumen:

El atributo SameSite protege las cookies de ataques de falsificación de solicitud entre sitios (CSRF). El navegador añade automáticamente cookies a cada solicitud HTTP realizada al sitio que la configura. Las cookies pueden almacenar datos confidenciales, como el ID de sesión y el token de autorización, o datos del sitio que se comparten entre diferentes solicitudes al mismo sitio durante una sesión. Un atacante puede realizar un ataque de suplantación de identidad generando una solicitud al sitio autenticado desde la página de un sitio de terceros cargada en el equipo cliente, ya que el navegador añade automáticamente la cookie a la solicitud.

El atributo SameSite de una cookie permite a los sitios web controlar dicho comportamiento e impide que los navegadores la agreguen a la solicitud si esta se genera al cargar la página de un sitio web de terceros. El atributo SameSite puede tener los tres valores siguientes:

Estricto: cuando se configura como Estricto, las cookies solo se envían junto con las solicitudes en la navegación de nivel superior.
Lax: cuando se configura en Lax, las cookies se envían con la navegación de nivel superior desde el mismo host, así como también con las solicitudes GET originadas en el host desde sitios de terceros (por ejemplo, en iframe, enlace, href, etc. y la etiqueta de formulario con el método GET únicamente).
Ninguna: Las cookies se envían en todas las solicitudes realizadas al sitio dentro de la ruta y el ámbito de dominio definidos para la cookie. Solicitudes

Ninguna: Se envían cookies en todas las solicitudes realizadas al sitio web dentro de la ruta y el ámbito de dominio definidos para la cookie. Las solicitudes generadas al enviar formularios mediante el método POST también pueden enviar cookies con la solicitud.

Tenga en cuenta que las cookies con el atributo SameSite con el valor None deben configurarse con el atributo Secure; de lo contrario, el navegador las rechazará. Además, algunas versiones específicas de navegadores rechazan la cookie SameSite con el valor None; por ejemplo, las versiones de Chrome de la 51 a la 66, las versiones de UC Browser en Android anteriores a la 12.13.2, las versiones de Safari y los navegadores integrados en macOS 10.14, y todos los navegadores en iOS 12 rechazan las cookies configuradas con SameSite=None. Una solución alternativa para este problema es configurar una cookie alternativa con un prefijo o sufijo, como "adjunto a cookiename". Los sitios web pueden buscar esta cookie heredada si no encuentran una cookie configurada con SameSite=None.

Ejecución:

Inspeccione el valor de la cookie resaltado en la respuesta HTTP de la sesión vulnerable. A la cookie le falta el atributo SameSite.

Implicación:

Los sitios que configuran cookies sin el atributo SameSite tienen un mayor riesgo de sufrir ataques CSRF. Mediante CSRF, un atacante puede suplantar la identidad de un usuario válido y obtener acceso no autorizado a la funcionalidad de la aplicación. Además, las versiones recientes de los navegadores podrían rechazar las cookies que no tengan el atributo SameSite.

Arreglar:

Añade el atributo SameSite a todas las cookies. Desde febrero de 2020, el navegador Chrome ha establecido SameSite como atributo obligatorio para todas las cookies. Cualquier cookie sin el atributo SameSite se rechaza o se trata con el comportamiento predeterminado, que equivale a establecer el valor del atributo en Lax. Por lo tanto, cualquier cookie que deba enviarse, independientemente del origen de la solicitud (por ejemplo, las cookies de análisis), debe tener el valor del atributo SameSite en "ninguno".
Además, recomendamos a los desarrolladores que sigan añadiendo mitigaciones CSRF tradicionales al sitio junto con el atributo SameSite. Es posible que muchos usuarios del sitio aún utilicen versiones antiguas de su navegador para acceder al sitio. Estas versiones no comprenden el atributo SameSite.

Referencia:

https://tools.ietf.org/html/borrador-ietf-httpbis-rfc6265bis-05
https://www.chromium.org/updates/same-site/incompatible-clients
https://developer.mozilla.org/es-ES/docs/Web/HTTP/Headers/Set-Cookie/SameSite

Nombres de archivos:
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec

Bajo — Gestión de caché: Envenenamiento de caché web

Resumen:

WebInspect ha detectado un caso de envenenamiento de caché web por solicitudes GET complejas. Estas solicitudes contienen parámetros en el cuerpo de la solicitud. El envenenamiento de caché web aprovecha el comportamiento del servidor web, lo que provoca que se muestre una respuesta maliciosa a las solicitudes legítimas. En este tipo de envenenamiento de caché web, la clave de caché se deriva de los parámetros de consulta originales pasados en la línea de solicitud. Sin embargo, el valor del parámetro de consulta utilizado por el servidor es el valor duplicado enviado como parte del cuerpo de la solicitud. Esta respuesta modificada se almacena en caché y se entrega a los usuarios con solicitudes legítimas.

WebInspect ha detectado que la caché podría estar envenenada. La acción se pasó en el cuerpo de la solicitud y se observa que la respuesta inicial difiere de la respuesta posterior para la misma consulta. Esto indica que la caché podría estar envenenada.

Ejecución:

Para comprobar si hay envenenamiento de caché web, ejecute los siguientes pasos:

El destructor de caché es un parámetro de consulta que se agrega a una URL, por ejemplo, cacheBuster=someRandomValue. Agregue un destructor de caché.
Buster a la URL.
Agregue el parámetro de consulta como parte del cuerpo de la solicitud, por ejemplo, param=someRandomValue.
Envíe la solicitud y espere la respuesta.
A continuación, elimine el cuerpo de la solicitud y envíe la solicitud.
La respuesta recibida es la respuesta maliciosa almacenada en caché.

Implicación:

Si el servidor web es susceptible al envenenamiento de caché web, puede provocar que los usuarios legítimos reciban respuestas maliciosas.

Arreglar:

No se deben aceptar solicitudes FAT GET. Se debe podar el cuerpo de la solicitud o no aceptarla.

Referencia:

Envenenamiento de caché web

Nombres de archivos:
- http://172.16.21.39:7001/SisAdhereVerFirmas/recursos?accion=opcionesMenu&random=0.944999190033595&ca

---

Bajo — HTML5: Encabezado obsoleto

Resumen:

El encabezado X-XSS-Protection está habilitado. X-XSS-Protection se refiere a un encabezado que utilizan los desarrolladores para controlar el comportamiento del navegador y ofrece protección contra secuencias de comandos entre sitios (XSS). Este encabezado se diseñó para filtrar páginas y eliminar componentes inseguros o detener la carga al detectar secuencias de comandos entre sitios. Internet Explorer 8 introdujo X-XSS-Protection.

componentes o dejar de cargar cuando se detectaba un ataque de scripts entre sitios. Internet Explorer 8 introdujo la protección X-XSS.
encabezado y también fue compatible con las versiones Edge 12-16, Chrome 4-77, Opera 15-64 y Safari 5-15.3.
Los navegadores modernos ya no mantienen, admiten ni mejoran la implementación de este encabezado.

Tener esta configuración habilitada no implica necesariamente una vulnerabilidad, pero en algunos casos estos filtros se utilizan para realizar ataques XSS contra usuarios que de otro modo serían imposibles. Los filtros XSS del navegador utilizan expresiones regulares para modificar la respuesta. Por ejemplo, una expresión regular para detectar la etiqueta <script> modificaría la etiqueta en la respuesta a <sc#ipt>. Esto detendría el ataque XSS actual, pero también tendría consecuencias imprevistas. Un atacante podría usar este conocimiento para deshabilitar scripts legítimos o scripts del lado del cliente que contengan funciones de seguridad añadiendo un parámetro malicioso. Al configurar X-XSS-Protection en "1; mode=block", se detiene la carga de toda la página, pero es vulnerable a ataques de canal lateral [3].

Este encabezado se puede configurar explícitamente en la configuración del servidor de aplicaciones o en el código de la aplicación. Se deben verificar todos los lugares para detectar configuraciones inseguras de este encabezado. Esta verificación se reporta una vez por host de forma predeterminada y se puede configurar para que reporte todas las instancias configurando la entrada de verificación "X-XSS-Protection_Aggressive Reporting".

Ejecución:

Haga clic en la pestaña de respuesta de la solicitud resaltada. Verá que el encabezado X-XSS-Protection aparece resaltado y su valor es 1.

Implicación:

Configurar el encabezado X-XSS-Protection no es efectivo en navegadores que ya no lo admiten. Habilitarlo podría aumentar el riesgo de ataques de scripts entre sitios contra la aplicación si un usuario accede a ella desde un navegador antiguo que admitiera esta función.

Arreglar:

Eliminar el encabezado de respuesta X-XSS-Protection.
OpenText recomienda utilizar la Política de seguridad de contenido (CSP) para protegerse contra ataques XSS.

Referencia:

1 Protección X-XSS - HTTP MDN
https://developer.mozilla.org/es-ES/docs/Web/HTTP/Headers/X-XSS-Protection

2 filtros XSS de IE8
https://media.blackhat.com/bh-eu-10/presentations/Lindsay_Nava/BlackHat-EU-2010-Lindsay-Nava-IE8-XSS-Filters-slides.pdf

3. Abusar del auditor XSS de Chrome para robar tokens PortSwigger Research
https://portswigger.net/research/abusing-chromes-xss-auditor-to-steal-tokens

Nombres de archivos:
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec

Informativo — Fuga de información del sistema: externa

Resumen:

Se encontró una URL o un nombre de archivo en los comentarios del archivo.

Nombres de archivos:
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/listaExpresiones.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/badfile123.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/inicioErr.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec

Informativo — Campo oculto

Resumen:

Resumen:

Si bien evita que se muestre información en la propia página web, la información enviada a través de campos de formulario ocultos es fácilmente accesible y podría brindar a un atacante información valiosa que resultaría útil para escalar su metodología de ataque.
Las recomendaciones incluyen no confiar en campos de formulario ocultos como solución de seguridad para cualquier área de la aplicación web que contenga información confidencial o acceso a funcionalidades privilegiadas como la funcionalidad de administración del sitio remoto.

Ejecución:

Cualquier atacante podría eludir una solución de seguridad de campo de formulario oculto viendo el código fuente de esa página en particular.

Implicación:

El mayor peligro de la explotación de una vulnerabilidad en el diseño de un campo de formulario oculto es que el atacante obtendrá información que le ayudará a orquestar un ataque mucho más peligroso.

Arreglar:

No utilice campos de formulario ocultos para transmitir información confidencial ni para mantener el estado de la sesión. Una solución viable es cifrar los valores ocultos de un formulario y descifrarlos cuando una operación de base de datos o un script vaya a utilizar esa información. Desde el punto de vista de la seguridad, el mejor método para almacenar temporalmente la información requerida por los distintos formularios es utilizar una cookie de sesión.

Independientemente de si están ocultos o no, si su sitio utiliza valores enviados mediante un formulario para crear consultas a la base de datos, no dé por sentado que los datos no son maliciosos. En su lugar, siga las siguientes recomendaciones para depurar la información proporcionada por el usuario.

Defina estrictamente el tipo de datos (por ejemplo, una cadena, un carácter alfanumérico, etc.) que aceptará la aplicación.

Utilice lo que es bueno en lugar de lo que es malo.

Validar la entrada para detectar caracteres incorrectos.

No muestre mensajes de error al usuario final que proporcionen información (como nombres de tablas) que pueda utilizarse para orquestar un ataque.

Define el conjunto de caracteres permitido. Por ejemplo, si un campo debe recibir un número, permite que solo acepte números.

Define las longitudes de datos máximas y mínimas que aceptará la aplicación.

Especifique rangos numéricos aceptables para la entrada.

Nombres de archivos:

- http://172.16.21.39:7001/SisAdhereVerFirmas/organizacionPolitica/navegarRegistrar.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarReporteInactivos.reniec

---

Informativo

A menudo mal utilizado: Carga de archivos

Resumen:

Permitir que los usuarios carguen archivos puede permitir que los atacantes inyecten contenido peligroso o código malicioso para ejecutarlo en el servidor.

WebInspect ha detectado capacidades de carga de archivos en este servidor.

Como parte de la comprobación de carga de archivos peligrosos, WebInspect podría haber cargado varios archivos al servidor. Todos los archivos cargados como parte de esta prueba tienen nombres de archivo con el formato WebInspect_ccccc_File_nn.extension

Dónde:
ccccc = Número de identificación de verificación de
WebInspect nn = Número aleatorio único

Localice y elimine estos archivos en el servidor una vez completada la prueba.

Implicación:

Independientemente del lenguaje de programación, los ataques más devastadores suelen implicar la ejecución remota de código, donde un atacante logra ejecutar código malicioso en el contexto del programa. Si se permite a los atacantes subir archivos a un directorio accesible desde la web y hacer que estos se transmitan a un intérprete de código (p. ej., JSP/ASPX/PHP), pueden provocar que el código malicioso contenido en estos archivos se ejecute en el servidor.

Las implicaciones exactas dependen de la naturaleza de los archivos que un atacante pueda subir. Estas incluyen desde la publicación de contenido no autorizado hasta la facilitación de ataques de phishing, e incluso la vulneración total del servidor web.

Incluso si un programa almacena los archivos subidos en un directorio inaccesible desde la web, los atacantes podrían aprovechar la capacidad de introducir contenido malicioso en el servidor para lanzar otros ataques. Si el programa es susceptible a la manipulación de rutas, la inyección de comandos o vulnerabilidades de inclusión de archivos peligrosos, un atacante podría subir un archivo con contenido malicioso y hacer que el programa lo lea o lo ejecute aprovechando otra vulnerabilidad.

Arreglar:

No acepte archivos adjuntos si puede evitarlos. Si un programa debe aceptar archivos adjuntos, restrinja la capacidad de un atacante para proporcionar contenido malicioso aceptando únicamente los tipos específicos de contenido que el programa espera. La mayoría de los ataques que se basan en el contenido subido requieren que los atacantes puedan proporcionar el contenido que elijan. Imponer restricciones al contenido que el programa aceptará limitará considerablemente el alcance de posibles ataques. Compruebe los nombres, las extensiones y el contenido de los archivos para asegurarse de que sean los esperados y aceptables para la aplicación. Dificulte al atacante la determinación del nombre y la ubicación de los archivos subidos. Estas soluciones suelen ser específicas del programa y varían desde almacenar los archivos subidos en un directorio con un nombre generado a partir de un valor aleatorio fuerte al inicializar el programa, hasta asignar a cada archivo subido un nombre aleatorio y rastrearlos mediante entradas en una base de datos.

Asegúrese de que se realicen los siguientes pasos para desinfectar el archivo que se recibe:

Limite los tipos de archivos que se pueden subir. Por ejemplo, en una página de subida de imágenes, se debe rechazar cualquier archivo que no sea .jpg.

Asegúrese de que el usuario web no tenga ningún control sobre el nombre y la ubicación del archivo cargado en el servidor.
Nunca utilice el nombre que el usuario le asigna.
Nunca derive el nombre del archivo del nombre de usuario o del ID de sesión del usuario web.
No coloque el archivo en un directorio accesible para usuarios web. Es preferible que esta ubicación esté fuera de la raíz web.
Asegúrese de que se establezcan permisos estrictos tanto para el archivo cargado como para el directorio en el que se encuentra.
No permita permisos de ejecución en los archivos subidos. Si es posible, deniegue todos los permisos a todos los usuarios excepto al usuario de la aplicación web.

Verifique que el archivo subido contenga el contenido adecuado. Por ejemplo, un archivo JPEG subido debe tener un encabezado JPEG estándar.

Referencia:

Carga segura de archivos en aplicaciones web PHP

Mapeo de estándares - Enumeración de debilidades comunes - (CWE) CWE ID 434

Carga de archivos sin restricciones de OWASP

Nombres de archivos:
- http://172.16.21.39:7001/SisAdhereVerFirmas/lote/navegarCargarArchivo.reniec

Informativo

HTML5: Falta la política de seguridad de contenido

Resumen:

La Política de Seguridad de Contenido (CSP) es un encabezado de seguridad de respuesta HTTP que desarrolladores y arquitectos de seguridad pueden usar para crear una lista de dominios permitidos desde los cuales el sitio puede cargar recursos. Este encabezado proporciona protección integral contra vulnerabilidades críticas como el cross-site scripting y el clickjacking. Además, la CSP restringe la ejecución de JavaScript en línea, la evaluación dinámica de código JavaScript desde cadenas y la creación de marcos del sitio desde dominios externos. Si bien la CSP no reemplaza la validación de entrada, puede ayudar a reducir significativamente el riesgo de XSS debido a vulnerabilidades desconocidas. La directiva frame-ancestors de la CSP es equivalente a X-Frame-Options y restringe los dominios que pueden crear marcos para el contenido del sitio.

Ejecución:

Acceda al enlace http://172.16.21.39:7001/SisAdhereVerFirmas a través de un proxy y observe el encabezado CSP faltante en la respuesta.
De forma predeterminada, WebInspect marca solo una instancia de esta vulnerabilidad por host porque es típico configurar este encabezado en el nivel de host en una configuración de servidor.

Realice los siguientes pasos para marcar todas las instancias de este problema:

Cree una nueva política con la selección de comprobaciones que desee incluir en un nuevo análisis. Recomendamos usar la política "En blanco" o "Pasiva" como base.
Seleccione esta verificación y desmarque la entrada de verificación “FlagAtHost” de la descripción estándar.
Guardar la política.
Vuelva a escanear con esta nueva política personalizada.

Implicación:

Los arquitectos y desarrolladores de seguridad pueden aprovechar CSP para reducir significativamente el riesgo de ataques XSS y clickjacking. Los encabezados CSP pueden restringir la fuga de información a dominios externos al restringir los dominios desde los que el sitio puede cargar contenido al visualizarse en el navegador.

Arreglar:

Define una política CSP adecuada para tu sitio. Esta política puede configurarse mediante un encabezado de respuesta HTTP o una etiqueta <meta />.

Por ejemplo:
Política de seguridad de contenido: origen predeterminado https://example.net; origen secundario 'none';
O
<meta http-equiv="Política de seguridad del contenido" content="origen predeterminado https://cdn.example.net; origen secundario 'ninguno'; origen del objeto 'ninguno'">

La Política de Seguridad de Contenido 2 es el estándar recomendado. La Política de Seguridad de Contenido 3 está en borrador. A continuación, se muestra un resumen de la compatibilidad de los navegadores modernos con el encabezado CSP:

Edge: Versiones 15-18; compatible con un error de nonce. Versión 75 y posteriores; totalmente compatible.
Chrome: Versiones 36-38; faltan las directivas plugin-types, child-src, frame-ancestors, base-uri y form-action.
Versión 39; faltan las directivas plugin-types, child-src, base-uri y form-action. Versión 40 y posteriores; totalmente compatible.
Firefox: Versiones 31-34; faltan las directivas plugin-types, child-src, frame-ancestors, base-uri y form-action.
Versión 35; faltan las directivas plugin-types, child-src, frame-ancestors y form-action. Versiones 36-44: faltan las directivas plugin-types y child-src. Versión 45 y posteriores: falta la directiva plugin-types.

Además, la directiva report-uri puede configurarse para recibir informes de intentos de incumplimiento de la política. Estos informes pueden utilizarse como una indicación temprana de problemas de seguridad en el sitio, así como para optimizar la política.

Referencia:

Política de seguridad de contenido Nivel 3
Política de seguridad de contenido de OWASP
Documentos web de MDN
Guía de referencia rápida de la política de seguridad del contenido (CSP)

Nombres de archivos:
- http://172.16.21.39:7001/SisAdhereVerFirmas

Informativo

HLI: Bibliotecas detectadas

Resumen:

Hacker Level Insights proporciona a desarrolladores y profesionales de seguridad más contexto sobre la seguridad general de su aplicación. Durante este análisis, se detectó que la versión 1.8.14 de jQuery UI estaba en uso. Si bien estos hallazgos no necesariamente representan una vulnerabilidad de seguridad, es importante tener en cuenta que los atacantes suelen realizar un reconocimiento de su objetivo para identificar debilidades o patrones conocidos. Saber qué puede ver el hacker proporciona contexto que puede ayudar a los equipos a proteger mejor sus aplicaciones.

.cvc-grid {ancho:100%;}.cvc-box {radio del borde: 15px; borde: 2px sólido #1a75ff; ancho: 100%; relleno: 10px;}.cvc-cat {posición: relativa; visualización: bloque en línea; borde inferior: 1px negro punteado;}.cvc-cat .nvd-tooltip {visibilidad: oculta; ancho: 400px; color de fondo: #3b96ff; color: #fff; alineación del texto: izquierda; radio del borde: 6px; relleno: 5px; posición: absoluta; índice z: 1;}.cvc-cat:hover .nvd-tooltip {visibilidad: visible;}.score-box{color:#000!importante;color de fondo:#f1f1f1!

importante; ancho: 75%; radio del borde: 8 px}.score-box div {radio del borde: 8 px}.score-box div div{relleno izquierdo: 10 px; peso de fuente: negrita;}.score-box > div:after,.score-box > div:before {contenido: ""; mostrar: tabla; borrar: ambos}.sb-good{color: #fff! importante;color de fondo:#87e547!importante}.sb-warn{color:#fff!importante;color de fondo:#fbc02d!importante}.sb-bad {color:#fff!important;background-color:#ff8686!important}.nvd-tooltip:before{content:"La Base de Datos Nacional de Vulnerabilidades (NVD) ayuda al análisis HLI de Fortify WebInspect a encontrar información sobre CVE, incluyendo un breve resumen mediante consultas de Enumeración de Plataforma Común (CPE). El Instituto Nacional de Estándares y Tecnología (NIST) mantiene la base de datos."}

Registros NVD conocidos para: cpe:2.3:a:jqueryui:jquery_ui:1.8.14

VULNERABILIDAD LIDAD:	VERSIÓN ENCONTRADA:	FIJADO VERSIÓN:	SEVERO Gracias:
CVE-2010-5312	1.8.14	1.10.0	Medio

La vulnerabilidad de secuencias de comandos entre sitios (XSS) en jquery.ui.dialog.js en el widget Dialog en jQuery UI anterior a 1.10.0 permite a atacantes remotos inyectar secuencias de comandos web o HTML arbitrarios a través de la opción de título.

VULNERABILIDAD LIDAD:	VERSIÓN ENCONTRADA:	FIJADO VERSIÓN:	SEVERO Gracias:
CVE-2021-41182	1.8.14	1.13.0	Medio

jQuery-UI es la biblioteca oficial de interfaz de usuario de jQuery. Antes de la versión 1.13.0, aceptar el valor de la opción `altField` del widget Datepicker de fuentes no confiables podía ejecutar código no confiable. Este problema se solucionó en jQuery UI 1.13.0. Cualquier valor de cadena pasado a la opción `altField` ahora se trata como un selector CSS. Una solución alternativa es no aceptar el valor de la opción `altField` de fuentes no confiables.

VULNERABILIDAD LIDAD:	VERSIÓN ENCONTRADA: VERSIÓN CORRECTA:		SEVERO Gracias:
CVE-2021-41183	1.8.14	1.13.0	Medio

jQuery-UI es la biblioteca oficial de interfaz de usuario de jQuery. Antes de la versión 1.13.0, aceptar el valor de varias opciones `*Text` del widget Datepicker de fuentes no confiables podía ejecutar código no confiable. Este problema se solucionó en jQuery UI 1.13.0. Los valores pasados a varias opciones `*Text` ahora se tratan siempre como texto puro, no como HTML. Una solución alternativa es no aceptar el valor de las opciones `*Text` de fuentes no confiables.

HTML. Una solución alternativa es no aceptar el valor de las opciones `*Texto` de fuentes no confiables.

VERSIÓN ENCONTRADA:	FIJADO VERSIÓN:	SEVERO Gracias:
VULNERABILIDAD LIDAD:		
1.8.14	1.13.0	Medio

CVE-2021-41184

jQuery-UI es la biblioteca oficial de interfaz de usuario de jQuery. Antes de la versión 1.13.0, aceptar el valor de la opción `of` de la utilidad `.position()` de fuentes no confiables podía ejecutar código no confiable. Este problema se solucionó en jQuery UI 1.13.0. Cualquier valor de cadena pasado a la opción `of` ahora se trata como un selector CSS. Una solución alternativa es no aceptar el valor de la opción `of` de fuentes no confiables.

VERSIÓN ENCONTRADA:	FIJADO VERSIÓN:	SEVERO Gracias:
VULNERABILIDAD LIDAD:		
1.8.14	1.13.2	Medio

CVE-2022-31160

jQuery UI es un conjunto seleccionado de interacciones de interfaz de usuario, efectos, widgets y temas creados sobre jQuery.
Las versiones anteriores a la 1.13.2 son potencialmente vulnerables a ataques de scripts entre sitios. Inicializar un widget checkboxradio en una entrada dentro de una etiqueta hace que el contenido de esa etiqueta principal se considere como la etiqueta de entrada. Llamar a `.checkboxradio( "refresh" )` en dicho widget y el HTML inicial que contiene entidades HTML codificadas provocará que se decodifiquen erróneamente. Esto puede provocar la posible ejecución de código JavaScript. El error se ha corregido en jQuery UI 1.13.2. Para solucionar el problema, alguien que pueda modificar el HTML inicial puede encapsular todo el contenido no incluido en la etiqueta dentro de un `span`.

Nombres de archivos:

- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec

Mejores prácticas

Fallo de cumplimiento: falta de política de privacidad

Resumen:

La aplicación web no proporcionó una política de privacidad en el marco de esta auditoría. Muchas iniciativas legislativas exigen que las organizaciones incluyan un documento de acceso público en su aplicación web que defina la política de privacidad de su sitio web. Por regla general, estas políticas de privacidad deben detallar la información que recopila una organización, su finalidad, las posibles vías de divulgación y los métodos para abordar posibles quejas.

Varias leyes que rigen las políticas de privacidad incluyen la Ley Gramm-Leach-Bliley, la Ley de Portabilidad y Responsabilidad del Seguro Médico (HIPAA), la Ley de Protección de la Privacidad en Línea de California de 2003, la Directiva de Protección de Datos de la Unión Europea y otras.

Ejecución:

Todas las páginas web accesibles dentro del alcance del análisis se muestrean en busca de contenido textual que suele constituir una declaración de política de privacidad. Se informa de una infracción al finalizar el rastreo de la aplicación web sin encontrar coincidencias con ninguna de las páginas.

Tenga en cuenta que la política de privacidad de su aplicación podría estar ubicada en otro host o en una sección del sitio que no se configuró durante el análisis. Para validarla, intente acceder a la política de privacidad de su sitio web y compruebe si se incluyó en el análisis.

Implicación:

La mayoría de las leyes de privacidad se crean para proteger a los residentes que utilizan el sitio web. Por lo tanto, las organizaciones de cualquier parte del mundo deben cumplir con estas leyes si atienden a clientes que residen en estas áreas geográficas. De no hacerlo, podrían ser demandadas por el gobierno correspondiente.

Arreglar:

Declare una política de privacidad integral para el sitio web y asegúrese de que sea accesible desde todas las páginas que soliciten información personal de los usuarios. Para verificar la corrección, vuelva a escanear el sitio para detectar y auditar los recursos recién añadidos.

Descripciones:
Cualquier política de privacidad de una aplicación web estándar debe incluir los siguientes componentes:

Una descripción del propósito previsto para la recopilación de los datos.
Una descripción del uso de los datos.
Métodos para limitar el uso y divulgación de la información.
Una lista de los tipos de terceros a quienes se podría divulgar la información.
Información de contacto para consultas y quejas.

Referencia:

Ley de Protección de la Privacidad en Línea de California
http://oag.ca.gov/privacy/COPPA
Conferencia Nacional de Legislación Estatal

Conferencia Nacional de Legislación Estatal http://
www.ncsl.org/issues-research/telecom/state-laws-related-to-internet-privacy.aspx

Ley Gramm-Leach-Bliley http://
www.gpo.gov/fdsys/pkg/PLAW-106publ102/pdf/PLAW-106publ102.pdf

Ley de Portabilidad y Responsabilidad del Seguro Médico de 1996 https://www.cms.gov/
Regulations-and-Guidance/HIPAA-Administrative-Simplification/HIPAAGenInfo/downloads/HIPAALaw.pdf

Ley de Portabilidad y Responsabilidad del Seguro Médico de 1996 http://ec.europa.eu/
justice/policies/privacy/docs/guide/guide-ukingdom_en.pdf http://172.16.21.39:7001/

Nombres de archivos:
- SisAdhereVerFirmas/login.reniec

Mejores prácticas

Violación de la privacidad: Autocompletar

Resumen:

Los navegadores más recientes incluyen funciones que guardan el contenido de los campos de formulario introducidos por los usuarios y completan automáticamente la siguiente vez que se accede a ellos. Esta función está habilitada por defecto y podría filtrar información confidencial, ya que se almacena en el disco duro del usuario. El riesgo de este problema aumenta considerablemente si los usuarios acceden a la aplicación desde un entorno compartido. Se recomienda desactivar la función de autocompletar en todos los formularios.

Referencia:

Microsoft:
Seguridad de autocompletar

Nombres de archivos:
- http://172.16.21.39:7001/SisAdhereVerFirmas/login.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/proceso/navegarConsultar.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/organizacionPolitica/navegarConsultar.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/inicioErr.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/js/jquery/jquery.form.js
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporte/navegarVerificacionSemiAutomatica.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/configuracion/navegarIpAcceso.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/configuracion/navegarConfigurarReportes.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/proceso/navegarRegistrar.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporte/navegarVerificacionAutomatica.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarControlVerificaSemi.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/configuracion/navegarMotivo.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/configuracion/navegarOrigenSolcitud.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarReporteBusAdherente.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/lote/navegarCargarArchivo.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarReporteProdPeriodo.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarReporteProdUsuario.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarConsultaRegRecepcionados. reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/organizacionPolitica/navegarRegistrar.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/lote/navegarRegistrar.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/verificacionAutomatica/navegarConsultar.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarReportePerito.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/programacion/navegarPendientes.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/
- http://172.16.21.39:7001/SisAdhereVerFirmas/ordenProduccion/navegarPendientesAsignar.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarConsultaRegVerificadas.reniec

- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarConsultaOrgVerificadas.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporte/navegarRecepcion.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarReporteGeneral.reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/reporteConsulta/navegarReporteBusRepresentante. reniec
- http://172.16.21.39:7001/SisAdhereVerFirmas/ordenProduccion/navegarPendientesCrear.reniec