ZAP Informes de Escaneo

Sitio: http://localhost:5173

Generado vie, 1 mar 2024 18:02:30

ZAP Version: 2.14.0

Sumario de Alertas

Nivel de riesgo	Número de Alertas
Alto	1
Medio	4
Bajo	1
Informativo	2

Alertas

Nombre	Nivel de riesgo	Número de Instancias
Metadatos de la Nube Potencialmente Expuestos	Alto	1
Cabecera Content Security Policy (CSP) no configurada	Medio	3
Configuración Incorrecta Cross-Domain	Medio	6
Falta de cabecera Anti-Clickjacking	Medio	3
Hidden File Found (Archivo Oculto Encontrado)	Medio	4
X-Content-Type-Options Header Missing	Bajo	6
Divulgación de información - Comentarios sospechosos	Informativo	9
Modern Web Application	Informativo	3

Detalles de la Alerta

Alto	Metadatos de la Nube Potencialmente Expuestos
Descripción	El Ataque a los Metadatos de la Nube intenta abusar de un servidor NGINX mal configurado para acceder a la instancia de los metadatos mantenidos por proveedores de servicios en la nube como AWS, GCP y Azure. Todos estos proveedores proporcionan metadatos a través de una dirección IP interna no enrutable '169.254.169.254' - esta puede ser expuesta por servidores NGINX configurados incorrectamente y accedida utilizando esta dirección IP en el campo head Host. Traducción realizada con la versión gratuita del traductor www.DeepL.com/Translator
URL	http://localhost:5173/latest/meta-data/
Método	GET
Ataque	169.254.169.254
Evidencia	

Other Info	Según el código de estado de la respuesta correcta, es posible que se hayan devuelto metadatos de nube en la respuesta (response). Compruebe los datos de respuesta para ver si se ha devuelto algún metadato de nube. Los metadatos devueltos pueden incluir información que permitiría a un atacante comprometer completamente el sistema.
Instancia	1
Solución	No confíe en ningún dato de usuario en las configuraciones de NGINX. En este caso, probablemente sea el uso de la variable $host que se establece desde el encabezado (header) 'Host' y puede estar controlado por un atacante.
Referencia	https://www.nginx.com/blog/trust-no-one-perils-of-trusting-user-input/
CWE Id	
WASC Id	
Plugin Id	90034

Medio	**Cabecera Content Security Policy (CSP) no configurada**
Descripción	La Política de seguridad de contenido (CSP) es una capa adicional de seguridad que ayuda a detectar y mitigar ciertos tipos de ataques, incluidos Cross Site Scripting (XSS) y ataques de inyección de datos. Estos ataques se utilizan para todo, desde el robo de datos hasta la desfiguración del sitio o la distribución de malware. CSP proporciona un conjunto de encabezados HTTP estándar que permiten a los propietarios de sitios web declarar fuentes de contenido aprobadas que los navegadores deberían poder cargar en esa página; los tipos cubiertos son JavaScript, CSS, marcos HTML, fuentes, imágenes y objetos incrustados como applets de Java, ActiveX, archivos de audio y video.
URL	http://localhost:5173/
Método	GET
Ataque	
Evidencia	
Other Info	
URL	http://localhost:5173/robots.txt
Método	GET
Ataque	
Evidencia	
Other Info	
URL	http://localhost:5173/sitemap.xml
Método	GET
Ataque	
Evidencia	
Other Info	
Instancia	3
Solución	Asegúrese de que su servidor web, servidor de aplicaciones, balanceador de carga, etc. esté configurado para establecer la cabecera Content-Security-Policy.
Referencia	https://developer.mozilla.org/en-US/docs/Web/Security/CSP /Introducing_Content_Security_Policy https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html http://www.w3.org/TR/CSP/ http://w3c.github.io/webappsec/specs/content-security-policy/csp-specification.dev.html http://www.html5rocks.com/en/tutorials/security/content-security-policy/ http://caniuse.com/#feat=contentsecuritypolicy http://content-security-policy.com/

CWE Id	693
WASC Id	15
Plugin Id	10038

Medio	Configuración Incorrecta Cross-Domain
Descripción	Descargas de datos del navegador web podría ser posible, debido a una desconfiguración del intercambio de recursos cruzados de origen (CORS) en el servidor web
URL	http://localhost:5173/
Método	GET
Ataque	
Evidencia	Access-Control-Allow-Origin: *
Other Info	La desconfiguración CORS en el servidor web permite a dominios cruzados leer peticiones de dominios de terceros arbitrariamente, usando APIs sin autenticación en este dominio. Las implementaciones de navegador web no permiten a terceros arbitrarios leer la respuesta de APIs autenticados, de todas formas. Esto reduce el riesgo de alguna forma. Esta desconfiguración podría ser usada por un atacante para acceder a datos que está disponible en una manera sin autenticación, pero que usa otra forma de seguridad, tal como la dirección IP lista-blanca.
URL	http://localhost:5173/@vite/client
Método	GET
Ataque	
Evidencia	Access-Control-Allow-Origin: *
Other Info	La desconfiguración CORS en el servidor web permite a dominios cruzados leer peticiones de dominios de terceros arbitrariamente, usando APIs sin autenticación en este dominio. Las implementaciones de navegador web no permiten a terceros arbitrarios leer la respuesta de APIs autenticados, de todas formas. Esto reduce el riesgo de alguna forma. Esta desconfiguración podría ser usada por un atacante para acceder a datos que está disponible en una manera sin autenticación, pero que usa otra forma de seguridad, tal como la dirección IP lista-blanca.
URL	http://localhost:5173/robots.txt
Método	GET
Ataque	
Evidencia	Access-Control-Allow-Origin: *
Other Info	La desconfiguración CORS en el servidor web permite a dominios cruzados leer peticiones de dominios de terceros arbitrariamente, usando APIs sin autenticación en este dominio. Las implementaciones de navegador web no permiten a terceros arbitrarios leer la respuesta de APIs autenticados, de todas formas. Esto reduce el riesgo de alguna forma. Esta desconfiguración podría ser usada por un atacante para acceder a datos que está disponible en una manera sin autenticación, pero que usa otra forma de seguridad, tal como la dirección IP lista-blanca.
URL	http://localhost:5173/sitemap.xml
Método	GET
Ataque	
Evidencia	Access-Control-Allow-Origin: *
Other Info	La desconfiguración CORS en el servidor web permite a dominios cruzados leer peticiones de dominios de terceros arbitrariamente, usando APIs sin autenticación en este dominio. Las implementaciones de navegador web no permiten a terceros arbitrarios leer la respuesta de APIs autenticados, de todas formas. Esto reduce el riesgo de alguna forma. Esta desconfiguración podría ser usada por un atacante para acceder a datos que está disponible en una manera sin autenticación, pero que usa otra forma de seguridad, tal como la dirección IP lista-blanca.
URL	http://localhost:5173/src/main.jsx

Método	GET
Ataque	
Evidencia	Access-Control-Allow-Origin: *
Other Info	La desconfiguración CORS en el servidor web permite a dominios cruzados leer peticiones de dominios de terceros arbitrariamente, usando APIs sin autenticación en este dominio. Las implementaciones de navegador web no permiten a terceros arbitrarios leer la respuesta de APIs autenticados, de todas formas. Esto reduce el riesgo de alguna forma. Esta desconfiguración podría ser usada por un atacante para acceder a datos que está disponible en una manera sin autenticación, pero que usa otra forma de seguridad, tal como la dirección IP lista-blanca.
URL	http://localhost:5173/vite.svg
Método	GET
Ataque	
Evidencia	Access-Control-Allow-Origin: *
Other Info	La desconfiguración CORS en el servidor web permite a dominios cruzados leer peticiones de dominios de terceros arbitrariamente, usando APIs sin autenticación en este dominio. Las implementaciones de navegador web no permiten a terceros arbitrarios leer la respuesta de APIs autenticados, de todas formas. Esto reduce el riesgo de alguna forma. Esta desconfiguración podría ser usada por un atacante para acceder a datos que está disponible en una manera sin autenticación, pero que usa otra forma de seguridad, tal como la dirección IP lista-blanca.
Instancia	6
Solución	Asegúrese que los datos sensibles no están disponibles de manera no autenticada (usando dirección IP listado-blanco, por ejemplo). Configurar el encabezado HTTP ""Access-Control-Allow-Origin" a un conjunto de dominios más restrictivo, o remover completamente todos los encabezados CORS, para permitir que el navegador web refuerce la política de mismo origen (SOP) en una manera mas restrictiva.
Referencia	https://vulncat.fortify.com/en/detail?id=desc.config.dotnet.html5_overly_permissive_cors_policy
CWE Id	264
WASC Id	14
Plugin Id	10098

Medio	**Falta de cabecera Anti-Clickjacking**
Descripción	La respuesta no incluye Content-Security-Policy con la directiva 'frame-ancestors' ni X-Frame-Options para proteger contra ataques de 'ClickJacking'.
URL	http://localhost:5173/sitemap.xml
Método	GET
Ataque	
Evidencia	
Other Info	
URL	http://localhost:5173/
Método	GET
Ataque	
Evidencia	
Other Info	
URL	http://localhost:5173/robots.txt
Método	GET

Ataque	
Evidencia	
Other Info	
Instancia	3
Solución	Los navegadores web modernos admiten los encabezados HTTP Content-Security-Policy y X-Frame-Options. Asegúrese de que uno de ellos esté configurado en todas las páginas web devueltas por su sitio/aplicación. Si espera que la página esté enmarcada solo por páginas en su servidor (por ejemplo, es parte de un FRAMESET), querrá usar SAMEORIGIN; de lo contrario, si nunca espera que la página esté enmarcada, debe usar DENY. Alternativamente, considere implementar la directiva "frame-ancestros" de la política de seguridad de contenido.
Referencia	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
CWE Id	1021
WASC Id	15
Plugin Id	10020

Medio	**Hidden File Found (Archivo Oculto Encontrado)**
Descripción	Se identificó un archivo confidencial como accesible o disponible. Esto puede filtrar información administrativa, de configuración o de credenciales que puede ser aprovechada por un individuo malintencionado para atacar más adelante el sistema o mejorar la manera en que realiza ataques de ingeniería social.
URL	http://localhost:5173/._darcs
Método	GET
Ataque	
Evidencia	HTTP/1.1 200 OK
Other Info	
URL	http://localhost:5173/.bzr
Método	GET
Ataque	
Evidencia	HTTP/1.1 200 OK
Other Info	
URL	http://localhost:5173/.hg
Método	GET
Ataque	
Evidencia	HTTP/1.1 200 OK
Other Info	
URL	http://localhost:5173/BitKeeper
Método	GET
Ataque	
Evidencia	HTTP/1.1 200 OK
Other Info	
Instancia	4

Solución	Considera si este componente es realmente necesario en producción; si no es así, desactívelo. Si es así, asegurar que el acceso requiera la autenticación y autorización adecuadas, o limita la exposición solo a sistemas internos o IPs de origen definidas, etc.
Referencia	https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-on-Web-Servers.html
CWE Id	538
WASC Id	13
Plugin Id	40035

Bajo	**X-Content-Type-Options Header Missing**
Descripción	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
URL	http://localhost:5173/
Método	GET
Ataque	
Evidencia	
Other Info	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.
URL	http://localhost:5173/@vite/client
Método	GET
Ataque	
Evidencia	
Other Info	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.
URL	http://localhost:5173/robots.txt
Método	GET
Ataque	
Evidencia	
Other Info	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.
URL	http://localhost:5173/sitemap.xml
Método	GET
Ataque	
Evidencia	
Other Info	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.
URL	http://localhost:5173/src/main.jsx
Método	GET

Ataque	
Evidencia	
Other Info	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.
URL	http://localhost:5173/vite.svg
Método	GET
Ataque	
Evidencia	
Other Info	This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.
Instancia	6
Solución	Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application /web server to not perform MIME-sniffing.
Referencia	http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx https://owasp.org/www-community/Security_Headers
CWE Id	693
WASC Id	15
Plugin Id	10021

Informativo	**Divulgación de información - Comentarios sospechosos**
Descripción	La respuesta parece contener comentarios sospechosos que pueden ayudar a un atacante. Nota: Las coincidencias realizadas dentro de los scripts o archivos se refieren a todo el contenido, no solo a los comentarios.
URL	http://localhost:5173/
Método	GET
Ataque	
Evidencia	from
Other Info	The following pattern was used: \bFROM\b and was detected in the element starting with: "<script type="module"> import RefreshRuntime from "/@react-refresh" RefreshRuntime. injectIntoGlobalHook(window) window.$RefreshR", see evidence field for the suspicious comment/snippet.
URL	http://localhost:5173/@vite/client
Método	GET
Ataque	
Evidencia	debug
Other Info	The following pattern was used: \bDEBUG\b and was detected 6 times, the first in the element starting with: " this.hmrClient.logger.debug(`[vite] invalidate ${this. ownerPath}${message ? `: ${message}` : ''}`);", see evidence field for the suspicious comment/snippet.
URL	http://localhost:5173/@vite/client
Método	GET
Ataque	

Evidencia	from
Other Info	The following pattern was used: \bFROM\b and was detected 2 times, the first in the element starting with: " const el = Array.from(document.querySelectorAll('link')).find((e) => ! outdatedLinkTags.has(e) && cleanUrl(e.href", see evidence field for the suspicious comment/snippet.
URL	http://localhost:5173/@vite/client
Método	GET
Ataque	
Evidencia	query
Other Info	The following pattern was used: \bQUERY\b and was detected 2 times, the first in the element starting with: " const [acceptedPathWithoutQuery, query] = acceptedPath.split(`? `);", see evidence field for the suspicious comment/snippet.
URL	http://localhost:5173/@vite/client
Método	GET
Ataque	
Evidencia	TODO
Other Info	The following pattern was used: \bTODO\b and was detected in the element starting with: " // TODO Trigger their dispose callbacks.", see evidence field for the suspicious comment /snippet.
URL	http://localhost:5173/@vite/client
Método	GET
Ataque	
Evidencia	where
Other Info	The following pattern was used: \bWHERE\b and was detected in the element starting with: "// Allow `ErrorOverlay` to extend `HTMLElement` even in environments where", see evidence field for the suspicious comment/snippet.
URL	http://localhost:5173/robots.txt
Método	GET
Ataque	
Evidencia	from
Other Info	The following pattern was used: \bFROM\b and was detected in the element starting with: "<script type="module"> import RefreshRuntime from "/@react-refresh" RefreshRuntime. injectIntoGlobalHook(window) window.$RefreshR", see evidence field for the suspicious comment/snippet.
URL	http://localhost:5173/sitemap.xml
Método	GET
Ataque	
Evidencia	from
Other Info	The following pattern was used: \bFROM\b and was detected in the element starting with: "<script type="module"> import RefreshRuntime from "/@react-refresh" RefreshRuntime. injectIntoGlobalHook(window) window.$RefreshR", see evidence field for the suspicious comment/snippet.
URL	http://localhost:5173/src/main.jsx
Método	GET
Ataque	
Evidencia	from
Other	The following pattern was used: \bFROM\b and was detected 4 times, the first in the element starting with: "import __vite__cjsImport0_react_jsxDevRuntime from "

Info	/node_modules/.vite/deps/react_jsx-dev-runtime.js?v=a200ff96"; const jsxDEV ", see evidence field for the suspicious comment/snippet.
Instancia	9
Solución	Eliminar todos los comentarios que devuelvan información que podría ayudar a un atacante y arreglar cualquier problema subyacente al que se refieran.
Referencia	
CWE Id	200
WASC Id	13
Plugin Id	10027

Informativo	**Modern Web Application**
Descripción	The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.
URL	http://localhost:5173/
Método	GET
Ataque	
Evidencia	<script type="module"> import RefreshRuntime from "/@react-refresh" RefreshRuntime. injectIntoGlobalHook(window) window.$RefreshReg$ = () => {} window.$RefreshSig$ = () => (type) => type window.__vite_plugin_react_preamble_installed__ = true </script>
Other Info	No links have been found while there are scripts, which is an indication that this is a modern web application.
URL	http://localhost:5173/robots.txt
Método	GET
Ataque	
Evidencia	<script type="module"> import RefreshRuntime from "/@react-refresh" RefreshRuntime. injectIntoGlobalHook(window) window.$RefreshReg$ = () => {} window.$RefreshSig$ = () => (type) => type window.__vite_plugin_react_preamble_installed__ = true </script>
Other Info	No links have been found while there are scripts, which is an indication that this is a modern web application.
URL	http://localhost:5173/sitemap.xml
Método	GET
Ataque	
Evidencia	<script type="module"> import RefreshRuntime from "/@react-refresh" RefreshRuntime. injectIntoGlobalHook(window) window.$RefreshReg$ = () => {} window.$RefreshSig$ = () => (type) => type window.__vite_plugin_react_preamble_installed__ = true </script>
Other Info	No links have been found while there are scripts, which is an indication that this is a modern web application.
Instancia	3
Solución	This is an informational alert and so no changes are required.
Referencia	
CWE Id	
WASC Id	
Plugin Id	10109