



Reporte de pruebas de seguridad - Reto

Integración de seguridad informática en redes y sistemas de software

Grupo 402

José Antonio González Martínez - A01028517

Luis Daniel Filorio - A01028418

Gabriel Muñoz Luna - A01028774

Rodrigo Sosa Rojas - A01027913

Bajo la instrucción de

Carlos Enrique Vega Álvarez

Lizbeth Peralta Malváez

Osvaldo Cecilia Martínez

Edith Carolina Arias Serna

Fecha de entrega

13/10/2024

Índice

Objetivos del reporte	p. 3
Introducción	p. 3
Pruebas DAST	
Objetivos de las pruebas	p. 4
Enfoque de las pruebas	p. 4
Herramientas y técnicas utilizadas para el análisis dinámico	p. 4
Configuración de las herramientas	p. 4
Configuración del entorno	p. 4
Alcance de las pruebas	p. 4
Limitaciones	p. 4
Evidencia de la ejecución de las pruebas	p. 5
Resumen de hallazgos	p. 5
Sugerencias de mitigación	p. 5
Pruebas SAST	
Objetivos de las pruebas	p. 6
Enfoque de las pruebas	p. 6
Herramientas y técnicas utilizadas para el análisis dinámico	p. 6
Configuración de las herramientas	p. 6
Configuración del entorno	p. 6
Alcance de las pruebas	p. 6
Limitaciones	p. 6
Evidencia de la ejecución de las pruebas	p. 7
Resumen de hallazgos	p. 9
Impacto potencial	p. 9
Análisis comparativo entre DAST y SAST	p. 9
Conclusiones	p. 10
Anexos	p. 10
Referencias bibliográficas	p. 12

Objetivos generales del reporte

Este reporte tiene como objetivo el realizar pruebas a nuestro sistema CRM y a sus diferentes componentes con la finalidad de encontrar diferentes vulnerabilidades a las cuales nos podemos ver expuestos. Dichas pruebas serán realizadas con herramientas externas que nos permiten analizar dichas vulnerabilidades.

Introducción: descripción general (técnico y funcional) de la solución de software que fue evaluada

En las pruebas se analizará un sistema CRM, el cual cuenta con diferentes módulos como lo son:

- Inicio de sesión
- Base de datos
- API
- Dashboard

Este sistema tiene como objetivo dar un manejo más sencillo de las donaciones que se realizan a Fundación Sanders, mostrando las donaciones que se realizan, acceso a una lista de todos los usuarios registrados en su página, y a diferentes gráficas que nos muestran diferentes análisis de las donaciones y los usuarios. Para el acceso de los administradores se cuenta con una autenticación la cual utiliza JWT el cual le proporciona un token al usuario basado en su rol, esto permitiendo distinguir entre administradores y usuarios y los permisos de cada uno.

De igual manera, tenemos un inicio de sesión única para los usuarios que no pertenecen a la fundación, en el cual solo se les muestra información de sus donaciones.

Todos los datos se registran del sistema quedan guardados en una base de datos de mongoDB, en la cual los datos sensibles, como lo es la contraseña de los usuarios, están hasheados para la protección de los mismo

Pruebas DAST

Objetivos de las pruebas

El objetivo de estas pruebas es reconocer las vulnerabilidades de nuestro sistema utilizando una herramienta externa, así como calcular los posibles riesgos que estos puedan causar.

Enfoque de las pruebas

Nos enfocaremos en analizar las vulnerabilidades al recibir un ataque de NOSQL tanto en backend como en frontend. Buscamos detectar cualquier posible riesgo que el manejo de datos pueda exponer

Herramientas y técnicas utilizadas para el análisis dinámico

Utilizaremos la herramienta ZAP. Sus siglas significan “Zed Attack Proxy”, herramienta creada por OWASP que permite detectar vulnerabilidades en sitios Web. La herramienta también ofrece posibles soluciones para mitigar o eliminar el problema.

Configuración de las herramientas

La herramienta ZAP estaba instalada en nuestro dispositivo y se le dió la liga que debía analizar, esta dando resultado del backend y el frontend.

Configuración del entorno

Las pruebas se realizaron con una computadora con SO Windows 10. La versión de ZAP utilizada es la 2.15.0, la actualmente gestionada.


Alcance de las pruebas

Se hicieron dos pruebas, una llamando a la página y otra al backend. De esta forma pudimos evaluar las posibles vulnerabilidades de ambas secciones de nuestro sistema.


Limitaciones: cualquier restricción que pueda afectar los resultados.

Es importante mencionar que las pruebas fueron realizadas a los componentes en nuestro sistema. Estas pruebas están analizando única y exclusivamente aquellos componentes que se utilizan activamente y no todas las funciones de las dependencias.

Evidencia de la ejecución de las pruebas

URL:	https://localhost:5173/latest/meta-data/
Riesgo:	 High
Confianza:	Low
Parámetro:	
Ataque:	169.254.169.254
Evidencia:	
CWE ID:	0
WASC ID:	0
Origen:	Activo (90034 - Metadatos de la Nube Potencialmente Expuestos)
Vector de Entrada:	
Descripción:	<p>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.</p> <p>Todos estos proveedores proporcionan metadatos a través de una dirección IP interna no enrutable '169.254.169.254' - esta puede ser expuesta por</p>
Otra información:	<p>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.</p> <p>Los metadatos devueltos pueden incluir información que permitiría a un atacante comprometer completamente el sistema.</p>

Metadatos de la Nube Potencialmente Expuestos

URL:	https://172.29.165.223:5173/latest/meta-data/
Riesgo:	 High
Confianza:	Low
Parámetro:	
Ataque:	169.254.169.254
Evidencia:	
CWE ID:	0
WASC ID:	0
Origen:	Activo (90034 - Metadatos de la Nube Potencialmente Expuestos)

Resumen de hallazgos

Después de realizar las pruebas tanto en backend como en frontend, hemos detectado que tenemos sólo una vulnerabilidad llamada “Metadatos de la Nube Potencialmente Expuestos”. Este error sugiere que los pasos de información por la nube pueden ser accesibles por un atacante a través de una dirección IP desprotegida, llevando a un posible acceso no autorizado.

De acuerdo a la herramienta ZAP, este error se debe a un servidor NGINX mal configurado, lo que puede llevar a comprometer el sistema considerablemente, dejándolo en un riesgo alto con impacto alto

Sugerencias de mitigación.

La aplicación sugiere 1 posible solución que busca mitigar posibles ataques:

- Revisar las configuraciones de NGINX y asegurarnos de que ignore información proveniente de IPs enrutables. Esto nos permitirá cerrar la ventana de oportunidad de un ataque y así evitar exponer los metadatos.

Pruebas SAST

Objetivos de las pruebas.

En estas pruebas vamos a buscar, mediante la herramienta Snyk, algunas vulnerabilidades dentro de nuestro sistema, específicamente dentro de nuestro backend y frontend, esto con el fin de analizar los dos grandes componentes de nuestro sistema.

Enfoque de las pruebas.

Estas pruebas están enfocadas en analizar las dependencias que tenemos instaladas en nuestros módulos, encontrando vulnerabilidades en la implementación o en versiones de nuestro sistema.

Herramientas y técnicas utilizadas para el análisis estático.

Se va a utilizar la herramienta Snyk, la cual es una plataforma que ayuda a poder proteger y arreglar sistemas para desarrolladores, analiza la severidad de las vulnerabilidades y nos indica cómo podemos arreglarlas dentro de nuestras dependencias.

Configuración de las herramientas.

Para este análisis se instala el módulo de Snyk dentro de nuestro sistema dentro de la raíz de lo que se quiera analizar, en nuestro caso del front y el backend de nuestro sistema, permitiéndonos cubrir gran parte del proyecto realizado.

Configuración del entorno

Todo nuestro sistema está siendo probado en windows 11, nuestra herramienta Snyk tiene la versión 1.1293.1

Alcance de las pruebas.

Se van a realizar dos pruebas iguales, una en el frontend y otra en el backend del proyecto, esto para poder encontrar de igual forma las diferentes vulnerabilidades dentro de los dos grandes componentes de nuestro sistema

Limitaciones.

Es importante notar que las pruebas realizadas analizan las dependencias que incluyen nuestros proyectos, por lo que solo estamos analizando los módulos utilizados dentro de cada una de las secciones.

Evidencia de la ejecución de las pruebas.

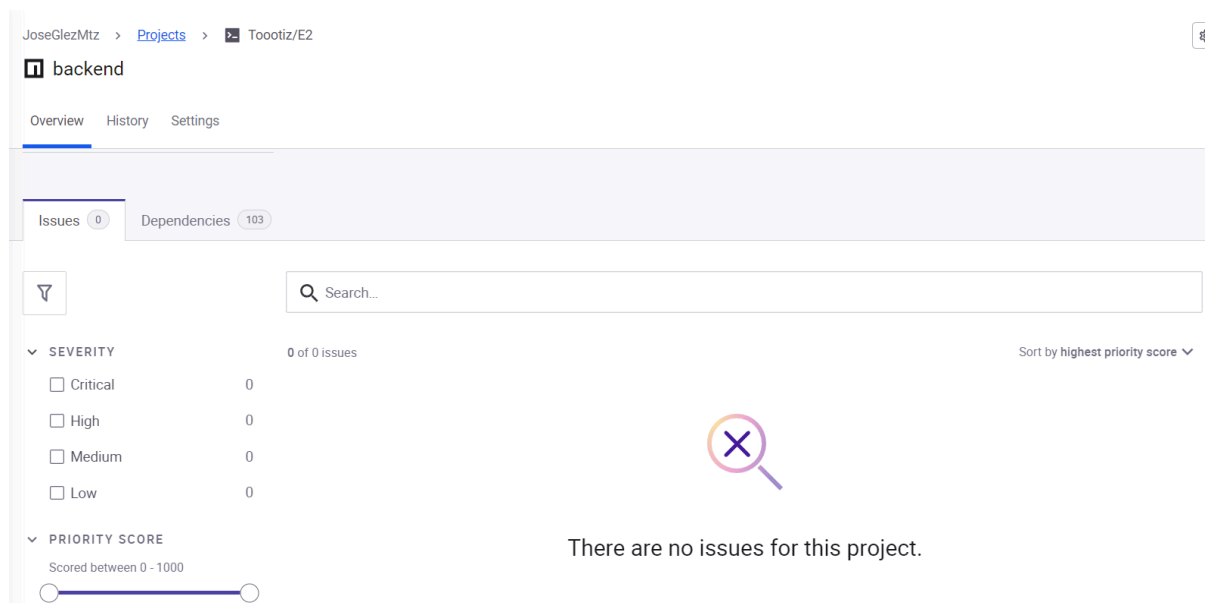
Backend

```
Testing /mnt/c/Users/jagle/OneDrive/Escritorio/Directorios/E2/Web/crm/backend...

Organization:    joseglezmtz-Vy63MakzjnCuUXr7W2Zkz4
Package manager: npm
Target file:     package-lock.json
Project name:    backend
Open source:     no
Project path:    /mnt/c/Users/jagle/OneDrive/Escritorio/Directorios/E2/Web/crm/backend
Licenses:        enabled

✓ Tested 103 dependencies for known issues, no vulnerable paths found.

Next steps:
- Run `snyk monitor` to be notified about new related vulnerabilities.
- Run `snyk test` as part of your CI/test.
```



En la parte de backend podemos ver que no encontramos vulnerabilidades dentro de nuestras dependencias, esto debido a que todo está actualizado y que está funcionando de manera segura.

Frontend

```
jose@LAP-JOSE:/mnt/c/Users/jagle/OneDrive/Escritorio/Directorios/E2/Web/crm/frontend$ snyk test

Testing /mnt/c/Users/jagle/OneDrive/Escritorio/Directorios/E2/Web/crm/frontend...

Organization:      joseglezmtz-Vy63MakzjnCuUXr7W2Zkz4
Package manager:  npm
Target file:       package-lock.json
Project name:      frontend
Open source:       no
Project path:      /mnt/c/Users/jagle/OneDrive/Escritorio/Directorios/E2/Web/crm/frontend
Licenses:          enabled

✓ Tested 171 dependencies for known issues, no vulnerable paths found.

Next steps:
- Run 'snyk monitor' to be notified about new related vulnerabilities.
- Run 'snyk test' as part of your CI/test.
```

frontend

Overview History Settings

🔍 Search...

▼ SEVERITY 0 of 0 issues Sort by highest priority score ▼

<input type="checkbox"/> Critical	0
<input type="checkbox"/> High	0
<input type="checkbox"/> Medium	0
<input type="checkbox"/> Low	0

▼ PRIORITY SCORE

Scored between 0 - 1000

0 ————— 1000

▼ "FIXED IN" AVAILABLE

<input type="checkbox"/> Yes	0
------------------------------	---

There are no issues for this project.

De igual manera vemos que la parte de frontend está libre de vulnerabilidades de dependencias

Estas pruebas nos permite ver que, en este caso los Node_Modules, se encuentran libre de vulnerabilidades por lo que no podrían ser explotados de alguna forma para poder dañar nuestro sistema

Resumen de hallazgos.

Tras haber realizado las pruebas podemos ver que el sistema no cuenta con vulnerabilidades dentro de las dependencias, sin embargo hay otros factores que podrían ser claves para detección de vulnerabilidades como lo es la conexión del frontend con el API o alguna falla en el envío de datos, estos factores son claves para encontrar vulnerabilidades pero no se pueden analizar con la herramienta seleccionada.

El análisis proporcionado por Snyk, nos ayuda mucho a poder ver el buen funcionamiento de nuestro sistema integrado con un buen uso de las dependencias.

Impacto potencial.

Las prueba Sast están enfocadas en encontrar vulnerabilidades dentro de las dependencias que usamos en todas nuestras secciones del proyecto, dichas dependencias nos ayudan con diferentes aspectos, como lo son el manejo de datos, el envío de los mismos entre otras cosas, por lo que el encontrar vulnerabilidades en el manejo de los datos podrían poner en peligro la integridad de nuestra base de datos o la confidencialidad de los mismos.

Al no encontrar vulnerabilidades con el análisis, podemos saber que nuestros datos están seguros tanto dentro de la base de datos como en el flujo entre secciones.

Análisis comparativo entre DAST y SAST.

Ambas de nuestras pruebas se basan en buscar diferentes vulnerabilidades mediante diferentes simulaciones, sin embargo las pruebas dast se basan en el código fuente y en el programa completo, mientras que las Sast las ejecutamos por separado en el front y el back del proyecto. Las pruebas Sast nos muestran vulnerabilidades dentro de nuestro programa y las Dast buscan posibles peligros de un ataque externo

Conclusiones.

Ya realizadas las pruebas podemos darnos cuenta que ambas pruebas nos ayudan a tener una perspectiva de cómo está desarrollado el proyecto evitando vulnerabilidades tanto en código fuente como en dependencias del proyecto, en este caso por la parte de las dependencias del proyecto no se nos mostró ninguna vulnerabilidad respecto a estas, sin embargo, somos conscientes de que por más que está prueba no arroje vulnerabilidades, estas aún existen dentro de este proyecto. Por otra parte, la prueba hecha al código fuente nos deja ver cómo podemos mejorar la seguridad de nuestro sistema con las configuraciones correctas.

Podemos concluir que estas pruebas de seguridad son fundamentales para garantizar la seguridad y la protección de datos en el sistema y que aunque responde a vulnerabilidades actuales en el sistema, también establece un proceso de mejora para futuras vulnerabilidades encontradas en el sistema.

Anexos: Reportes generados por las herramientas utilizadas.

A.1 - ZAP

Metadatos de la Nube Potencialmente Expuestos

Source	raised by an active scanner (Metadatos de la Nube Potencialmente Expuestos)
Reference	<ul style="list-style-type: none"> ▪ https://www.nginx.com/blog/trust-no-one-perils-of-trusting-user-input/

Cabecera Content Security Policy (CSP) no configurada

Source	raised by a passive scanner (Cabecera Content Security Policy (CSP) no configurada)
CWE ID	693
WASC ID	15
Reference	<ul style="list-style-type: none"> ▪ 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 ▪ https://www.w3.org/TR/CSP/ ▪ https://w3c.github.io/webappsec-csp/ ▪ https://web.dev/articles/csp ▪ https://caniuse.com/#feat=contentsecuritypolicy ▪ https://content-security-policy.com/

JoseGlezMtz > [Projects](#) > Tootiz/E2

backend

Overview History Settings

Issues 0

Dependencies 103

SEVERITY

0 of 0 issues

Sort by highest priority score

☐ Critical

0

☐ High

0

☐ Medium

0

☐ Low

0

PRIORITY SCORE

Scored between 0 - 1000

There are no issues for this project.

frontend

Overview History Settings

Issues 0

Dependencies 103

SEVERITY

0 of 0 issues

Sort by highest priority score

☐ Critical

0

☐ High

0

☐ Medium

0

☐ Low

0

PRIORITY SCORE

Scored between 0 - 1000

"FIXED IN" AVAILABLE

☐ Yes

0

There are no issues for this project.

Referencias bibliográficas

- Snyk. (s. f.). *Developer security* | Snyk. <https://snyk.io/>
- ZAP (s. f.). <https://www.zaproxy.org/>