

MÁSTER EN CIBERSEGURIDAD ONLINE

Identificación y Mitigación de Vulnerabilidades en Microservicios: Un Enfoque Práctico

TFM elaborado por: Andrew Steeve Ramirez Guzman

Tutor/a de TFM: Raimundo Alcázar Quesada

Bogotá, Colombia – 17/08/2024

Resumen

El presente Trabajo Fin de Máster se enfoca en la identificación y mitigación de vulnerabilidades dentro de arquitecturas de microservicios, un enfoque cada vez más popular en el desarrollo de software moderno. A través de un análisis exhaustivo de un sistema de microservicios, se han aplicado metodologías estándar para la detección de vulnerabilidades críticas que podrían comprometer la seguridad y la integridad de la información. El estudio incluye un análisis detallado de fases como el reconocimiento, escaneo, explotación y la redacción de un informe técnico con las medidas de mitigación sugeridas. Este enfoque práctico no solo identifica las debilidades en la infraestructura, sino que también propone soluciones viables y efectivas para fortalecer la seguridad del sistema.

Abstract

This master’s Thesis focuses on the identification and mitigation of vulnerabilities within microservices architectures, an increasingly popular approach in modern software development. Through an exhaustive analysis of a microservices system, standard methodologies have been applied to detect critical vulnerabilities that could compromise the security and integrity of information. The study includes a detailed analysis of phases such as reconnaissance, scanning, exploitation, and the drafting of a technical report with suggested mitigation measures. This practical approach not only identifies weaknesses in the infrastructure but also proposes viable and effective solutions to strengthen the system's security.

Índice

Lista de Figuras

Lista de Tablas

Introducción y Objetivos

Introducción

En la era digital actual, la ciberseguridad se ha convertido en una prioridad crítica para organizaciones de todo tipo y tamaño. Con el creciente uso de arquitecturas de microservicios, que permiten el desarrollo de aplicaciones modulares y escalables, también han surgido nuevos desafíos en términos de seguridad. Estas arquitecturas, si bien ofrecen ventajas significativas en términos de flexibilidad y despliegue, también presentan vulnerabilidades que pueden ser explotadas si no se gestionan adecuadamente.

La motivación para elegir el tema de este Trabajo Fin de Máster (TFM) radica en la necesidad imperiosa de comprender y abordar las vulnerabilidades inherentes a las arquitecturas de microservicios. A través de un enfoque práctico, este TFM busca no solo identificar estas vulnerabilidades, sino también proponer estrategias efectivas para su mitigación, contribuyendo así a la seguridad global de sistemas críticos en entornos empresariales.

La elección de este tema responde a un interés personal y profesional por las áreas de seguridad informática y desarrollo de software, donde la capacidad de proteger la integridad, confidencialidad y disponibilidad de los sistemas es esencial para el éxito de cualquier organización. Este trabajo pretende ser una contribución significativa al campo de la ciberseguridad, proporcionando un análisis detallado y soluciones prácticas aplicables a arquitecturas de microservicios.

Objetivos

Realizar un análisis exhaustivo de vulnerabilidades en sistemas basados en microservicios, identificando posibles amenazas y proponiendo soluciones efectivas para mitigar los riesgos asociados.

Objetivos Específicos

* **Identificación de Vulnerabilidades Críticas en Arquitecturas de Microservicios**
* Analizar un sistema de microservicios para identificar las principales vulnerabilidades que pueden comprometer la seguridad de este.  
  Utilizar herramientas y técnicas de análisis reconocidas para evaluar la seguridad del sistema.
* **Desarrollo de un Plan de Mitigación de Riesgos**
  + Proponer soluciones específicas para cada una de las vulnerabilidades identificadas, basadas en las mejores prácticas de la industria.
  + Desarrollar un plan de acción que detalle los pasos necesarios para implementar estas soluciones en un entorno real.
* **Evaluación de la Eficacia de las Medidas de Mitigación**
  + Medir el impacto de las soluciones propuestas en la seguridad del sistema, utilizando métricas y criterios de evaluación claros.
  + Ajustar las estrategias de mitigación según los resultados obtenidos, asegurando la máxima protección del sistema.

Desarrollo Del Contenido

1. Plantear el Escenario del Análisis
   1. Descripción de la Aplicación Web Vulnerable

La aplicación web vulnerable seleccionada para este análisis es un sistema de reservas en línea que permite a los usuarios gestionar reservas de hoteles y vuelos. Esta aplicación está diseñada utilizando una arquitectura de microservicios, que ofrece ventajas significativas en términos de escalabilidad y modularidad, pero también introduce varios desafíos de seguridad que necesitan ser abordados.

* + 1. Frontend de la Aplicación

El frontend de la aplicación está desarrollado utilizando tecnologías web estándar:

* **HTML**: Se utiliza para estructurar las páginas web, creando una interfaz de usuario intuitiva y fácil de navegar.
* **CSS**: Es utilizado para diseñar y estilizar la apariencia de la aplicación, asegurando una experiencia de usuario coherente y visualmente atractiva.
* **JavaScript**: Se emplea para agregar interactividad a la aplicación, permitiendo la manipulación dinámica de contenido y la gestión de eventos en el lado del cliente.

El frontend actúa como la interfaz principal a través de la cual los usuarios interactúan con el sistema, enviando solicitudes a los microservicios del backend y presentando la información de manera accesible.

* + 1. Backend de la Aplicación

El backend de la aplicación está desarrollado utilizando **Spring Boot**, un marco de trabajo basado en Java que facilita la creación de aplicaciones robustas y escalables. El backend se conecta a una base de datos **MySQL** centralizada, donde se almacena toda la información relevante, como los detalles de usuarios, reservas, disponibilidad de hoteles y vuelos, entre otros.

* + 1. Arquitectura del Sistema

La aplicación sigue una arquitectura de microservicios, lo que significa que las diferentes funcionalidades están separadas en servicios independientes que se comunican entre sí. Esta arquitectura se ilustra en la siguiente figura:

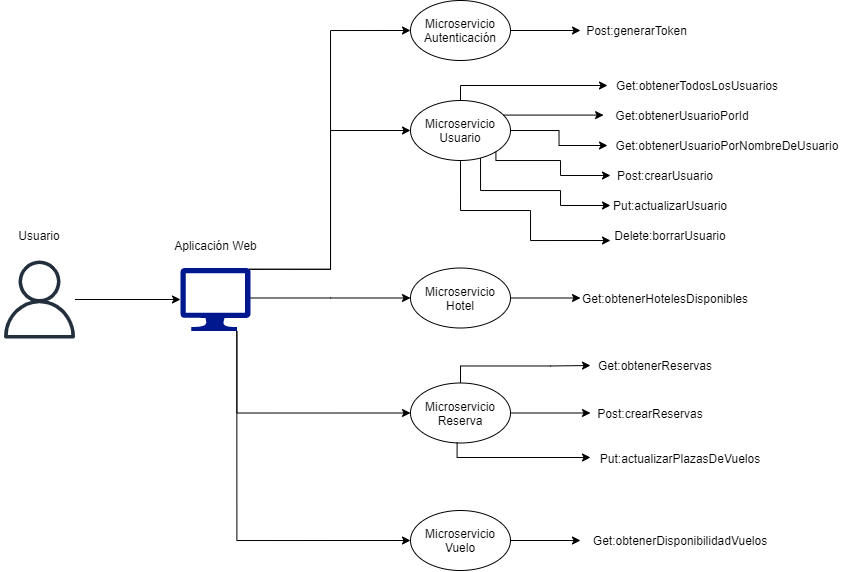


Figura 1. Esquema aplicación vulnerable

En esta arquitectura:

Los usuarios interactúan con la **Aplicación Web**, que actúa como un intermediario entre ellos y los microservicios.

* Cada microservicio tiene una función específica, lo que permite la modularidad y escalabilidad del sistema.
* Los **Microservicios** están conectados a una base de datos MySQL que almacena toda la información relevante.
  1. Objetivo del Análisis

El objetivo principal de este análisis es identificar y mitigar las vulnerabilidades presentes en la arquitectura de microservicios de la aplicación. A través de este análisis, se pretende mejorar la seguridad general del sistema, asegurando que las vulnerabilidades críticas sean abordadas y que las medidas de mitigación sean efectivas en proteger los datos y la integridad del sistema.

1. Listar las Acciones a Llevar a Cabo

Para realizar un análisis exhaustivo de las vulnerabilidades presentes en la aplicación web, se seguirán una serie de fases bien estructuradas. Cada fase está diseñada para abordar un aspecto específico del proceso de auditoría de seguridad, desde la preparación inicial hasta la implementación de un plan de acción basado en los hallazgos. A continuación, se describen las acciones que se llevarán a cabo en cada fase del análisis.

* 1. Fase de Preparación

La fase de preparación se centrará en obtener y preparar el entorno necesario para ejecutar la aplicación en contenedores Docker. Las acciones incluyen:

* **Descarga del archivo de configuración**: Se explicará cómo obtener el archivo necesario para configurar el entorno de contenedores Docker que alojará la aplicación.
* **Preparación del entorno con Docker**: Una vez obtenido el archivo, se procederá a preparar el entorno de contenedores que ejecutará la aplicación y sus microservicios.
  1. Fase de Reconocimiento

En esta fase, se identificará la superficie de ataque de la aplicación web. Las acciones incluyen:

* **Mapeo de la superficie de ataque**: Se identificará cada uno de los puntos de entrada de la aplicación web, como las APIs expuestas, formularios de entrada de datos, y cualquier otro punto por el que los usuarios puedan interactuar con la aplicación. El objetivo es descubrir posibles vectores de ataque que un atacante podría explotar.
  1. Fase de Escaneo

El enfoque de esta fase será detectar vulnerabilidades específicas en la aplicación web utilizando herramientas automatizadas y luego validar manualmente estos hallazgos. Las acciones incluyen:

* **Escaneo de vulnerabilidades en la aplicación web**: Se llevará a cabo un escaneo exhaustivo de la aplicación utilizando herramientas automatizadas para identificar posibles vulnerabilidades en los microservicios y en la interfaz web.
* **Análisis manual de resultados**: Tras el escaneo automatizado, se revisarán los resultados para eliminar falsos positivos y entender mejor las vulnerabilidades reales que afectan la aplicación web.
  1. Fase de Explotación

En esta fase, el objetivo será intentar explotar las vulnerabilidades identificadas en la aplicación web para evaluar su impacto sin. Las acciones incluyen:

* **Explotación de vulnerabilidades**: Se procederá a explotar manualmente las vulnerabilidades identificadas durante la fase de escaneo, simulando los pasos que un atacante seguiría para comprometer la seguridad de la aplicación.
* **Captura de evidencias**: Durante la explotación, se documentarán todas las pruebas exitosas con registros y otros datos relevantes que evidencien la explotación.
  1. Fase de Análisis

En esta fase, se evaluará el impacto de las vulnerabilidades descubiertas exclusivamente en la aplicación web. Las acciones incluyen:

* **Análisis de impacto**: Se evaluará el impacto de cada vulnerabilidad en la aplicación web, clasificándolas en función de su severidad, como crítica, alta, media o baja.
* **Priorización de la mitigación**: Se determinará qué vulnerabilidades deben ser corregidas de manera inmediata debido a su alto impacto en la aplicación web, y cuáles pueden ser mitigadas en un plazo más largo.
  1. Propuesta de Plan de Acción

Finalmente, se formulará un plan de acción detallado que guiará la implementación de las soluciones propuestas para mitigar las vulnerabilidades identificadas en la aplicación web. Las acciones incluyen:

* **Definición de un plan de acción**: Se propondrá un plan de acción que detalla los pasos específicos que deben seguirse para implementar las soluciones de seguridad recomendadas en la aplicación web. Este plan incluirá plazos, responsables, y recursos necesarios para garantizar la mejora de la seguridad del sistema.

1. Fase de Preparación

En esta fase, se llevará a cabo la configuración y despliegue de la aplicación en un entorno de contenedores Docker. El objetivo es garantizar que la aplicación esté completamente operativa, incluyendo la base de datos MySQL, antes de proceder con el análisis de vulnerabilidades. A continuación, se detallan los pasos necesarios para descargar y ejecutar la aplicación, así como para restaurar la base de datos en el contenedor correspondiente.

* 1. Descarga del archivo docker-compose.yml

El primer paso en la preparación del entorno es obtener el archivo docker-compose.yml, que contiene la configuración necesaria para levantar los servicios de la aplicación en contenedores Docker.

Para descargar el archivo, utiliza la herramienta curl, la cual permite realizar solicitudes a servidores HTTP y HTTPS para transferir datos. La siguiente instrucción permite descargar el archivo docker-compose.yml directamente desde el repositorio de GitHub:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Figura 2. Comando Curl para descargar archivo docker-compose.yml

Este archivo contiene las definiciones de los servicios necesarios para la ejecución de la aplicación, incluyendo la base de datos MySQL y otros microservicios.

* 1. Descargar las imágenes desde Docker Hub

Una vez descargado el archivo docker-compose.yml, es fundamental asegurarse de que las imágenes de Docker utilizadas en la configuración estén actualizadas.

Para garantizar que estás utilizando las versiones más recientes de las imágenes, utiliza el siguiente comando:

Patrón de fondo

Descripción generada automáticamente

Figura 3. Comando para actualizar versión de imágenes

Este comando descarga las versiones más recientes de las imágenes especificadas en el archivo docker-compose.yml desde Docker Hub.

* 1. Levantar los servicios en Docker

Con el archivo docker-compose.yml en tu sistema y las imágenes actualizadas, el siguiente paso es levantar los servicios en Docker.

Para iniciar todos los servicios definidos en el archivo docker-compose.yml, utiliza el siguiente comando:

Patrón de fondo

Descripción generada automáticamente

Figura 4. Comando para levantar ejecutar los servicios

El parámetro -d ejecuta los servicios en segundo plano, permitiéndote continuar utilizando la terminal para otros comandos.

Una vez que los servicios estén en funcionamiento, podrás acceder a la aplicación web a través de la siguiente URL <http://localhost:8080>

Esta URL te permitirá interactuar con la interfaz de usuario de la aplicación, que está siendo ejecutada en uno de los contenedores Docker.

* 1. Restaurar la base de datos MySQL en el contenedor

Con la aplicación en funcionamiento, el siguiente paso es restaurar la base de datos MySQL utilizando un archivo SQL de respaldo. Esto es crucial para que la aplicación disponga de los datos necesarios para operar correctamente.

Primero, es necesario copiar el archivo SQL de respaldo al contenedor Docker donde se está ejecutando MySQL. Utiliza el siguiente comando:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 5. Comando para copiar archivo SQL al contenedor

El archivo SQL puede ser descargado desde el siguiente enlace:

<https://raw.githubusercontent.com/GuzmanAndrew/ApuntesHackingCTF/master/Maestria%20IMF/M8.%20Trabajo%20Final%20de%20M%C3%A1ster/app_viajes.sql?token=GHSAT0AAAAAACV4GOL3S4BYTGIBKTU44WFQZWB6AKA>

Una vez que el archivo SQL ha sido copiado, debes acceder al contenedor MySQL para ejecutar el proceso de restauración. Utiliza el siguiente comando para acceder al contenedor:

**Patrón de fondo

Descripción generada automáticamente**

Figura 6. Comando para acceder al contenedor

Dentro del contenedor, utiliza el cliente MySQL para restaurar el archivo SQL. El comando es el siguiente:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 7. Comando para restaurar la base de datos

Esto restaurará la base de datos viajes con la información contenida en el archivo SQL.

Finalmente, si necesitas interactuar directamente con la base de datos MySQL desde el host, puedes hacerlo utilizando el siguiente comando:

Patrón de fondo

Descripción generada automáticamente

Figura 8. Comando conexión base de datos

Esto te permitirá conectarte al contenedor MySQL como si fuera una base de datos local, utilizando las credenciales apropiadas.