Desarrollo de un proceso replicable de seguridad en capas para aplicaciones web con Node.js: Identificación y mitigación de vulnerabilidades comunes.

Monografía para optar por el título de Ingeniería Electrónica y Telecomunicaciones
Modalidad Práctica Profesional

Andrés Felipe Diago Matta

Director: Msc. Erwin Meza Vega
Asesor de la empresa: Ing. Samara Catalina Enriquez Urbano

Universidad del Cauca
Facultad de Ingeniería en Electrónica y Telecomunicaciones
Programa de Ingeniería en Electrónica y Telecomunicaciones
Popayán, abril de 2023

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

LISTA DE FIGURAS

CNPq	–	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DS	–	Desenvolvimento Sustentável
ITV	–	Instituto Tecnológico Vale
MI	–	Mineração
UFOP	–	Universidade Federal de Ouro Preto

LISTA DE FIGURAS

Capítulo 1

Introduction

1.1. PLANTEAMIENTO DEL PROBLEMA

En los últimos años ha tenido lugar un crecimiento del uso de las Tecnologías de la Información y de las Comunicaciones (TIC). Esto ha provocado un gran incremento en el interés de las empresas por estar online y prestar sus servicios a través de plataformas digitales, actualizando sus herramientas de trabajo y digitalizando al máximo sus medios de producción. Este proceso de digitalización se ha visto potenciado por la proliferación de los servicios de almacenamiento y gestión en la nube, así como por la proliferación de dispositivos móviles inteligentes de bajo coste [1].

Por esta razón, es importante abordar el tema de la seguridad, ya que la sociedad actual usa un gran número de aplicaciones, desde la banca en línea y aplicaciones de trabajo remoto hasta el entretenimiento personal y el comercio electrónico. Es por esto que, las aplicaciones se convierten el objetivo principal de los atacantes, que se aprovechan de las vulnerabilidades como los fallos de diseño, así como de las debilidades de las API, el código abierto, los widgets de terceros y el control de acceso, buscando acceder a bases de datos, servidores y otros datos sensibles. Si se logra la exposición de datos sensibles, es posible lanzar ataques de referencia u otras formas de fraude en línea.

Los atacantes digitales tienen la ventaja de contar con el alcance y anonimato de quien los propicia, por lo que significa un gran reto para cualquier organización. Como prácticas de seguridad habituales se tiene el cambio de contraseñas, prohibir acceso a dispositivos y hacer una constante actualización de software. Pero esto no implica que la seguridad de las aplicaciones sea un factor que siempre se considere, siendo así un elemento por lo general ignorado y vulnerable. Esto genera una alta probabilidad de enfrentarse a amenazas que se generan por fallos del sistema a raíz de una inadecuada codificación, servidores mal configurados y un mal diseño en la aplicación.

Este trabajo de grado se va a desarrollar para la empresa WIZIT MIND BLOWING SOLUTIONS S.A.S, la cual es una empresa que se especializa en crear soluciones tecnológicas a la medida para sus clientes. El portafolio de servicios que ofrece, se encuentran soluciones en desarrollo de software, ciencia de datos, servicios en la nube, consultoría digital, entre otras líneas de negocio. Dentro de las soluciones que tienen

una relación con plataformas tecnológicas, digitalización de procesos y software que está en la web, se utiliza entre otras tecnologías Node.js, que es una plataforma de desarrollo muy popular para aplicaciones web, y con su creciente adopción, la seguridad de las aplicaciones desarrolladas en Node.js se ha convertido en una preocupación crítica.

Con lo mencionado anteriormente, este trabajo tiene como propósito la definición de un esquema/proceso replicable para la validación de pre-requisitos técnicos puede ayudar a minimizar la presencia de vulnerabilidades en estas aplicaciones, permitiendo un análisis de las vulnerabilidades comunes en aplicaciones desarrolladas con Node.js y la identificación de los pre-requisitos técnicos necesarios para prevenirlas. También implica la definición de un proceso replicable para validar la presencia de estos pre-requisitos técnicos en las plataformas de la empresa.

Capítulo 2

Marco Teórico

2.1. ESTADO DEL ARTE

A continuación, se describen algunos de los trabajos encontrados en la literatura disponible y relacionada con el objeto del presente trabajo de grado.

Investigaciones internacionales:

SEGURIDAD DE APLICACIONES WEB BASADAS EN LAS TECNOLOGÍAS NODE.JS Y MONGODB: ESTUDIO Y CASO DE USO [1] Este proyecto realizado en el año 2018, presenta las principales amenazas que se pueden encontrar en aplicaciones web desarrolladas mediante enfoques centrados en el lado del cliente, teniendo en cuenta la ubicuidad del cliente y el acceso a la información y plataformas mediante dispositivos móviles. Esto se realiza como parte del diseño y la implementación de aplicación web que busca controlar las mercancías y la logística de una empresa. Como parte de este proceso, se realiza un estudio de las vulnerabilidades de las aplicaciones cloud móviles, con el fin de que toda la solución software sea diseñada considerando las vulnerabilidades estudiadas a lo largo del trabajo, de forma que cuando se realice la implementación se han tenido en cuenta las principales amenazas a la seguridad de aplicaciones web desarrolladas en el lado del cliente, implementándose en cada caso las contramedidas necesarias.

Técnicas de mitigación para principales vulnerabilidades de seguridad en aplicaciones web [7] En este proyecto desarrollado en el año 2018, se analiza las principales vulnerabilidades que se encuentran en las aplicaciones web, y se propone el uso de diferentes técnicas para mitigar esas debilidades. El autor menciona que el activo más valioso de las organizaciones actualmente son los datos de los sistemas de información, y por esta razón, se hace de vital importancia para las empresas la seguridad de estos recursos frente a posibles ataques. Además, menciona que en el desarrollo de las aplicaciones web se debe tener en cuenta las amenazas que involucra el entorno web, por lo que existen vulnerabilidades en la codificación del aplicativo, en la transmisión de los datos, sus comunicaciones, el almacenamiento, controles de acceso, confidencialidad e integridad. Luego de realizar el análisis y listado de las vulnerabilidades, las cuales se agrupan en los temas de inyecciones; autenticación y gestión de sesiones; exposición de datos sensibles; control de acceso; y secuencia

de comandos en sitios cruzados. Se menciona que existen diversos mecanismos y técnicas de mitigación, por lo cual se establece unos mecanismos para mitigar cada una de esas debilidades listadas. Concluyendo que no existe un solo mecanismo para brindar seguridad en los sistemas, por lo que es necesario aplicar varias técnicas para mitigar las vulnerabilidades, las cuales deberán inicializar desde el proceso de desarrollo del aplicativo hasta la finalización del ciclo de vida.

Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web [5] En este estudio se menciona que hay un crecimiento en la complejidad de las tecnologías de la información, lo cual implica un mayor riesgo para los sistemas informáticos, teniendo como consecuencia el aumento en el número de ataques que se aprovechan de las vulnerabilidades. Por esta razón, se revisa algunas de las técnicas y herramientas utilizadas para la detección, realizando una matriz de trazabilidad entre ataques, vulnerabilidades, técnicas y herramientas que determinarán cuales debilidades pueden ser mitigadas con la utilización de dichas técnicas y herramientas. Para lograr esto, se utilizó el protocolo de la revisión sistemática, lo cual les permite realizar la matriz de trazabilidad, y hacer una propuesta para la utilización de herramientas para cada ataque basado en vulnerabilidades.

Seguridad web con NodeJS en un sistema de gestión de horarios laborales [3] En este trabajo llevado a cabo en el año 2017, como proyecto de fin de máster, el autor desarrolla una aplicación de gestión de horarios laborales, abordando los temas de seguridad web cuando se utiliza NodeJS y el framework ExpressJS. Se da una introducción al contexto del lenguaje javascript, el cual está viviendo una gran evolución que ha generado la aparición de muchos frameworks, como Angular, React o Vue para el frontend, o Node y Meteor para el backend. En el mismo tiempo de esta evolución, también han crecido los ataques informáticos como las filtraciones de datos confidenciales, ataques a servidores y capturas de información como WannaCry o Petya. Luego se define lo que es NodeJS, que es una tecnología diseñada para la ejecución de tareas por parte del servidor, contando con ciertas ventajas frente a otros sistemas tradicionales como lo es su característica de ser dirigido por eventos no bloqueantes. Mientras que ExpressJS, es un framework minimalista de NodeJS, que permite una flexibilidad al desarrollar. En el apartado de seguridad, es posible observar la identificación que se realiza de los ataques web más conocidos, y la forma como se busca evitar estas amenazas, como la utilización del estándar de JSON Web Tokens para la autenticación de usuarios, evitando los ataques de robo de sesión.

Seguridad en aplicaciones Web [4] En este trabajo desarrollado en el año 2015, el autor presenta la construcción de una aplicación web que permite analizar una determinada vulnerabilidad web en un servidor externo. Para este caso, se elige la SQL injection (inyección de código SQL), ya que es una de las amenazas más criticas y extendidas, por lo que el autor sugiere que es buena idea tener la posibilidad de realizar un análisis de SQL injection desde un navegador web, facilitando así las pruebas de penetración que se realizan en búsqueda de debilidades. Esta aplicación es capaz de realizar este análisis, procesar la información y almacenarla para el usuario. En contraste, el autor define esta vulnerabilidad, explicando el contexto donde este tipo de ataque sucede, y la importancia que tiene por su peligrosidad, siendo así el ataque

que OWASP declaró como el más crítico en un sistema. El autor también mencionan que, SQL es el lenguaje de acceso a base de datos que se caracteriza por su sencillez y gestores de bases de datos como MySQL, Oracle o SQLite, que ha provocado que este sea el más utilizado. Por esto, en el trabajo se menciona como protegerse frente a este ataque, donde se puede destacar que lo más importante es interpretar los datos recibidos en el servidor como código, lo cual es la base de todo el abanico de técnicas que existen.

Investigaciones nacionales:

AMENAZAS, VULNERABILIDADES, FACTORES DE RIESGO Y DEFENSA EN PROFUNDIDAD EN APLICACIONES WEB [6] En este artículo realizado en el año 2016, se explica la importancia que tiene para las organizaciones realizar o contar con una aplicación web, ya que estas permiten una comunicación activa entre el usuario y la información, demostrando como las organizaciones al momento de exponer sus servicios informáticos a redes de acceso tendrán que realizar un esfuerzo significativo para asegurar que la información y recursos están protegidos. Se define lo que es una aplicación web, los tipos de desarrollo que tienen las aplicaciones web y los tipos de estructuras en capas que se tiene, la cual consta de el navegador, el servidor donde se aloja el código, y la capa de base de datos. Además, se hace una identificación de las principales amenazas y cómo afectan a las aplicaciones, para aclarar las posibles vulnerabilidades y los controles adecuados que se pueden seguir en cada capa de riesgo. El autor concluye diciendo que la utilización de una metodología adecuada de gestión del riesgo y la implementación oportuna de controles reduce notablemente los incidentes de seguridad.

RETOS DE LA SEGURIDAD INFORMÁTICA EN SERVIDORES NODE JS [2] Este trabajo de grado llevado a cabo en el año 2016, menciona la evolución del lenguaje javascript para brindar una interfaz de usuario de fácil navegación, con dinamismo, para mejorar la experiencia de usuario. Considerando la importancia del nacimiento de NodeJS, que permite traer este lenguaje del lado del servidor para competir con otros lenguajes. Teniendo en cuenta esto, junto con las necesidades organizacionales por los desarrollos de páginas web, surgen retos en cuanto a la seguridad informática, ya que de esto depende la credibilidad de la organización. Sin embargo, el autor describe los riesgos y desventajas de javascript, que se pueden presentar por el echo de que este se utiliza del lado del cliente, y la falta de confidencialidad que genera. En este sentido, la seguridad en NodeJS depende de los desarrolladores, debido a las configuraciones predeterminadas mínimas y la arquitectura. Es por esto que el autor menciona el proyecto OWASP top 10, para describir las vulnerabilidades más importantes, y el projecto OWASP NodeGoat que permite permite conocer como se fucionan las vulnerabilidades aprojectos NodeJS.

El presente trabajo de grado contribuye con el área de interés en el área de los sistemas telemáticos. En particular lo relacionado con la seguridad de las aplicaciones web.

- Identificar las vulnerabilidades que se pueden generar cuando se desarrolla con NodeJS en conjunto con express.

- Propuesta de un modelo replicable para que minimice las vulnerabilidades cuando se desarrolla con NodeJS.

2.2. OBJETIVOS

2.2.1. Objetivo general

Definir un esquema/proceso replicable para la validación de pre-requisitos técnicos que minimicen la presencia de vulnerabilidades en plataformas desarrolladas en la empresa WIZIT MIND BLOWING SOLUTIONS S.A.S con Node.js.

2.2.2. Objetivos específicos

- Caracterizar y priorizar posibles vulnerabilidades que existen al desarrollar aplicaciones nodejs en conjunto con express.
- Generar recomendaciones para prevenir ataques a partir de vulnerabilidades en situaciones específicas[1].
- Desarrollar un piloto que aplique las reglas definidas.
- Evaluar la efectividad del esquema/proceso replicable propuesto en la prevención de vulnerabilidades en plataformas desarrolladas en la empresa con Node.js, a través de la aplicación piloto y la comparación de los resultados obtenidos antes y después de aplicar las reglas definidas.

2.3. METODOLOGÍA, ACTIVIDADES Y CRONOGRAMA

2.3.1. METODOLOGÍA

La metodología que se utilizará como referencia en el desarrollo del trabajo de grado en modalidad práctica profesional es la metodología del PMI. Este tipo de metodología proporciona un desarrollo del proyecto de manera gradual en la ejecución de los objetivos propuestos a través de procesos que se mencionan a continuación: Proceso de iniciación, Proceso de planificación, Proceso de ejecución, Proceso de supervisión y control, Proceso de cierre del proyecto.

2.3.2. FASES Y ACTIVIDADES

[1] Manual de reglas/ checklist, en concordancia con la norma ISO 27001

Tabla 2.1: Fases y Actividades

FASE	NOMBRE	ACTIVIDADES
I	Proceso de iniciación	1. Recopilar información sobre prácticas de seguridad en la empresa WIZIT MIND BLOWING SOLUTIONS S.A.S, y sobre la norma ISO 27001: Realizar entrevistas o reuniones con representantes de la empresa para conocer las prácticas de seguridad actuales, los procedimientos implementados y las políticas establecidas. 2. Identificar los sistemas y aplicaciones desarrollados en Node.js: Obtener un inventario de las plataformas y aplicaciones desarrolladas en Node.js por la empresa, así como su importancia y criticidad. 3. Analizar las vulnerabilidades comunes en aplicaciones web con Node.js: Realizar una revisión de la literatura y recopilar información sobre las vulnerabilidades más frecuentes que afectan a las aplicaciones web desarrolladas en Node.js. 4. Definir el alcance y los objetivos específicos del proyecto: Asegurarse de que los objetivos establecidos anteriormente sean claros y alineados con las necesidades de la empresa WIZIT MIND BLOWING SOLUTIONS S.A.S.
II	Proceso de Planificación	5. Establecer el alcance del proceso, indicando qué aplicaciones y plataformas de la empresa estarán cubiertas por el esquema de seguridad en capas. 6. Identificar las capas de seguridad y las medidas correspondientes. 7. Establecer criterios de selección de tecnologías y herramientas disponibles para el desarrollo con Node.js.

Continúa en la siguiente página

Tabla 2.1 – *Fases y Actividades*

FASE	**NOMBRE**	**ACTIVIDADES**
III	Proceso de Ejecución	8. Implementar las capas de seguridad seleccionadas, con las medidas de seguridad correspondientes en cada capa. 9. Implementar las herramientas y tecnologías de seguridad elegidas durante la fase de planificación. 10. Elaborar documentación detallada que describa las medidas de seguridad implementadas en cada capa, para su correcta aplicación en la empresa WIZIT MIND BLOWING SOLUTIONS S.A.S.
IV	Proceso de Supervisión	11. Evaluar el estado de seguridad de las aplicaciones y verificar la efectividad de las medidas implementadas. 12. Documentar los hallazgos, identificar áreas de mejora y proponer acciones correctivas o preventivas.
V	Proceso de Cierre del Proyecto	13. Entrega de documentos (Documento final y archivos del modelo replicable). 14. Entrega del modelo de seguridad en capas a la empresa WIZIT MIND BLOWING SOLUTIONS S.A.S. 15. Preparación y realización de la sustentación.

2.3.3. CRONOGRAMA

Actividades/ Tiempo		MES 1				MES 2				MES 3				MES 4				MES 5				MES 6				MES 7				MES 8				MES 9			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	**Proceso de iniciación**																																				
1.1	Recopilar información sobre prácticas de seguridad en la empresa WIZIT MIND BLOWING SOLUTIONS S.A.S, y sobre la norma ISO 27001: Realizar entrevistas o reuniones con representantes de la empresa para conocer las prácticas de seguridad actuales, los procedimientos implementados y las políticas establecidas.																																				
1.2	Identificar los sistemas y aplicaciones desarrollados en Node.js: Obtener un inventario de las plataformas y aplicaciones desarrolladas en Node.js por la empresa, así como su importancia y criticidad.																																				
1.3	Analizar las vulnerabilidades comunes en aplicaciones web con Node.js: Realizar una revisión de la literatura y recopilar información sobre las vulnerabilidades más frecuentes que afectan a las aplicaciones web desarrolladas en Node.js.																																				
1.4	Definir el alcance y los objetivos específicos del proyecto: Asegurarse de que los objetivos establecidos anteriormente sean claros y alineados con las necesidades de la empresa WIZIT MIND BLOWING SOLUTIONS S.A.S.																																				
2	**Proceso de Planificación**																																				
2.2	Establecer el alcance del proceso, indicando qué aplicaciones y plataformas de la empresa estarán cubiertas por el esquema de seguridad en capas.																																				
2.3	Identificar las capas de seguridad y las medidas correspondientes.																																				
2.4	Establecer criterios de selección de tecnologías y herramientas disponibles para el desarrollo con Node.js.																																				
3	**Proceso de Ejecución**																																				
3.1	Implementar las capas de seguridad seleccionadas, con las medidas de seguridad correspondientes en cada capa.																																				
3.2	Implementar las herramientas y tecnologías de seguridad elegidas durante la fase de planificación.																																				
3.3	Elaborar documentación detallada que describa las medidas de seguridad implementadas en cada capa, para su correcta aplicación en la empresa WIZIT MIND BLOWING SOLUTIONS S.A.S.																																				
4	**Proceso de Supervisión**																																				
4.1	Evaluar el estado de seguridad de las aplicaciones y verificar la efectividad de las medidas implementadas.																																				
4.2	Documentar los hallazgos, identificar áreas de mejora y proponer acciones correctivas o preventivas.																																				
5	**Proceso de Cierre del Proyecto**																																				
5.1	Entrega de documentos (Documento final y archivos del modelo replicable).																																				
5.2	Entrega del modelo de seguridad en capas a la empresa WIZIT MIND BLOWING SOLUTIONS S.A.S.																																				
5.3	Preparación y realización de la sustentación.																																				

2.4. RECURSOS, PRESUPUESTO Y FUENTES DE FINANCIACIÓN

En la tabla 2.2 se presenta el presupuesto que se utilizará durante la realización del trabajo de grado para una duración de 9 meses, con base en los criterios de la *Guía de Elaboración de Anteproyectos de Trabajo de Grado - Modalidad Trabajo de Investigación* de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca. Para el cálculo de los valores de los recursos humanos se consideró la última actualización del valor por punto para los empleados públicos docentes, según el decreto 885 del 2 de junio de 2023 y cuyo valor corresponde a $18,845.00 COP por hora. El presupuesto se planteó de acuerdo con las siguientes características:

Director: Msc. Erwin Meza Vega, con disponibilidad de 2 horas semanales durante la realización del proyecto y un reconocimiento de 2,5 puntos la hora.

Estudiante: Andrés Felipe Diago Matta, con disponibilidad de 30 horas semanales y un reconocimiento de 1,5 puntos la hora.

Tabla 2.2: Recursos y presupuesto del trabajo de grado.

RECURSOS	**FUENTES**		**TOTAL**
	ESTUDIANTES	**FIET**	
Director		$3.392.100	$3.392.100
Co-director			
Estudiantes	$61.057.800		$61.057.800
RECURSOS TÉCNICOS			
Recursos Hardware			
Utilización Equipo		$2.367.504	$2.367.504
Impresora	$300.00		$300.00
Otros	$250.00		$250.00
Recursos Software			
OptSim		$3.200.000	$3.200.000
Matlab		$2.000.000	$2.000.000
Recursos Bibliográficos			
Documentación	$460.00		$460.00
Recursos Varios			
SUBTOTAL	$62.067.800	$10.959.604	$72.567.404
AUI (20 %)	$12.413.560	$2.191.921	$14.513.481
TOTAL	$74.481.360	$13.151.525	$87.080.885

2.5. CONDICIONES DE ENTREGA

- **Monografía:** Documento donde se evidencia el trabajo elaborado para alcanzar los objetivos propuestos. Contiene la base conceptual de conocimiento, la descripción detallada del problema, la propuesta y los resultados obtenidos mediante la ejecución del proyecto de grado.
- **Anexos:** Documentación relacionada con el modelo desarrollado, no incluido en la monografía.
- **Artículo:** Documento donde se muestran los resultados y aportes planteados para el trabajo de investigación, en formato IEEE.

Bibliografía

[1] Pedro Alberto Ruiz González. «SEGURIDAD DE APLICACIONES WEB BASADAS EN LAS TECNOLOGÍAS NODE.JS Y MONGODB: ESTUDIO Y CASO DE USO». En: (2018). URL: https://repositorio.uam.es/bitstream/handle/10486/684807/ruiz_gonzalez_pedroalberto_tfm.pdf?sequence=1&isAllowed=y.

[2] Arias Melo Yeison Hernando. «RETOS DE LA SEGURIDAD INFORMÁTICA EN SERVIDORES NODE JS». En: (2016). URL: http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/2671/Trabajo%20de%20grado.pdf?sequence=1&isAllowed=y.

[3] Jorge Martínez Lascorz. «Seguridad web con NodeJS en un sistema de gestión de horarios laborales». En: (2017). URL: https://oa.upm.es/48283/8/TFM_JORGE_MARTINEZ_LASCORZ.pdf.

[4] Diego Losada Regos. «Seguridad en aplicaciones Web». En: (2015). URL: https://openaccess.uoc.edu/bitstream/10609/42542/6/dlosadarTFM0615Memoria.pdf.

[5] Ana Laura Hernández Saucedo y Jezreel Mejia Miranda. «Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web». En: (Year). URL: https://www.redalyc.org/pdf/5122/512251501005.pdf.

[6] Carlos Andrés Ortegón Serna. «AMENAZAS, VULNERABILIDADES, FACTORES DE RIESGO Y DEFENSA EN PROFUNDIDAD EN APLICACIONES WEB». En: (2016). URL: http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/4913/00005093.pdf?sequence=1&isAllowed=y.

[7] Alex Zambrano et al. «Técnicas de mitigación para principales vulnerabilidades de seguridad en aplicaciones web». En: (2018). URL: https://www.researchgate.net/profile/Teresa-Guarda/publication/331178479_Mitigation_techniques_for_security_vulnerabilities_in_web_applications/links/5fabe891a6fdcc331b9478b4/Mitigation-techniques-for-security-vulnerabilities-in-web-applications.pdf.

2.6. ACTA DE PROPIEDAD INTELECTUAL