Fecha de elaboración: 10.04.2023			
Tipo de documento	TID: **X**	Obra Creación:	Proyecto Investigación:
Título	**Implementación de remediaciones de vulnerabilidades identificadas en aplicaciones desarrolladas en PHP (V 7.4.33) sobre un servidor con sistema operativo Linux (Ubuntu), mediante escaneos de seguridad (OWASP) realizados en una empresa pyme de Call Center**		
Autor(es)	**María Roció Camargo Villa y Miguel Alejandro Barreto Fonseca**		
Tutor(es)	**Fabio González Mendieta**		
Fecha de finalización	31.03.2023		
Temática	**Análisis de vulnerabilidades**		
Tipo de investigación	**Descriptiva, cualitativa**		

Resumen

La investigación tuvo como objetivo analizar las principales vulnerabilidades que se presentan en un servidor con sistema operativo Linux Ubuntu instalado y en una aplicación web programada en lenguaje PHP (V 7.4.33) desde hace más de 10 años, de acuerdo al top 10 de OWASP, el escaneo se realizó con las herramientas nmap y Owasp Zap para descubrir estas vulnerabilidades, la interpretación de cada una, nivel de impacto de acuerdo al CVE y la mejor manera de remediarlas.

Palabras clave

Ataques, Call Center, CVE, CSRF, Confidencialidad, Cross-site Scripting (XSS), Disponibilidad, Integridad, Lenguaje de Programación, Linux, Nmap, OWASP, Página Web, PHP, Remediaciones, Riesgo, Seguridad de la Información, SQL Injection, Top 10 de Vulnerabilidades, Ubuntu, Vulnerabilidades.

Planteamiento del problema

Es constante las vulnerabilidades que se encuentran en los sistemas, en el caso de Linux, por ser un sistema operativo reconocido por su flexibilidad en el uso de aplicaciones para empresas, es un objetivo codiciado por los atacantes para robar, eliminar, modificar

información o denegar servicios. A pesar de que algunas de ellas se sanean con los parches de seguridad que publica el proveedor, para otras es necesario realizar una serie de ajustes en configuraciones y aplicar buenas prácticas de programación en las aplicaciones desarrolladas, para evitar algún tipo de ataque.

De acuerdo con Areito (2008) informa que “el número de incidentes de seguridad en internet se incrementan continuamente. Los incidentes están relacionados con la explotación de vulnerabilidades dirigidas a comprometer la seguridad de un sistema o de una red” por lo que es importante para una empresa cerrar las posibles brechas de seguridad para que su negocio no sea interrumpido.

Pregunta

¿Cuáles posibles remediaciones se pueden dar a las vulnerabilidades identificadas en las aplicaciones desarrolladas en lenguaje PHP (V 7.4.33) y alojadas en un servidor con sistema operativo Linux (Ubuntu)?

Objetivos

Describir las posibles remediaciones de las vulnerabilidades identificadas en aplicaciones desarrolladas en PHP (V 7.4.33) sobre un servidor con sistema operativo Linux (Ubuntu), mediante escaneos de seguridad (OWASP) realizados en una empresa pyme de Call Center.

Marco teórico

El estudio implica la realización de un escaneo utilizando herramientas como Nmap y Owasp Zap, para obtener información relevante sobre posibles vulnerabilidades en el sistema y las aplicaciones.

Una vez obtenidos los resultados del escaneo, se realizará una interpretación y clasificación de los mismos. Además, se plantean las mejores prácticas para sanear las vulnerabilidades encontradas, tomando como guía el top 10 de OWASP (Open Web

Application Security Project) y el sistema de puntaje de CVE (Common Vulnerabilities and Exposures).

JPMorgan (2014) los ciberdelincuentes lograron acceder a la red del banco y robar datos de nombres, dirección, teléfono, emails a 83 Millones de clientes que se usaron en suplantación de identidad, estafas o fraudes.

Ashely Madisson: (2015) hackearon esta página web de citas extramatrimoniales exigiendo el cierre de la página, expandieron los datos de más de 37 Millones de usuarios en todo el mundo. Se encontró múltiples vulnerabilidades en la página, una de ellas permitía el uso de contraseñas débiles, muchos usuarios usaban 123456 o la palabra password como clave.

Página(s): 14

Método

Sampieri (2014) proponen un modelo general de investigación de enfoque cualitativo, que consta de seis pasos: planteamiento del problema, marco teórico, diseño de la investigación, recolección de datos, análisis de datos, conclusiones y recomendaciones. En su libro "Metodología de la investigación", los autores explican que el planteamiento del problema implica definir la cuestión de investigación y formular preguntas relevantes. El marco teórico, por su parte, consiste en revisar la literatura sobre el tema para identificar conceptos y teorías importantes. En cuanto al diseño de la investigación, los autores destacan la importancia de seleccionar una metodología y técnicas de recolección de datos apropiadas para responder a las preguntas de investigación. El siguiente paso es la recolección de datos, la cual se realizará con la herramienta de escaneo de vulnerabilidades. Una vez obtenidos los datos, se procede al análisis de los

mismos para identificar analizar y comparar los hallazgos encontrados. Finalmente, se presentan las conclusiones de la investigación y se formulan recomendaciones para la remediación de las vulnerabilidades.

Según la norma ISO 27001:2005 proporciona un marco de trabajo con un enfoque cualitativo, para la gestión de la seguridad de la información. En general, el proceso de escaneo de vulnerabilidades debe ser parte del proceso de gestión de riesgos de seguridad de la información y se deben seguir las mejores prácticas para garantizar que se realice de manera efectiva y eficiente, para ello se debe seguir las siguientes fases:

- Identificación de activos críticos
- Selección de herramientas de escaneo de vulnerabilidades
- Escaneos de vulnerabilidades
- Análisis y corrección de vulnerabilidades
- Monitoreo continuo

Página(s): 23

Resultados, hallazgos u obra realizada

Se identificó diferencias en los resultados de las herramientas de escaneo de vulnerabilidades que se realizó sobre la aplicación web, en el desarrollo de este proyecto usamos Owasp Zap, en la organización internamente usan la herramienta Vega y el proveedor contratado por la organización para realizar escaneo con un conjunto de herramientas, entre ellas nexus y acunetix, por lo que la herramienta de Owasp Zap arrojaron diferentes vulnerabilidades a las reportadas por el proveedor, esto se debe a que OWASP ZAP, Vega y Nexus son herramientas de escaneo de vulnerabilidades web

que utilizan diferentes técnicas y algoritmos para identificar y reportar vulnerabilidades en una aplicación web.

Cada herramienta tiene sus propios métodos de escaneo y su propio conjunto de reglas y patrones para detectar vulnerabilidades.

Además, cada herramienta se actualiza de forma independiente y puede tener diferentes versiones o bases de datos de vulnerabilidades. Por lo tanto, es común que estas herramientas muestran diferentes resultados en un escaneo de vulnerabilidades de una misma aplicación web.

Es importante destacar que las herramientas de escaneo de vulnerabilidades no son infalibles y pueden tener limitaciones en la detección de ciertos tipos de vulnerabilidades o en la identificación de falsos positivos o negativos. Por lo tanto, se recomienda utilizar varias herramientas y complementar el escaneo con pruebas manuales para obtener una evaluación más completa de la seguridad de una aplicación web.

Página(s): 29

Conclusiones

Según el análisis realizado con la herramienta Nmap y la herramienta Owasp-Zap, se identificaron varias vulnerabilidades en el servidor web y en la aplicación web de la compañía. Estas vulnerabilidades representan riesgos para la seguridad de la infraestructura y la aplicación, lo que podría permitir a los atacantes acceder, manipular o divulgar información confidencial, así como ejecutar código malicioso.

El análisis realizado reveló varias vulnerabilidades que requieren atención y remedios para fortalecer la seguridad de la infraestructura y la aplicación web. Se recomienda implementar las soluciones propuestas y seguir las mejores prácticas de seguridad para mitigar los riesgos identificados. Además, es fundamental mantener los sistemas actualizados y aplicar parches de seguridad de manera regular para evitar futuras vulnerabilidades conocidas.

Actualmente se cuenta con variedad de herramientas que permiten realizar un escaneo de vulnerabilidades a los sistemas de información, por ello, las empresas deben concientizarse sobre la importancia de tener personal calificado y ético para realizar estas actividades periódicas con el fin de evitar fuga o pérdidas de información.

Documentar buenas prácticas de desarrollo para la creación y modificación de aplicaciones web, en este caso en particular, se evidenció que es una aplicación que se desarrolló hace más de 10 años, por lo que es importante al menos una vez al año realizar escaneo a todas las aplicaciones con el fin de mantener, tanto el código, como la información de la base de datos, segura, íntegra y disponible. Posterior a la remediación se debe volver a escanear para confirmar si las vulnerabilidades fueron remediadas correctamente.

Es importante tener presente que se descubren nuevas vulnerabilidades, por lo que se es necesario estar suscrito en algún boletín de alertas, por ejemplo al de incibe-cert y estar consultado la página de CVE: https://cve.mitre.org con el fin de mantenerse actualizado y estar alerta ante cualquier eventualidad que ponga en riesgo los sistemas de información.

Los fabricantes de sistemas operativos publican parches de seguridad cuando descubren una amenaza, por lo que se debe mantener los sistemas operativos parchados o actualizados a la última versión estable disponible.

La alta gerencia debe conocer y apoyar los procesos de seguridad de la información, de acuerdo a Carpentier (2016) “El impacto de las diferentes amenazas varía considerablemente según el efecto sobre la empresa, algunas tienen un impacto sobre la confidencialidad o la integridad de datos, otras actúan sobre la disponibilidad de los sistemas” (P. 43). Si un riesgo se materializa conlleva a gastos económicos y/o inversión de tiempo, al afectar la triada de la seguridad de la información puede llevar a una mala reputación de la compañía (mala imagen), en contraste, para una compañía es más solvente tener un sistema blindado a perder la información por algún tipo evento y césar actividades ocasionando una pérdida económica o reputacional que impactante significativamente a la compañía.

Productos derivados

Implementación de remediaciones de vulnerabilidades identificadas en aplicaciones desarrolladas en PHP (V 7.4.33) sobre un servidor con sistema operativo Linux (Ubuntu), mediante escaneos de seguridad (OWASP) realizados en una empresa pyme de Call Center

Maria Rocío Camargo Villa
Cód. 12226006

Miguel Alejandro Barreto Fonseca
Cód. 12226024

Corporación Universitaria Unitec
Especialización en seguridad de la información
Seminario de Investigación II

Bogotá, Colombia
10 de abril de 2023

Implementación de remediaciones de vulnerabilidades identificadas en aplicaciones desarrolladas en PHP (V 7.4.33) sobre un servidor con sistema operativo Linux (Ubuntu), mediante escaneos de seguridad (OWASP) realizados en una empresa pyme de Call Center

Maria Rocío Camargo Villa
Cód. 12226006

Miguel Alejandro Barreto Fonseca
Cód. 12226024

Fabio Antonio González
Director

Corporación Universitaria Unitec
Especialización en seguridad de la información
Seminario de Investigación II

Bogotá, Colombia
10 de abril de 2023

Tabla de contenido

Tabla de figuras

Resumen

Se analizará las principales vulnerabilidades que se presentan en un servidor con sistema operativo Linux Ubuntu instalado y en una aplicación web programada en lenguaje PHP (V 7.4.33) desde hace más de 10 años, de acuerdo al top 10 de OWASP, el escaneo se realizó con las herramientas nmap y Owasp Zap para descubrir estas vulnerabilidades, la interpretación de cada una, nivel de impacto de acuerdo al CVE y la mejor manera de remediarlas.

Linux es un sistema operativo muy requerido por las organizaciones por su adaptabilidad y fiabilidad. Ubuntu Server es una de sus distribuciones, Gutiérrez (2015), afirmó que: “Ubuntu es uno de los sistemas operativos de base LINUX, este sistema operativo se distribuye como software libre, de entre los sistemas de software libre basados en este lenguaje este sea probablemente el más orientado al gran público, debido a su facilidad de uso. Está distribuido por Canonical y la fundación Ubuntu, Esta distribución ocupa casi la mitad de la cuota de los usuarios que se deciden por sistemas basados en LINUX, uno de los usos del sistema Ubuntu es configurarlo para desarrollar y publicar páginas web y aplicaciones de acuerdo con el Core del negocio, por esto es muy apetecido por los atacantes.

En el mundo virtual existen gran variedad de aplicaciones web desarrolladas en diferentes lenguajes y personas dedicadas a encontrar los puntos débiles, ya sea para fortalecerlos o sacar algún tipo de provecho monetario o de reputación. Sea cual sea la intención se encuentran las vulnerabilidades y es un trabajo constante estarlas saneando para mantener la información segura y evitar fuga, robo y/o pérdida del activo más valioso: La información.

Areito (2008), escribió que: “La seguridad de los sistemas de información es una disciplina en continua evolución, la meta final de la seguridad es permitir que una organización cumpla con todos sus objetivos de negocio o misión”, se resalta

la importancia y relevancia que tienen la seguridad de la información sobre el cumplimiento de los objetivos y misión de una compañía, una compañía sin seguridad de la información está destinada a ser presa fácil de los atacantes. Con la escala de puntajes del CVE, se clasificará las vulnerabilidades que se descubran en Linux y con OWASP (documento donde la fundación OWASP publica el listado de las vulnerabilidades que encuentran en las aplicaciones web, esta es actualizada cada 3 o 4 años) se analizará el top 10 de vulnerabilidades más importantes y de más impacto enfocándose a las aplicaciones desarrolladas en lenguajes PHP.

Palabras Clave

Ataques, Call Center, CVE, CSRF, Confidencialidad, Cross-site Scripting (XSS), Disponibilidad, Integridad, Lenguaje de Programación, Linux, Nmap, OWASP, Página Web, PHP, Remediaciones, Riesgo, Seguridad de la Información, SQL Injection, Top 10 de Vulnerabilidades, Ubuntu, Vulnerabilidades.

1. Problema de investigación

Es constante las vulnerabilidades que se encuentran en los sistemas, en el caso de Linux, por ser un sistema operativo reconocido por su flexibilidad en el uso de aplicaciones para empresas, es un objetivo codiciado por los atacantes para robar, eliminar, modificar información o denegar servicios. A pesar de que algunas de ellas se sanean con los parches de seguridad que publica el proveedor, para otras es necesario realizar una serie de ajustes en configuraciones y aplicar buenas prácticas de programación en las aplicaciones desarrolladas, para evitar algún tipo de ataque.

El sitio web w3techs.com (2023), realizo un análisis del uso de lenguajes de programación para la creación de sitios web, los resultados demuestran que PHP es el lenguaje de programación más utilizado en la creación de sitios web, con un 77.5% de uso, algunas de las razones por la que los desarrolladores prefieren PHP es por su facilidad de uso, tiene un comunidad activa que contribuye a proporcionar recursos y herramientas que facilitan el desarrollo, es eficaz y escalable, lo que lo hace una opción popular para empresas y desarrolladores que buscan construir aplicaciones web rápidas y escalables, pero también es popular entre los hackers por ellos las aplicaciones desarrolladas en PHP están expuestas a una variedad vulnerabilidades de seguridad, como SQL Injection, cross-site scripting (XSS), falsificación de solicitudes entre sitios (CSRF), entre otras. Los desarrolladores deben implementar las mejores prácticas de seguridad y realizar pruebas de vulnerabilidad periódicas para garantizar que sus aplicaciones estén protegidas contra estos riesgos.

De acuerdo con Areitio (2008) informa que “el número de incidentes de seguridad en internet se incrementan continuamente. Los incidentes están relacionados con la explotación de vulnerabilidades dirigidas a comprometer la

seguridad de un sistema o de una red” por lo que es importante para una empresa cerrar las posibles brechas de seguridad para que su negocio no sea interrumpido.

2. Justificación

A lo largo de la historia se escuchan noticias de grandes robos de información, ya sea a grandes, medianas o pequeñas empresas, algunas con el objetivo de dañarles la reputación, de sacar algún beneficio monetario o sencillamente para demostrar que los atacantes pueden hacerlo, por esto y con la evolución de la tecnologías de la información, aparecen las normas y/o estándares (ISO 27001 – gestión de vulnerabilidades) que nos dan pautas para mantener nuestra infraestructura asegurada. Por esto se debe estar atento, es necesario estar actualizando, revisando, probando, escaneando los sistemas de información, con el fin de evitar lo más posible que esto suceda.

Urbina (2016) afirma que “Es tan valiosa la información, tanto en el ámbito personal como en el de las empresas, que es considerada un activo en extremo valioso; por esta razón, siempre está expuesta a amenazas. Imaginemos, aunque sea por un momento, que somos poseedores de grandes sumas de dinero en efectivo u oro que conservamos en nuestra casa, ¿qué es lo primero que haríamos para proteger esos activos? Sin duda, lo que haríamos sería resguardarlos lo más posible, a fin de evitar un robo, por la sencilla razón de que son demasiado valiosos. Y quien los roba o sustrae obtiene un beneficio personal y daña seriamente nuestro patrimonio” por lo cual, la información, ya sea personal o empresarial es muy valiosa y debemos protegerla de las personas mal intencionadas y las amenazas que actualmente se presentan.

3. Pregunta de investigación

¿Cuáles posibles remediaciones se pueden dar a las vulnerabilidades identificadas en las aplicaciones desarrolladas en lenguaje PHP (V 7.4.33) y alojadas en un servidor con sistema operativo Linux (Ubuntu)?

4. Objetivos

4.1 Objetivo General

Describir las posibles remediaciones de las vulnerabilidades identificadas en aplicaciones desarrolladas en PHP (V 7.4.33) sobre un servidor con sistema operativo Linux (Ubuntu), mediante escaneos de seguridad (OWASP) realizados en una empresa pyme de Call Center.

4.2 Objetivos Específicos

- Escanear los sistemas para descubrir las vulnerabilidades
- Describir las principales vulnerabilidades que se presentan en servidores con sistema operativo Linux y aplicaciones desarrolladas en lenguaje PHP (V 7.4.33) de acuerdo con el top 10 publicado por la fundación OWASP.
- Clasificar las vulnerabilidades de acuerdo con el CVSS.
- Interpretar los resultados de la herramienta.
- Plantear las soluciones para sanear las vulnerabilidades.

5. Marco teórico y estado del arte

5.1 Alcance

Es un estudio tipo descriptivo, porque se busca presentar información detallada sobre las principales vulnerabilidades que pueden afectar un servidor con sistema operativo Linux y las aplicaciones de un Call Center desarrolladas en lenguaje PHP (V 7.4.33). El estudio implica la realización de un escaneo utilizando herramientas como Nmap y Owasp Zap, para obtener información relevante sobre posibles vulnerabilidades en el sistema y las aplicaciones.

Una vez obtenidos los resultados del escaneo, se realizará una interpretación y clasificación de los mismos. Además, se plantean las mejores prácticas para sanear las vulnerabilidades encontradas, tomando como guía el top 10 de OWASP (Open Web Application Security Project) y el sistema de puntaje de CVE (Common Vulnerabilities and Exposures).

5.2 Marco conceptual

Linux

Es una serie de sistemas operativos, en su mayoría de versión libre, con o sin entorno gráfico, si se usa en modo consola el consumo de recursos de la máquina se minimiza, su mayor uso es en controlar servidores con tareas específicas.

Torvalds (1991) Linux es una de las tecnologías más importantes y revolucionarias de nuestra era. Es un sistema operativo libre y gratuito que ha transformado la informática, la ciencia y la sociedad en general, y que sigue siendo un ejemplo inspirador de cómo la cooperación y el intercambio de conocimientos pueden crear cosas sorprendentes y poderosas.

Ubuntu

Sistema operativo de código abierto de Linux basado en Debian que incluye software libre, existen varias versiones: Desktop, Cloud, Phone, Tablet y Server. El software de Ubuntu Server es sin entorno gráfico, solo líneas de comando con el fin de que el servidor use sus recursos para su función principal.

Shuttleworth (2023) Ubuntu es más que un sistema operativo, es una comunidad. Nuestra misión es llevar el software libre al mundo, y hacerlo accesible para todos.

PHP

Procesador de hipertexto, lenguaje de programación de código abierto que se puede incrustar en HTML y se usa para desarrollo de aplicaciones web. PHP puede emplearse en los sistemas operativos principales incluyendo Linux.

Sklar (2014) PHP es un lenguaje de programación popular y potente que puede utilizarse para crear una amplia variedad de aplicaciones web y aplicaciones de línea de comandos.

Owasp

Es una fundación que por medio de software de código abierto trabaja para mejorar la seguridad en el sistema operativo y aplicaciones. OWASP TOP 10 Es un listado de las 10 principales vulnerabilidades de seguridad en aplicaciones web que publica la Fundación en su página web, esta es actualizada cada año. Adicionalmente, publican una guía de desarrollo web seguro y guías para testeo de aplicaciones web.

Cvss

En español sus siglas significan Sistema de Puntuación de Vulnerabilidad Común, es una escala de valoración de riesgo y gravedad en seguridad de los sistemas con las siguientes métricas:

Base: rango de 0 a 10

Temporal: sus medidas son: Explotabilidad, nivel de remediación y confianza en el informe

Del entorno: son las características de la vulnerabilidad afectadas por el entorno del usuario: Potencial de daños colaterales, distribución de destino, requisito de confidencialidad, requisitos de integridad, requisitos de disponibilidad.

CVE

Sus siglas significan Common Vulnerabilities and Exposures, lo que traduce: vulnerabilidades y exposiciones comunes, es una lista de vulnerabilidades comunes, donde se informa la calificación de riesgo, los sistemas operativos que afecta, posibles soluciones para mitigarlas.

Mitre Corporation (2021) El CVE (Common Vulnerabilities and Exposures) es un diccionario público de vulnerabilidades de seguridad de software mantenida por la organización Mitre Corporation. Cada CVE asigna un identificador único, compuesto por números y letras, a una vulnerabilidad específica de software, lo que permite a los usuarios de todo el mundo identificar y referirse a la misma vulnerabilidad en sus sistemas y herramientas de seguridad.

5.3 Marco Legal

Ley 1273 de 2009 (05 enero de 2009):

"De la protección de la información y los datos"" en esta ley detallan los atentados informáticos que una organización puede afectarse si no realiza una buena práctica de análisis y remediación de vulnerabilidades en los sistemas y las penalidades de una persona si llega a incurre en estos, consta de dos capítulos y diez artículos.

En su capítulo primero informan "De los atentados contra la confidencialidad, la integridad y la disponibilidad de los datos y de los sistemas informáticos" los artículos que competen a esta investigación relacionada con las vulnerabilidades informáticas son:

Artículo 269A. Acceso abusivo a un sistema informático: informa que aquel que ingrese sin autorización a un sistema o parte de él, tendrá una pena de prisión de 48 a 96 meses y una multa de 100 a 1000 smmlv.

Artículo 269C. Interceptación de datos informáticos: si alguien, sin orden judicial intercepte datos informáticos durante el transporte de este, tendrá una pena de prisión de 36 a 72 meses.

Artículo 269D Daño informático: si se destruye, altere, borre, deteriore datos informáticos, sin tener el derecho para hacerlo, tendrá una pena de prisión de 48 a 96 meses y una multa de 100 a 1000 smmlv.

Artículo 269E Uso de software malicioso: el que produzca, distribuya, venda, envíe virus o programas que puedan afectar un sistema, tendrá una pena de prisión de 48 a 96 meses y una multa de 100 a 1000 smmlv.

Articulo 269F Violación de datos personales: aquel que sin derecho**,** obtenga, venda, intercambie, envíe, compre o realice cualquier transferencia con

datos que ha obtenido sin autorización de bases de datos, archivos o ficheros, tendrá una pena de prisión de 48 a 96 meses y una multa de 100 a 1000 smmlv.

Artículo 269G Suplantación de sitio web para capturar de datos personales: Aquel que desarrolle, ejecute, transacciones o envié programas, enlaces o ventanas emergentes sin autorización, modifique nombres de dominio para re direccionar a otras direcciones ip, por ejemplo, en páginas de bancos u otro sitio web, tendrá una pena de prisión de 48 a 96 meses y una multa de 100 a 1000 smmlv.

Artículo 269H Circunstancias de agravación punitiva: en este artículo relacionan las acciones que agravan la pena de los artículos antes mencionados, algunas de ellas son: si los delitos son cometidos por un servidor público, si se obtiene beneficio propio o de un tercero, si se realiza en redes de sistemas estatales u oficiales, sector financiero, nacionales o extranjeros, entre otros.

Ley Estatutaria 1266 de 2008: (diciembre de 2008)

En esta ley se describen las disposiciones reglamentarias del hábeas data y el manejo de la información contenida en las bases de datos comerciales, crediticios, personales, financieras y de servicios.

5.4 Antecedentes teóricos

El uso del internet cada día va en aumento y esto lleva a que gran variedad de empresas se vean en la necesidad de dar a conocer su negocio por medio de una página web y así mismo, publican aplicaciones para consultar o auto gestionar los servicios que ofrecen; La mayoría de las aplicaciones funcionan consultando bases de datos exponiendo la información a quien la necesite, por lo tanto, es fundamental tener políticas de seguridad para controlar los accesos a los sistemas y no ser un blanco fácil para los hackers.

Explotar vulnerabilidades en Las entidades bancarias y redes sociales son muy apetecido por los hackers, a lo largo de la historia hemos visto casos como:

JPMorgan (2014) los ciberdelincuentes lograron acceder a la red del banco y robar datos de nombres, dirección, teléfono, emails a 83 Millones de clientes que se usaron en suplantación de identidad, estafas o fraudes.

Twitter (2020) hackearon esta red saltándose la doble autenticación a varias cuentas de empresarios como: Bill Gates, Elon Musk, Barack Obama entre otros, publicaron desde sus cuentas que devolvería el doble de bitcoins aquellos que enviaran la criptomoneda a una dirección. Varias personas cayeron en esta falsa publicación.

Ashely Madisson: (2015) hackearon esta página web de citas extramatrimoniales exigiendo el cierre de la página, expandieron los datos de más de 37 Millones de usuarios en todo el mundo. Se encontró múltiples vulnerabilidades en la página, una de ellas permitía el uso de contraseñas débiles, muchos usuarios usaban 123456 o la palabra password como clave.

Tipos de ataques:

Pasivos: Los ataques pasivos se caracterizan por escuchar o interceptar el tráfico de red sin modificar o alterar los datos con el fin de obtener información, son difíciles de identificar, algunas de las técnicas más usadas son Sniffing y análisis de datos, Bocanegra (2014) señalaron que "No necesariamente se necesita de conocimientos profundos en informática, solo se necesita que la persona sea curiosa y el primer paso para cometer un ataque pasivo es observar y recopilar información sobre el entorno donde se encuentra la red inalámbrica a la cual se quiere obtener acceso" (p. 3).

Activos: Estos ataques modifican o eliminan datos, deniegan servicios, por esta razón se pueden identificar y aplicar medidas para prevenirlos. De acuerdo con Bocanegra (2014) "·si el atacante ya ha recolectado suficiente información mediante ataques pasivos, puede utilizarla para realizar ataques activos en los que se altera la información que circula en la red, afectando la integridad y disponibilidad de la misma" estos ataques pueden suplantar, re actuar, modificar y denegar servicios.

Tipos de vulnerabilidades:

Físicas: Son debilidades o fallas en la seguridad física de un sistema informático, dispositivo o instalación. Estas vulnerabilidades pueden incluir puertas de acceso, sistemas de alarmas deficientes o mal configurados, sistemas de vigilancia débiles. Romero (2018) describe un ejemplo de vulnerabilidad física "en muchas ocasiones se tiene los accesos a la infraestructura crítica y no se tiene los accesos pertinentes, cualquier persona podría abrir la puerta, podría entrar constituye un gran riesgo para la organización porque cualquier usuario podría ingresar con una USB y copiar la información, podría infectar la infraestructura" (p. 46)

Lógicas: Se refieren a debilidades en los sistemas operativos, el software, las aplicaciones o las configuraciones de red que pueden ser vulneradas por atacantes para acceder sin autorización a sistemas o información. Algunas de ellas son: contraseñas débiles, falta de parches de seguridad, sistemas mal configurados, errores de programación, puertos abiertos innecesarios o protocolos de seguridad obsoletos. Según Romer (2018) “Las vulnerabilidades de desarrollo, aquí se puede mencionar las inyecciones de código en SQL, Cross Site Scripting, esto puede variar dependiendo del tipo de aplicación, la validación de los datos” (p. 47)

Tipos de escáner de vulnerabilidades:

Basados en los host: Identifica problemas en los sistemas operativos de los host, software, navegadores, entre otros.

Basados en la red: Su objetivo es detectar puertos abiertos, servicios desconocidos que se ejecutan sobre ellos, configuración de firewall, paquetes que se envían, etc...

Basado en bases de datos: su objetivo es detectar posibles vulnerabilidades de SQL Injection

Métodos de escaneo de vulnerabilidades:

Caja blanca: Con esta prueba se tiene acceso a todos los sistemas de información, aplicaciones y/o arquitectura con privilegios de usuario administrador, el análisis tipo caja blanca “utilizara cierto usuarios con ciertos privilegios dentro de la red y accediendo a los servicios, dentro de los productos dentro de los softwares que se quieren auditar y así poder verificar si se puede realizar alguna

acción adicional en base los privilegios que se han brindado” de acuerdo a Romero (2018 p. 47)

Caja negra: Se cuenta con muy poca información sobre los sistemas a escanear, como por ejemplo, saber el nombre y página de web de la organización, en base a esta información, se procede a investigar en busca de la mayor cantidad de información que se pueda encontrar, de acuerdo a Romero (2018) con solo “ una sola dirección IP, algún nombre de alguna empresa, etc., a partir de aquí empieza como tal a buscar información, todo lo posible relacionado para la exploración y así poder obtener la mayor cantidad de información posible de dicha dirección ip, el resto de los equipos probablemente que se encuentran dentro de algún rango de direcciones ip asociado, aquí no se realiza ninguna instrucción, solo se detecta y se documenta la vulnerabilidad”

6. Método

Sampieri (2014) proponen un modelo general de investigación de enfoque cualitativo, que consta de seis pasos: planteamiento del problema, marco teórico, diseño de la investigación, recolección de datos, análisis de datos, conclusiones y recomendaciones. En su libro "Metodología de la investigación", los autores explican que el planteamiento del problema implica definir la cuestión de investigación y formular preguntas relevantes. El marco teórico, por su parte, consiste en revisar la literatura sobre el tema para identificar conceptos y teorías importantes. En cuanto al diseño de la investigación, los autores destacan la importancia de seleccionar una metodología y técnicas de recolección de datos apropiadas para responder a las preguntas de investigación. El siguiente paso es la recolección de datos, la cual se realizará con la herramienta de escaneo de vulnerabilidades. Una vez obtenidos los datos, se procede al análisis de los mismos para identificar analizar y comparar los hallazgos encontrados. Finalmente, se presentan las conclusiones de la investigación y se formulan recomendaciones para la remediación de las vulnerabilidades.

Según la norma ISO 27001:2005 proporciona un marco de trabajo con un enfoque cualitativo, para la gestión de la seguridad de la información. En general, el proceso de escaneo de vulnerabilidades debe ser parte del proceso de gestión de riesgos de seguridad de la información y se deben seguir las mejores prácticas para garantizar que se realice de manera efectiva y eficiente, para ello se debe seguir las siguientes fases:

- Identificación de activos críticos: Se debe tener un inventario de activos e identificar los activos críticos de información a los cuales se le va a realizar el escaneo y los riesgos asociados a este.

Los activos de información escogidos para realizar el escaneo de vulnerabilidades son: un servidor con sistema operativo Linux versión 3.10.0 El servidor tiene instalado un servicio de PHP (V 7.4.33) en apache y base de datos Mysql (V 15.1) y La aplicación PQR que está alojada en la página web de la compañía.

Este escaneo no representa ningún riesgo para el servidor o la aplicación, ya que las herramientas se usarán sólo con el fin de descubrir las vulnerabilidades más no atacarlas, se realizará un ataque pasivo, a pesar de esto, se verifica que cuenten con backup del código y base de datos al día, en caso de presentarse modificación en código o base de datos durante el escaneo se pueda recuperar la aplicación.

- Selección de herramientas de escaneo de vulnerabilidades: Hay gran variedad de herramientas de escaneo de vulnerabilidades disponibles en el mercado, algunas gratuitas, Es importante seleccionar herramientas que sean efectivas para los sistemas a escanear. Las herramientas escogida para este proyecto son:

Nmap: Lyon (2021) describe el uso de Nmap como una herramienta de escaneo de puertos de código abierto ampliamente utilizada en el ámbito de la seguridad informática. La herramienta se utiliza para descubrir y mapear redes, identificar sistemas y servicios, y evaluar la seguridad de los sistemas informáticos. Nmap se usará para escanear el servidor Linux

Owasp zap: OWASP (2021) es una herramienta de prueba de penetración de código abierto ampliamente utilizada en el ámbito de la seguridad informática. Tiene una interfaz gráfica de usuario y un conjunto de características para

identificar vulnerabilidades en aplicaciones web. Owasp zap se usará para escanear la aplicación PQR alojada en el servidor

- Escaneos de vulnerabilidades: se debe realizar escaneos de vulnerabilidades periódicamente para identificar nuevas vulnerabilidades o cambios en la red o sistemas. Los escaneos pueden ser realizados tanto de manera interna como externa (Caja negra, gris o blanca).

- Análisis y corrección de vulnerabilidades: Después de realizar el escaneo de vulnerabilidades, se deben analizar los resultados y tomar medidas para corregir las vulnerabilidades identificadas. Esto puede hacerse mediante la instalación de parches de seguridad (gestión de parches), ajuste en configuraciones, el aislamiento de los sistemas informáticos vulnerables o, en última instancia, mediante el cierre del sistema.

- Monitoreo continuo: Una vez que se hayan realizado los escaneos de vulnerabilidades y se hayan tomado medidas para corregir las vulnerabilidades identificadas, es importante realizar un retest, para validar que las vulnerabilidades se corrigieron efectivamente, se recomienda

- Monitorear continuamente los sistemas para detectar posibles nuevas vulnerabilidades y tomar medidas para corregirlas.

- Documentación: Es importante documentar todos los procesos relacionados con el escaneo de vulnerabilidades y la gestión de riesgos de seguridad de la información. Esto puede incluir políticas y procedimientos, informes de escaneo de vulnerabilidades, análisis de vulnerabilidades y

documentación de las medidas tomadas para corregir las vulnerabilidades identificadas.

En este proyecto se llegará a la fase de análisis y se describirán las posibles soluciones a las vulnerabilidades encontradas, posteriormente se realizará una correlación de los resultados con el proceso de gestión de vulnerabilidades internas, con el objetivo de encontrar similitudes o diferencias entre los resultados.

7. Cronograma y presupuesto

7.1 Cronograma

En la figura 1 se plasma el cronograma de actividades desde la semana dos de Agosto 2022 desde la planeación del proceso, hasta la semana cuatro de octubre donde se documentan los resultados finales.

Figura 1 - Cronograma de actividades

Vulnerabilidades Serviefectivo								
PROCESO	DIC	ENE	FEB	MAR	ABR	MAY	JUN	JUL
Planeación proceso								
Definición conceptos								
Instalación herramientas de escaneo								
Escaneo de vulnerabilidades del sistema								
Analisis de resultados								
Remediación de vulnerabilidades								
Re-escaneo validación de remediación								
Documentos resultados								
	2022	2023	2023	2023	2023	2023	2023	2023

Nota: Fuente propia.

7.2 Presupuesto

OWASP ZAP es un software libre de seguridad informática utilizado para identificar vulnerabilidades en aplicaciones web. Según OWASP (2022), "ZAP es una herramienta fácil de usar que puede ayudar a cualquier persona interesada en mejorar la seguridad de sus aplicaciones web" (párrafo 1). Como software libre, está disponible para su descarga y uso sin costo alguno.

Horas Ingeniero:

Valor hora ingeniero:	100.000 x hora
Horas escaneo:	8 horas
Horas análisis resultados:	8 horas
Horas realización informe:	4 horas
Presentación informe:	2 horas
Retest	6 Horas
Total horas:	28 Horas
Valor total en pesos	$2.800.000 pesos mcte
Nota: Son valores aproximados	

Remediación de vulnerabilidades:

Valor hora ingeniero	70.000 x hora
Cronograma de vulnerabilidades	4 horas
Remediación	200 Horas
Total horas	204 Horas
Valor total en pesos	$ 14.280.000 pesos mcte
Nota: Son valores aproximados	

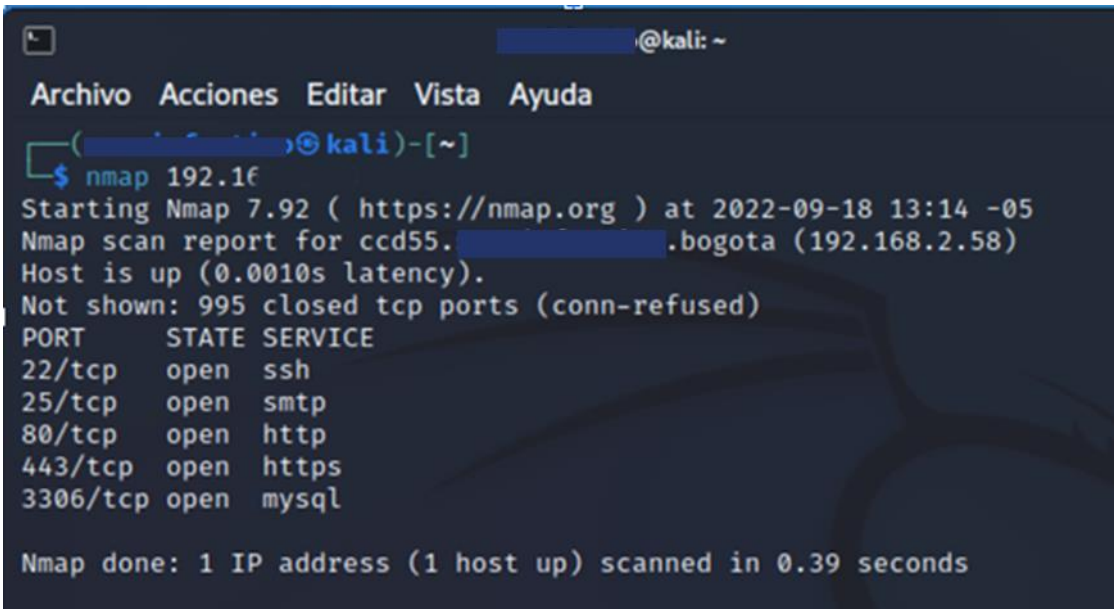
Nota: Fuente propia.

8. Resultados o hallazgos

8.1 Análisis con Herramienta Nmap

Se analizó el servidor web con la herramienta nmap con un equipo con sistema operativo kali Linux, con el objetivo de verificar qué puertos se encuentran abiertos.

Figura 2 - Análisis de la ip privada del servidor web con nmap



Fuente: propia

Como se ve en la Figura 2 se encuentran los siguientes puertos abiertos:

- **puerto tcp 22, servicio SSH**: es el protocolo para facilitar comunicación encriptada entre cliente servidor, permite a un usuario conexión remota al servidor.

Figura 3 - Funcionamiento protocolo SSH

Nota: Imagen tomada de: Qué es el protocolo SSH y cómo configurarlo para mejorar la seguridad de acceso a los servidores Linux (p. 4) por Cardenal Gardok Hostalia https://pressroom.hostalia.com/contents/ui/theme/images/WP-Hostalia-protocolo-SSH.pdf

De acuerdo con Gardok (sf) “el uso del protocolo SSH es totalmente seguro, esto no quiere decir que esté ajeno a sufrir algún tipo de ataque que ponga en riesgo nuestra información. Por este motivo, los usuarios tienen la opción de modificar la configuración por defecto que trae este protocolo para hacerlo aún más seguro, como puede ser el cambio del puerto por defecto o el número máximo de reintentos para conectarse al servidor” (p 6), Para fortalecer la seguridad en este servicio, es necesario cambiar el puerto 22 que tiene configurado por defecto, en Linux se deben seguir los siguientes pasos:

- ✓ Localizar el fichero “sshd_config”, usualmente está en “/etc/ssh”
- ✓ ingresar al fichero “sshd_config” con un programa de edición, puede ser nano, vi o vim
- ✓ editar el puerto 22, por cualquier valor , por ejemplo 1987
- ✓ para fortalecer la seguridad aún más este servicio, se puede deshabilitar el acceso root, editamos la opción “PermitRootLogin” se debe poner

un “no”; El usuario root es quien tiene más privilegios sobre un servidor, una buena práctica de seguridad es impedir el logueo con este usuario, para iniciar sesión se realizará con otro usuario creado con menos privilegios y posterior al ingreso y de ser necesario ingresar con root, se puede hacer con el comando “sudo”.

- **Puerto tcp 25, servicio SMTP**: Protocolo que utiliza el servidor para enviar y recibir correos electrónicos a través de internet.

MITRE Corporation. (2021) De acuerdo al CVE en puerto 25 publicaron una vulnerabilidad que codificaron CVE-2021-43270, donde se puede usar el servicio SMTP en texto claro sin cifrar por el puerto 25, se catalogó con gravedad media ya que el impacto afecta parcialmente solo a la confidencialidad del sistema. Un atacante puede explotar esta vulnerabilidad para enviar spam.

La solución para remediar esta vulnerabilidad es cerrar el puerto 25 y hacer uso del puerto 465 para el envío de email, ya que por este puerto si va cifrada la comunicación.

- **Puerto tcp 80, servicio HTTP**: protocolo que sirve para la navegación de páginas web no seguras, es decir, páginas que no tiene instalado un certificado SSL para cifrar los datos.

NCIBE-CERT. (2020). En junio de 2020 publicaron una vulnerabilidad de denegación de servicios (dos) relacionada al puerto 80, el atacante puede enviar spam de cabeceras HTTP incompletas bloqueando el acceso al dashboard, esta vulnerabilidad se codificó con CVE-2020-10280 con severidad alta.

Para mitigar esta vulnerabilidad se recomienda aplicar los parches de seguridad y actualizaciones que publica el fabricante cuando estén disponibles y de la página oficial del proveedor.

- **Puerto tcp 443, servicio HTTPS**: protocolo para la navegación de páginas web seguras.

INCIBE-CERT. (2019). En abril de 2019 publicó una vulnerabilidad donde un atacante con accesos de red, sin necesidad de autenticarse, al servidor web, podría ejecutar comandos con privilegios de administrador, su código es CVE-2019-6579 catalogado con crítica.

Para mitigar se recomienda proteger el acceso a la red por medio de reglas de firewall, segmentación de la red y/o cortafuegos, adicional es buena práctica mantener el sistemas operativo actualizado con sus parches de seguridad

- **Puerto tcp 3306, servicio MYSQL**: puerto por defecto para el servicio de base de datos de mysql.

Para asegurar este servicio de los atacantes, es necesario limitar las conexiones solo para las ips requeridas.

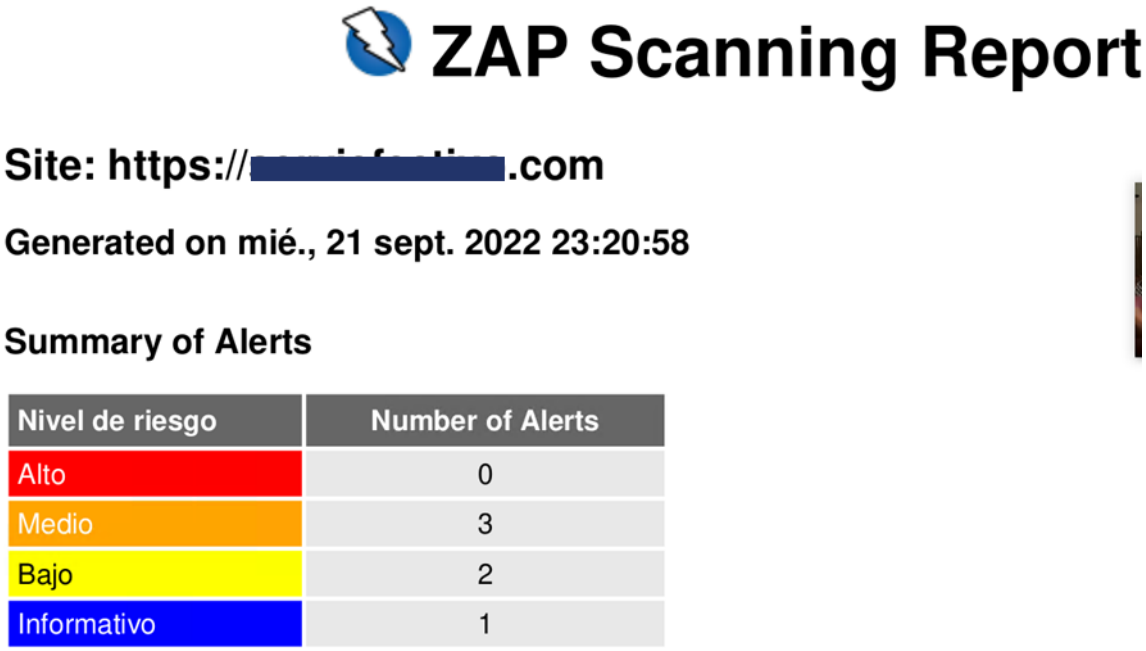
INCIBE-CERT. (2011). publicaron una vulnerabilidad en Mysql versión 5.5.8 que se esté ejecutando en sistemas operativos Windows, el atacante realiza denegación de servicios modificando los paquetes que pasan a través del puerto 3306. El código de la vulnerabilidad es CVE-2011-5049, está catalogada con riesgo medio.

En este caso el servicio de Mysql está funcionando sobre un sistema operativo Linux y su versión es la 15.1, por lo que no aplica la vulnerabilidad. En caso de aplicar la remediación sería actualizar la versión del Mysql.

8.2 Análisis con Owasp Zap

Con la herramienta owasp-zap se analizó una url que están publicadas en la página web de la empresa la compañía: pqr

Figura 4 - Resultado escaneo de urls



ZAP Scanning Report

Site: https://[redacted].com

Generated on mié., 21 sept. 2022 23:20:58

Summary of Alerts

Nivel de riesgo	Number of Alerts
Alto	0
Medio	3
Bajo	2
Informativo	1

Nota: fuente propia

En la figura 4 muestra el resumen de la cantidad de vulnerabilidades encontradas agrupadas por nivel de riesgo.

Vulnerabilidades nivel de riesgo medio

Figura 5 - Alertas de nivel de riesgo medio.

Nombre	Nivel de riesgo	Number of Instances
Application Error Disclosure	Medio	1
Ausencia de fichas (tokens) Anti-CSRF	Medio	1
Content Security Policy (CSP) Header Not Set	Medio	3

Nota: fuente propia

Las vulnerabilidades de riesgo medio encontradas son:

Application Error Disclosure

Esta vulnerabilidad hace referencia a las Aplicaciones con mensajes de error o advertencia que pueden divulgar información confidencial, como la ubicación de un archivo. De acuerdo a Ferrer M. (2015) “los mensajes de error enviados por el servidor, que suelen ser de gran utilidad durante el proceso de desarrollo de la aplicación, se vuelve contra nosotros cuando siguen apareciendo en una aplicación que se encuentra en la etapa de producción, por lo que es necesario deshabilitar todos estos mensajes y editar algunos otros (como los que se envían cuando el servidor no encuentra algún archivo en particular) que también puede ser utilizado por los atacantes para obtener información de nuestro sistema” (p. 125)

Hay 3 opciones para solucionar este error en aplicaciones desarrolladas en PHP (V 7.4.33):

✓ Modificar las variables display_errors=’off’ y logs_errors=’on’, en el archivo php.ini, esta solución quita los mensajes de error de todas las aplicaciones.

✓ La siguiente solución se debe realizar a cada aplicación donde se requiera, se debe modificar o crear en el archivo .htaccess dentro de cada aplicación incluyendo los parámetros:

```
php_flag display_errors off
php_flag log_errors on
```

✓ Con la siguiente opción solo se ajusta los mensajes de errores en el archivo php, incluyendo estos parámetros:

```
<?php
error_reporting(0);
?>
```

Ausencia de fichas (tokens) Anti-CSRF

La presencia o ausencia de tokens Anti-CSRF en una aplicación web puede afectar la seguridad de la aplicación, Los tokens Anti-CSRF son una medida de seguridad que ayuda a prevenir ataques de falsificación de solicitudes entre sitios (CSRF), que pueden comprometer la integridad de los datos y la privacidad de los usuarios. Kratzke, N. (2019).

Con esta vulnerabilidad un atacante puede aprovechar los campos de un formulario para falsificar una petición haciéndose pasar por un usuario determinado.

Para solucionar esta vulnerabilidad en aplicaciones php (V 7.4.33) se deben crear tokens aleatorios cada vez que se envié información en formularios, la solución de código sugerida es la siguiente:

1. En el formulario donde se solicitan los datos, se debe crear en una variable de sesión un token aleatorio

```
$_SESSION['token'] = md5(uniqid(mt_rand(), true));
```

2. Crear un input tipo oculto que se llame token y como valor le otorgamos la variable de $_SESSION['token']

```
<input type="hidden" name="token" value="<?php echo $_SESSION['token'] ?? '' ?>">
```

3. En el formulario que recibe los datos, verificar si el valor del input existe y compararlo con la variable $_SESSION['token'], si es efectiva debe continuar con el desarrollo del proceso, caso contrario, si la comparación no da un resultado positivo, se debe retornar a la página de inicio o informar del error.

```
$token = filter_input(INPUT_POST, 'token', FILTER_SANITIZE_STRING);

if (!$token || $token !== $_SESSION['token']) {
    // return 405 http status code
    header($_SERVER['SERVER_PROTOCOL'] . ' 405 Method Not Allowed');
    exit;
} else {
    // process the form
```

}

Content Security Policy (CSP) Header Not Set

Esta vulnerabilidad informa que no hay configurado Políticas de Seguridad del Contenido dentro del código de la aplicación. Configurar estas políticas de contenido brinda una capa de seguridad que ayuda a mitigar o prevenir algunos tipos de ataques (Cross site Scripting, SQL Injection entre otros). De acuerdo con Mozilla (2022) en su guía de seguridad en HTTP informa que “El principal objetivo del CSP es mitigar y reportar ataques XSS. Los ataques XSS se aprovechan de la confianza del navegador en el contenido que recibe del servidor. El navegador de la víctima ejecutará los scripts maliciosos porque confía en la fuente del contenido, aun cuando dicho contenido no provenga de donde se supone.”

Dentro del CSP se puede configurar las directivas individualmente asignándoles un recurso específico o dejarlas configuradas con default-src ‘self’ para que el agente de usuario le asigne el mismo recursos a todas las directivas

Listado de directivas que maneja el CSP:

- ✓ child-src
- ✓ connect-src
- ✓ font-src
- ✓ frame-src
- ✓ img-src
- ✓ manifest-src
- ✓ media-src

- ✓ object-src
- ✓ prefetch-src
- ✓ script-src
- ✓ script-src-elem
- ✓ script-src-attr
- ✓ style-src
- ✓ style-src-elem
- ✓ style-src-attr
- ✓ worker-src

Hay dos opciones para solucionar esta vulnerabilidad:

- una opción para remediar esta vulnerabilidad es Incluir la siguiente línea de código en el encabezado de los archivos de php:

Con default-src ‘self’

```
<?php
    header("Content-Security-Policy: default-src 'self'");
?>
```

O asignándole un recurso específico a una directiva

```
<?php
    header("Content-Security-Policy: default-src 'self'; script-src https://www.ejemplo.com");
?>
```

- la segunda opción es agregar en el archivo .htacess la siguiente línea de código de cabecera

Con default-src ‘self’

Header add Content-Security-Policy "default-src 'self'"

O asignándole un recurso específico a una directiva

Header add Content-Security-Policy "default-src 'self'; script-src https://www.ejemplo.com"

Vulnerabilidades nivel de riesgo bajo

Figura 6 - Alertas de nivel de riesgo bajo.

Cross-Domain JavaScript Source File Inclusion	Bajo	4
El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ""X-Powered-By""	Bajo	1

Nota: fuente propia

Cross-Domain JavaScript Source File Inclusion

De acuerdo a la documentación del CWE de la vulnerabilidad, está identificada con el código CWE-829, esta vulnerabilidad se refiere que encontró archivos javascript en un domino tercero, un atacante podría aprovechar esta vulnerabilidad ejecutando código malicioso,

Para sanear esta vulnerabilidad es necesario cambiar la ruta del archivo por la del dominio propio, previamente descargado el archivo a referenciar. Con base a la solución planteada por las mejores prácticas de OWASP (2010) donde afirmar que la se “Asegúrese de que los archivos fuente de JavaScript se carguen solo desde fuentes confiables y que los usuarios finales de la aplicación no puedan controlar las fuentes.”

Se debe cambiar la ruta de googleapis.com

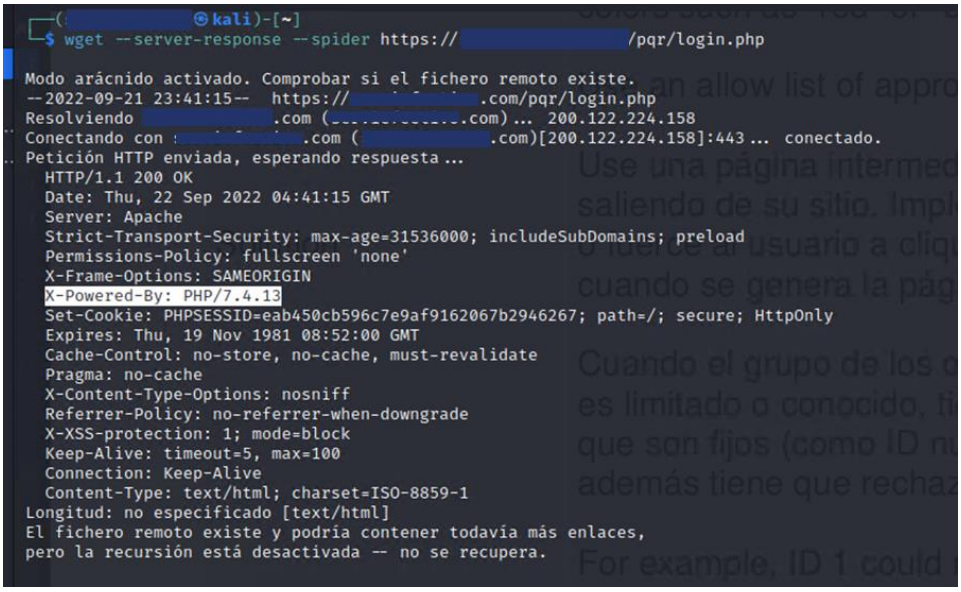
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"> </script>

Por la de la compañia.com

<script src="https://compañia.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

El servidor divulga información mediante un campo(s) de encabezado de respuesta HTTP ""X-Powered-By""

Figura 7 - Análisis de página web con wget --server-response –spider

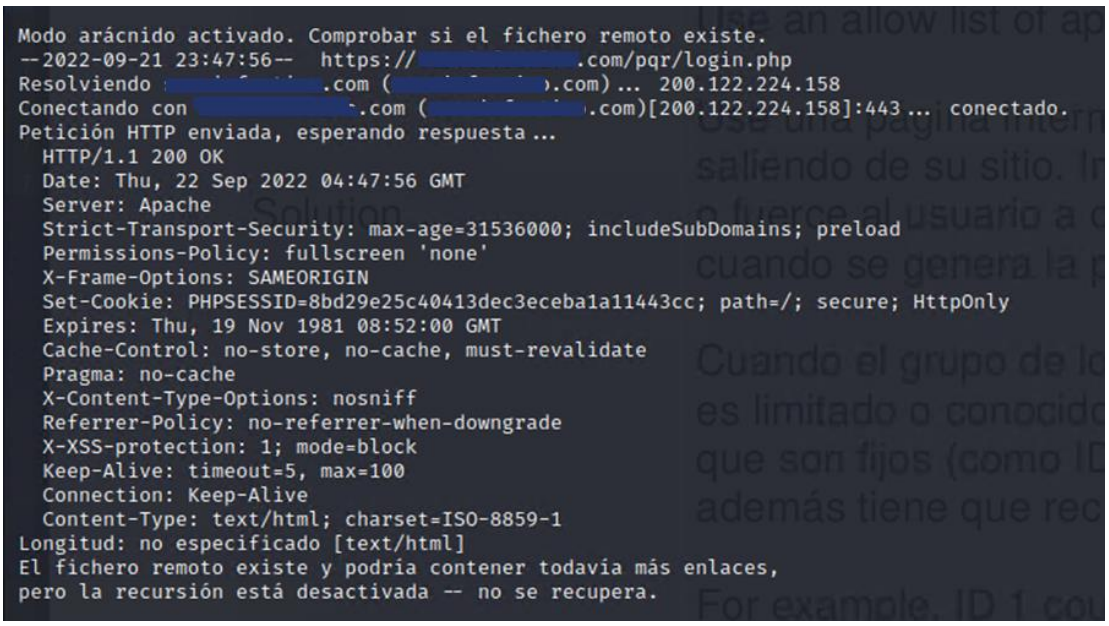


Nota: Cuando se invoca con esta opción, Wget se comportará como una araña web, lo que significa que no descargará las páginas, solo verificará que estén allí. Fuente propia

Como se ve en la figura 6 se realizó una verificación de la información que expone el servidor y notamos que brinda información sensible en cuanto a versiones del lenguaje de programación que puede ser explotado por un atacante.

Para sanear esta vulnerabilidad, se debe modificar el archivo de configuración de php: php.ini y modificar la variable expose_php = Off, posterior a esto aplicar cambios reiniciando el servicio de php.

Figura 8 - Análisis de página de la compañia después de la remediación



Vulnerabilidades nivel de riesgo informativo

Figura 9 - Alertas de nivel de riesgo bajo.

Divulgación de información - Comentarios sospechosos	Informativo	2

Nota: fuente propia

Divulgación de información - Comentarios sospechosos

Esta vulnerabilidad informa que se están dejando comentarios con contenido que puede revelar detalles que pueden ayudar y dar pistas a un atacante.

Esta vulnerabilidad se evidencio en este programa, porque hay comentarios que tienen la palabra Select, y el programa lo toma como parte de una sentencia SQL que se está exponiendo. La remediación es crear buenas prácticas para comentar el código, dejar de usar expresiones de consultas como: select, update, delete o información de ip privadas, de acuerdo con OWASP Testing Guide v3.0 (2008) “los comentarios incluidos en línea en código HTML podrían revelar información interna, que no debería estar disponible, a un atacante potencial. A veces, incluso el código fuente queda convertido en comentario porque ya no es necesario, pero este comentario es filtrado de forma no intencionada dentro de las páginas HTML retornadas a los usuarios.” (p 103)

8.3 Cuestionario sobre vulnerabilidades de la compañía

Se realiza cuestionario acerca del proceso de escaneo y saneo de vulnerabilidades en la empresa, se entrevista al oficial de seguridad de la compañía.

- ¿Tienen documentado un proceso de vulnerabilidades de acuerdo a los numerales 4.2.1 Establecimiento del SGSI, punto d-3 identificar las vulnerabilidades que podrían ser aprovechadas por las amenazas Y A.12.6 Gestión de la vulnerabilidad técnica de la norma ISO27001:2013?

Rta: Si, dentro del manual de políticas de seguridad de la información, se encuentra el proceso para de Análisis de vulnerabilidades, la empresa tiene contratada un empresa externa, donde cada año realiza un test de vulnerabilidades de caja negra y caja blanca, para el análisis de caja blanca, se le habilita una VPN al proveedor para que pueda acceder a la red de la compañía, posterior al escaneo emiten un informe técnico y otros informes detallados de las vulnerabilidades encontradas, clasificadas por su criticidad de acuerdo al puntaje del CVSS; El comité de seguridad de la información se reúne para analizar los resultados y empezar a remediarlas, se da prioridad a las vulnerabilidades con clasificación crítica y alta.

Cuando se termina la remediación se le informa al proveedor, para que realice un retest y así confirmar si las vulnerabilidades fueron o no remediadas.

- ¿Tienen conocimiento de la herramienta que usa el proveedor para realizar el escaneo de vulnerabilidades?

Rta: El proveedor nos ha informado que usa varias herramientas de escaneo, algunas de ellas son: nessus y acunetix.

- ¿Internamente en la organización, realizan algún escaneo de vulnerabilidades?

Rta: Si, para las aplicaciones web se usa Vega es una herramienta libre y para la red se usa las herramientas de kali Linux

- ¿Tienen documentado un proceso de desarrollo seguro de acuerdo al numeral A.12 adquisición, desarrollo y mantenimiento de sistemas de información?

Rta: Si, tenemos una política de desarrollo seguro, usamos la metodología CRUD Create, read, update y delete, para los desarrollos que se realizan, adicional en el manual de políticas se describe buenas prácticas a tener en cuenta en el momento de crear una aplicación como lo son: tener los ambientes de desarrollo, pruebas y producción separados, la aplicación debe pedir cambio de clave a los usuarios cada 30 días, se debe asegurar cerrar las conexiones a bases de datos que se abran durante la ejecución de la aplicación, manejo de roles y perfiles, entre otros.

- ¿Por qué a pesar de tener una política de desarrollo seguro, se presentan vulnerabilidades sobre las aplicaciones?

Rta: Las vulnerabilidades encontradas son sobre aplicaciones antiguas, anteriormente solo se tenía un persona que se encargaba de los desarrollos y tenía demás funciones, por lo que no se tenía el tiempo para realizar mantenimiento y actualización sobre estas. Hace 1 año se contrató una persona para apoyar estas funciones.

- Nosotros realizamos un escaneo de vulnerabilidades a una aplicación de la compañía, y en los resultados arrojaron 3 vulnerabilidades de criticidad media, ¿cómo se sanearon esas vulnerabilidades?

Rta: La vulnerabilidad Aplication Error Disclosore, se soluciona ajustando las configuraciones del servidor para que no muestre los mensajes de error en las aplicaciones

De las vulnerabilidades de Anti-CSRF y CSP: Estas vulnerabilidades no fueron reportadas por el proveedor por lo que no se ha solucionado.

- ¿Sobre la aplicación PQR el proveedor reportó vulnerabilidades?

Rta: Si, sobre la aplicación se encontraron dos vulnerabilidades, una clasificada como alta: SQL injection y otra baja acerca de la dependencia de Javascript.

- ¿Ya se solucionaron estas vulnerabilidades reportadas por el proveedor?

Rta: de la aplicación específica, ya se solucionaron las vulnerabilidades, internamente usamos el escáner Vega que es gratuito para confirmar que estas fueron saneadas efectivamente

8.4 Correlación de Análisis de los resultados

Se identificó diferencias en los resultados de las herramientas de escaneo de vulnerabilidades que se realizó sobre la aplicación web, en el desarrollo de este proyecto usamos Owasp Zap, en la organización internamente usan la herramienta Vega y el proveedor contratado por la organización para realizar escaneo con un conjunto de herramientas, entre ellas nexus y acunetix, por lo que la herramienta de Owasp Zap arrojaron diferentes vulnerabilidades a las reportadas por el proveedor, esto se debe a que OWASP ZAP, Vega y Nexus son herramientas de escaneo de vulnerabilidades web que utilizan diferentes técnicas y algoritmos para identificar y reportar vulnerabilidades en una aplicación web.

Cada herramienta tiene sus propios métodos de escaneo y su propio conjunto de reglas y patrones para detectar vulnerabilidades.

Además, cada herramienta se actualiza de forma independiente y puede tener diferentes versiones o bases de datos de vulnerabilidades. Por lo tanto, es común que estas herramientas muestran diferentes resultados en un escaneo de vulnerabilidades de una misma aplicación web.

Es importante destacar que las herramientas de escaneo de vulnerabilidades no son infalibles y pueden tener limitaciones en la detección de ciertos tipos de vulnerabilidades o en la identificación de falsos positivos o negativos. Por lo tanto, se recomienda utilizar varias herramientas y complementar el escaneo con pruebas manuales para obtener una evaluación más completa de la seguridad de una aplicación web.

9. Conclusiones

Se cumplió con los objetivos planteados en la investigación ya que se logran detectar nuevas vulnerabilidades sobre la aplicación y se dan recomendaciones para ser subsanadas o remediadas.

Según el análisis realizado con la herramienta Nmap y la herramienta Owasp-Zap, se identificaron varias vulnerabilidades en el servidor web y en la aplicación web de la compañía. Estas vulnerabilidades representan riesgos para la seguridad de la infraestructura y la aplicación, lo que podría permitir a los atacantes acceder, manipular o divulgar información confidencial, así como ejecutar código malicioso

El análisis realizado reveló varias vulnerabilidades que requieren atención y remedios para fortalecer la seguridad de la infraestructura y la aplicación web. Se recomienda implementar las soluciones propuestas y seguir las mejores prácticas de seguridad para mitigar los riesgos identificados. Además, es fundamental mantener los sistemas actualizados y aplicar parches de seguridad de manera regular para evitar futuras vulnerabilidades conocidas.

Actualmente se cuenta con variedad de herramientas que permiten realizar un escaneo de vulnerabilidades a los sistemas de información, por ello, las empresas deben concientizarse sobre la importancia de tener personal calificado y ético para realizar estas actividades periódicas con el fin de evitar fuga o pérdidas de información.

Documentar buenas prácticas de desarrollo para la creación y modificación de aplicaciones web, en este caso en particular, se evidenció que es una aplicación que se desarrolló hace más de 10 años, por lo que es importante al menos una vez al año realizar escaneo a todas las aplicaciones con el fin de

mantener, tanto el código, como la información de la base de datos, segura, íntegra y disponible. Posterior a la remediación se debe volver a escanear para confirmar si las vulnerabilidades fueron remediadas correctamente.

Es importante tener presente que se descubren nuevas vulnerabilidades, por lo que se es necesario estar suscrito en algún boletín de alertas, por ejemplo al de incibe-cert y estar consultado la página de CVE: https://cve.mitre.org con el fin de mantenerse actualizado y estar alerta ante cualquier eventualidad que ponga en riesgo los sistemas de información.

Los fabricantes de sistemas operativos publican parches de seguridad cuando descubren una amenaza, por lo que se debe mantener los sistemas operativos parchados o actualizados a la última versión estable disponible.

La alta gerencia debe conocer y apoyar los procesos de seguridad de la información, de acuerdo a Carpentier (2016) “El impacto de las diferentes amenazas varía considerablemente según el efecto sobre la empresa, algunas tienen un impacto sobre la confidencialidad o la integridad de datos, otras actúan sobre la disponibilidad de los sistemas” (P. 43). Si un riesgo se materializa conlleva a gastos económicos y/o inversión de tiempo, al afectar la triada de la seguridad de la información puede llevar a una mala reputación de la compañía (mala imagen), en contraste, para una compañía es más solvente tener un sistema blindado a perder la información por algún tipo evento y césar actividades ocasionando una pérdida económica o reputacional que impactante significativamente a la compañía.

10. Referencias

Areito, J. (2008). Seguridad de la información. Madrid: Ediciones Paraninfo.

Bocanegra J. (2014). Hacking a redes inalámbricas, Recuperado de http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/2845/Trabajo%20de%20grado1609.pdf?sequence=1&isAllowed=y

Carpentier, J.-F. (2016). La Seguridad Informática En La Pyme: Situación Actual Y Mejores Prácticas. Barcelona: ENI.

Ferrer J. (2015). Implantación de aplicaciones web en entornos internet, intranet y extranet. Grupo Editorial RA-MA. Recuperado de: https://books.google.com.co/books?id=Go6fDwAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Gardoki, C. Qué es el protocolo SSH y cómo configurarlo para mejorar la seguridad de acceso a los servidores Linux. Hostalia whitepapers, 1-8. Recuperado de https://pressroom.hostalia.com/contents/ui/theme/images/WP-Hostalia-protocolo-SSH.pdf

Gutiérrez J. (2015). Instalación y configuración del software de servidor web. Elearning. Recuperado de: https://books.google.es/books?hl=es&lr=lang_es&id=UHpXDwAAQBAJ&oi=fnd&pg=PA9&dq=ubuntu+servidor+web&ots=bHXdc1MYl5&sig=EWJD7t8C2Zpy_3J19VoJiJyRptw#v=onepage&q=ubuntu%20servidor%20web&f=false

Hernández Sampieri, R., Fernández-Collado, C., & Baptista Lucio, P. (2014). Metodología de la investigación (6ª ed.). McGraw-Hill Education.

IBM (2021), Common Vulnerability Scoring System (CVSS). España, Recuperado de https://www.ibm.com/docs/es/qsip/7.3.2?topic=vulnerabilities-common-vulnerability-scoring-system-cvss

INCIBE-CERT. (2011). CVE-2011-5049. Recuperado de https://www.incibe-cert.es/alerta-temprana/vulnerabilidades/cve-2011-5049

ISO 27001 (2015), https://www.icontec.org/eval_conformidad/certificacion-iso-27001-sistemas-de-gestion-de-seguridad-de-la-informacion-2/

Kratzke, N. (2019). Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet. Recuperado de: https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

Lyon G. (2021). Nmap Security Scanner. Recuperado, de https://npcap.com/guide

Lerdorf, R. (2020). Best practices for securing PHP applications. [Blog post]. Recuperado de https://www.php.net/security-best-practices

Maucaylle Leandres, A. (2019). Construcción de un modelo de red virtual para aplicar técnicas de hacking ético y poder analizar los eventos relacionados a la seguridad informática sobre una infraestructura virtual. Recuperado de: https://repositorio.unajma.edu.pe/bitstream/handle/20.500.14168/489/Alex_Tesis_Bachiller_2019.pdf?sequence=1&isAllowed=y

MITRE Corporation. (2020). CVE List Home. Recuperado de https://cve.mitre.org

Mitre Corporation. (2021). Common Vulnerabilities and Exposures (CVE). Recuperado de https://cve.mitre.org/about/index.html

Moreno, A. C., Sánchez, D. F. A., & Sánchez, F. J. A.(2016) Identificación De Riesgos y Vulnerabilidades En Puertos De Servicios Informáticos.

Mozilla. (2022). Content Security Policy (CSP). Mozilla Developer Network. Recuperado de https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP

OWASP. (2015). Guía de pruebas de OWASP (versión 3.0) [PDF]. Recuperado de https://owasp.org/www-pdf-archive/Guía_de_pruebas_de_OWASP_ver_3.0.pdf

República De Colombia. Ley Estatutaria 1266 DE 2008. (2008).. Recuperado de: http://www.secretariasenado.gov.co/senado/basedoc/ley_1266_2008.html

Romero Castro, M. I., Figueroa Morán, G. L., Vera Navarrete, D. S., Álava Cruzatty, J. E., Parrales Anzúles, G. R., Álava Mero, C. J., Murillo Quimiz, A. L., & Castillo Merino, M. A. (2018). Introducción a la seguridad informática y el análisis de vulnerabilidades (Vol. 46 de Ingeniería y Tecnología). 3Ciencias. recuperado de: https://play.google.com/books/reader?id=5Z9yDwAAQBAJ&pg=GBS.PA46&hl=es

Silva F., Segadas L. y Kowask E. (2016), Gestión de seguridad de la información RENATA, recuperado de : https://www.cedia.edu.ec/assets/docs/publicaciones/libros/GTI8.pdf

Sklar, D. (2014). PHP Cookbook: Solutions & Examples for PHP Programmers (3rd ed.). Sebastopol, CA: O'Reilly Media.

Shuttleworth (2023), M. (s.f.). Recuperado de https://ubuntu.com/about

The OWASP Foundation (2021) OWASP Top Ten. Europa, Recuperado de https://owasp.org/www-project-top-ten/

Torvalds, L. (1991). Linux: A portable operating system. Recuperado de https://www.kernel.org

Urbina, G. B. (2016). Introducción a la seguridad informática. México: Grupo Editorial Patria.

W3Techs. (2023). Programming Language Statistics. Recuperado de https://w3techs.com/technologies/overview/programming_language/all

CORPORACIÓN UNIVERSITARIA
UNITEC

Por intermedio del presente documento en mi calidad de autor o titular de los derechos de propiedad intelectual de la obra que adjunto, titulada **Implementación de remediaciones de vulnerabilidades identificadas en aplicaciones desarrolladas en PHP (V 7.4.33) sobre un servidor con sistema operativo Linux (Ubuntu), mediante escaneos de seguridad (OWASP) realizados en una empresa pyme de Call Center**, autorizo a la Corporación universitaria Unitec para que utilice en todas sus formas, los derechos patrimoniales de reproducción, comunicación pública, transformación y distribución (alquiler, préstamo público e importación) que me corresponden como creador o titular de la obra objeto del presente documento.

La presente autorización se da sin restricción de tiempo, ni territorio y de manera gratuita. Entiendo que puedo solicitar a la Corporación universitaria Unitec retirar mi obra en cualquier momento tanto de los repositorios como del catálogo si así lo decido.

La presente autorización se otorga de manera no exclusiva, y la misma no implica transferencia de mis derechos patrimoniales en favor de la Corporación universitaria Unitec, por lo que podré utilizar y explotar la obra de la manera que mejor considere. La presente autorización no implica la cesión de los derechos morales y la Corporación universitaria Unitec los reconocerá y velará por el respeto a los mismos.

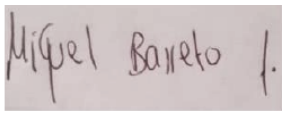
La presente autorización se hace extensiva no sólo a las facultades y derechos de uso sobre la obra en formato o soporte material, sino también para formato electrónico, y en general para cualquier formato conocido o por conocer. Manifiesto que la obra objeto de la presente autorización es original y la realicé sin violar o usurpar derechos de autor de terceros, por lo tanto, la obra es de mi exclusiva autoría o tengo la titularidad sobre la misma. En caso de presentarse cualquier reclamación o por acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión asumiré toda la responsabilidad, y saldré en defensa de los derechos aquí autorizados para todos los efectos la Corporación universitaria Unitec actúa como un tercero de buena fe. La sesión otorgada se ajusta a lo que establece la ley 23 de 1982.

Para constancia de lo expresado anteriormente firmo, como aparece a continuación.

Firma

Nombre Maria Rocio Camargo Villa
CC. 1.010.187.477

Firma



Nombre Miguel Alejandro Barreto F.
CC. 1.014.180.015