UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

Proyecto de investigación previo a la obtención del título de Ingeniero en Sistemas y Computación

TRABAJO DE TITULACIÓN

MANUAL DE IMPLEMENTACIÓN DE UN PROCESO HARDENING PARA MITIGAR VULNERABILIDADES EN EL SERVIDOR WEB NGINX DE LA UNACH.

Autora:

Karina Paola Pinduisaca Guashpa

Tutor:

Ing. Lorena Molina Valdiviezo., Ph.D.

RIOBAMBA – ECUADOR

2022

DERECHO DE AUTORÍA

Yo, Karina Paola Pinduisaca Guashpa, con cédula de ciudadanía 060511475-0, autor (a) (s) del trabajo de investigación titulado: **Manual de implementación de un proceso hardening para mitigar vulnerabilidades en el servidor web Nginx de la UNACH**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 29 - 11 - 2022.

KARINA PAOLA PINDUISACA GUASHPA

0605114750

DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL;

Quienes suscribimos, catedráticos designados Tutor y Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **Manual de implementación de un proceso hardening para mitigar vulnerabilidades en el servidor web Nginx de la UNACH**, con cédula de identidad número 0605114750, certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha asesorado durante el desarrollo, revisado y evaluado el trabajo de investigación escrito y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 29 - 11 - 2022

Ing. Ximena Quintana
PRESIDENTE DEL TRIBUNAL DE GRADO

Ing. Fernando Molina., Ph.D.
MIEMBRO DEL TRIBUNAL DE GRADO

Ing. Ana Congacha
MIEMBRO DEL TRIBUNAL DE GRADO

Ing. Lorena Molina Valdiviezo., Ph.D.
TUTOR

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **Manual de implementación de un proceso hardening para mitigar vulnerabilidades en el servidor web Nginx de la UNACH** por Karina Paola Pinduisaca Guashpa, con cédula de identidad número 0605114750, bajo la tutoría PhD. Lorena Molina Valdiviezo; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 29 - 11 - 2022

Ing. Ximena Quintana.
Presidente del Tribunal de Grado

Ing. Fernando Molina., Ph.D.
MIEMBRO DEL TRIBUNAL DE GRADO

Ing. Ana Congacha., Mag.
MIEMBRO DEL TRIBUNAL DE GRADO

CERTIFICADO ANTIPLAGIO

Dirección
Académica
VICERRECTORADO ACADÉMICO

en movimiento
SGC
SISTEMA DE GESTIÓN DE LA CALIDAD
UNACH-RGF-01-04-02.20
VERSIÓN 02: 06-09-2021

CERTIFICACIÓN

Que, **PINDUISACA GUASHPA KARINA PAOLA** con CC: **0605114750**, estudiante de la Carrera **INGENIERÍA EN SISTEMAS Y COMPUTACIÓN**, **NO VIGENTE**, Facultad de **INGENIERÍA**; ha trabajado bajo mi tutoría el trabajo de investigación **titulado "MANUAL DE IMPLEMENTACIÓN DE UN PROCESO HARDENING PARA MITIGAR VULNERABILIDADES EN EL SERVIDOR WEB NGINX DE LA UNACH"** cumple con el 8 %, de acuerdo al reporte del sistema Anti plagio **URKUND**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 22 de noviembre de 2022

LORENA PAULINA MOLINA VALDIVIEZO

PhD. Lorena Molina
TUTOR(A) TRABAJO DE INVESTIGACIÓN

DEDICATORIA

El presente proyecto de investigación dedico a Dios quien ha sido mi guía a lo largo de mi carrera universitaria y de mi vida.

A mis padres por ser el principal cimiento para la construcción de mi formación profesional, por inculcar el ejemplo de esfuerzo, valentía y palabras de aliento hicieron de mí una mejor persona. A mis hermanas por estar conmigo en cada momento, a mis adoradas sobrinas por llenarme de alegría día tras día, A toda mi familia por siempre acompañarme en cada uno de mis sueños.

A mi tío Paulo Cesar que desde el cielo me guía y me protege para seguir adelante en el desarrollo de mis proyectos.

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios todopoderoso por brindarme salud y fortaleza en cada momento de mi vida estudiantil.

Agradezco a mis padres Luis y Mónica por ser los principales motivadores en este proceso, por guiarme en el camino del bien con sus valiosos consejos; gracias por su confianza permitieron que logre culminar mi carrera profesional.

Mi gratitud a la Universidad Nacional de Chimborazo por haberme permitido formarme, gracias a cada docente que fue participe de este proceso.

Un agradecimiento al Departamento de Tecnologías de la Información y Comunicación por el apoyo incondicional para poder culminar con éxito este proyecto.

Agradezco a Alex Manobanda por estar presente en el caminar de mi vida, apoyándome para continuar con mis metas.

Un agradecimiento a mi tutora de tesis Ph.D. Lorena Molina por compartirme su conocimiento, orientación y paciencia para poder realizar el proyecto de investigación.

INDICE GENERAL

INDICE DE TABLAS

INDICE DE FIGURAS

RESUMEN

Hoy en día la seguridad informática se ha convertido en un punto crítico de las organizaciones, debido a que la mayoría de los procesos laborales y académicos están soportados en infraestructuras informáticas, bajo redes de comunicación e internet, altamente expuestas a riesgos de seguridad de la información.

Por consiguiente, el presente proyecto de investigación se elaboró un manual de implementación de un proceso hardening para mitigar vulnerabilidades en el servidor web ngnix de la UNACH protegiendo así la seguridad de la información.

La investigación es inferencial por el análisis pre y post acerca de las vulnerabilidades, se realizó un análisis de la situación actual del servidor encontrando un total de 7 vulnerabilidades las cuales son vulnerabilidades de encabezado.

Finalmente, mediante Pentest Tools se realizó el escaneo de puertos abiertos: 80 y de puertos cerrados: 113 y 443; además se midió el uso del CPU de 1,8 se redujo a 0,078 %, así como el tiempo de respuesta del servidor obtuvo una mejora de 5,2 a 1,09 segundos, confirmando así la efectividad de la implementación del proceso de hardening.

Palabras Clave: Ataques informáticos, Hardening, Servidor Ngnix

ABSTRACT

Today, computer security has become a critical point in organizations because most of the work and academic processes are supported by computer infrastructures, under communication networks and the Internet, which are highly exposed to information security risks.

Therefore, in this research project, a manual for the implementation of a hardening process was prepared to mitigate vulnerabilities in the UNACH Nginx web server, thus protecting the security of the information.

The investigation is inferential due to the pre and post-analysis of the vulnerabilities; an analysis of the current situation of the server was carried out, finding a total of 7 header vulnerabilities.

Finally, using Pentest Tools, the scan of open ports was performed: 80 and closed ports: 113 and 443; In addition, the CPU usage from 1.8 was reduced to 0.078%, as well as the server response time improved from 5.2 to 1.09 seconds, thus confirming the effectiveness of the implementation of the hardening process.

Keywords: Computer attacks, Hardening, Ngnix server

Firmado electrónicamente por:
DANILO RENEE YEPEZ OVIEDO

Reviewed by:
Danilo Yépez Oviedo
English professor
UNACH0601574692

CAPÍTULO I

INTRODUCCIÓN

Desde el comienzo de la computación una de las razones fundamentales de la informática ha sido la seguridad de la información. El surgimiento de ataques virales, robo de información, o daño a equipos computacionales afectan hoy día a las empresas en torno a sus activos informáticos, limitando sus operaciones administrativas.

La seguridad informática es la ciencia encargada de los procesos y métodos que buscan procesar y transmitir la información, la principal acción de la seguridad informática es la de minimizar los riesgos [1].

Así pues, el hardening se presenta como una de las medidas más importantes de seguridad a implementar en las organizaciones, ya que permite establecer distintas barreras de protección frente a los posibles atacantes, ya sean tanto a nivel externo e interno algunos de ellos pueden ser simplemente personas que ocasionan daños sin intención o por indebida manipulación de los equipos informáticos.

Dentro del contexto global la seguridad siempre busca la gestión de riesgos, es decir, que se tengan mecanismos o acciones para evitar o prevenir ataques de la mejor forma. Según [2] la seguridad podría ser catalogada como la ausencia de riesgo, e involucra cuatro acciones principales:

- Prevenir el riesgo
- Transferir el riesgo
- Mitigar el riesgo
- Aceptar el riesgo

En la Universidad Nacional de Chimborazo (UNACH) se cuenta con un servidor en el cual se alojan distintos servicios tanto educativos como administrativos, los cuales permiten el correcto desarrollo de las actividades que se generan dentro de la institución. Como resultado de la pandemia generada por el COVID-19, las organizaciones tuvieron que realizar cambios profundos en su forma de gestionar el trabajo y la información, siendo uno de los sectores afectados la educación. Hoy en día los servicios web de la

Institución cuentan con una mayor afluencia de usuarios, por lo cual, es necesario aplicar procesos que provean seguridad a los datos que se almacenan en los servidores.

Por lo tanto, la aplicación del proceso hardening en la red de la UNACH permitió mitigar las vulnerabilidades del servidor institucional, mediante la implementación de un proceso que evalué el ataque informático y la determinación de los pasos a seguir en caso de que el servidor institucional sufriera un ataque cibernético.

PLANTEAMIENTO DEL PROBLEMA

1.1. Problema y Justificación

En el contexto actual de la pandemia originada por el virus del COVID-19 uno de los principales sectores afectados es el educativo, en todos sus niveles. Las instituciones pasaron de una modalidad de educación presencial a una virtual (online), a la cual también migraron los procesos administrativos. En consecuencia, las infraestructuras tecnológicas reciben en la actualidad un mayor flujo de información, con lo cual se han visto expuestas a ataques informáticos que pueden vulnerar los datos de sus sistemas institucionales y afectar el correcto desarrollo de las actividades de sus miembros.

La seguridad informática se ha convertido, por tanto, en un punto crítico de estas organizaciones, debido a que la mayoría de los procesos laborales y académicos están soportados en infraestructuras informáticas, bajo redes de comunicación e internet, altamente expuestas a riesgos de seguridad de la información. El proceso hardening, por ejemplo, es un proceso de seguridad informática cuyo objetivo es mitigar vulnerabilidades, mediante la eliminación de software, servicios, usuarios, entre otros, a fin de evitar que un ataque se concrete.

En los momentos actuales, donde el quehacer administrativo y académico dependen totalmente de la tecnología, es relevante la búsqueda de mecanismos de aseguramiento de la información y mitigación de riesgos tecnológicos. La reducción de vulnerabilidades fortalecerá la infraestructura de red LAN de la UNACH, así como impactará en los niveles de disponibilidad de los servicios que provee el servidor web Ngnix.

Por lo antes expuesto, la seguridad del servidor web Ngnix de la UNACH podría ser vulnerada, por tal razón se ha visto oportuno estudiar el proceso de hardening con el fin de elaborar un manual que documente dicho proceso, con el objetivo de mitigar vulnerabilidades para brindar mayor seguridad a la información.

1.2. Objetivos

1.2.1. Objetivo General

Elaborar un manual de implementación de un proceso hardening para la mitigación de vulnerabilidades en el servidor web nginx de la UNACH.

1.2.2. Objetivos Específicos

- Estudiar el proceso Hardening de servidores para la mitigación de ataques cibernéticos en entornos de redes LAN (Local Area Network).
- Identificar y evaluar las vulnerabilidades en el servidor web Nginx de la Universidad Nacional de Chimborazo.
- Implementar el proceso de hardening en un entorno simulado, para la minimización de riesgos en el servidor web Nginx.

CAPITULO II

MARCO TEÓRICO

2.1. Mitigación de vulnerabilidades en servidores web

Una infraestructura tecnológica es un conjunto de hardware y software que ayuda a la institución a prestar sus servicios a los usuarios finales, es decir, es donde se instalan las aplicaciones que necesita la organización para su funcionamiento y su gestión interna. Se constituye, por lo tanto, en los dispositivos que permiten la transmisión de la señal (líneas, microondas, satélites), el transporte del mismo (protocolos de comunicación y dispositivos de enrutamiento), así como los dispositivos de computación y los programas que están involucrados en el transporte de la información (sistemas operativos y protocolos de comunicación) que llega al usuario [3].

2.1.1. Servidor

Un servidor es un ordenador o una partición de este muy potente que se encarga de almacenar archivos y distribuirlos en internet para que sean accesibles a los usuarios, en el mundo de la informática se le llama servidor al programa que ofrece una serie de servicios a esto se suele acceder por medio de programas especiales que se denomina clientes el caso es que por extensión se suele denominar servidor al ordenador en el que funciona estos programas técnicamente un servidor es un equipo que tiene instalado un software que sirve recursos útiles o información que necesitamos [4].

2.1.1.1. Tipos de servidores

Los diferentes tipos de servidores pueden cumplir con múltiples tareas, por ejemplo, los servidores web o los servidores de correo; y otras menos conocidas, pero igual de importantes, como los servidores Dynamic Host Configuración Protocol o Domain Name System que se usan a diario de manera transparente al usuario. A continuación, se describe de manera general los diferentes tipos de servidores [5]:

Tabla 1: Tipos de servidores

Tipo de Servidor	Descripción
Servidor de Correo	Uno de los servicios más conocidos es sin duda el servicio de correo electrónico que se encarga de recibir, almacenar y acceder al correo, realmente no es un servicio único sino un grupo de servicios
Servidor VOIP	Es un servidor que permite utilizar la red local de datos IP para canalizar el tráfico telefónico tal como se trataráde una red telefónica convencional
Servidor Proxy	Permite acceder a otras redes a través de él, puede ser usado en una red local para permitir a las computadoras navegar en internet de forma controlada
Servidor VPN	Establece comunicaciones seguras mediante un túnel cifrado a través de una red insegura como Internet.
Servidor Web	Permite que las páginas web, las imágenes y videos puedan ser publicadas y distribuidas en internet, además permite dar cimientos a aplicaciones basadas en http.
Servidor DNS	Es un servidor similar a una base de datos que permite apartir de un dominio de internet poder localizar el servidor que aloja este sitio web o el sistema de correo.

2.1.2. Seguridad informática

Se puede definir a la seguridad informática como la disciplina encargada de plantear y diseñar las normas, procedimientos, métodos y técnicas, con la finalidad de obtener un sistema de información seguro, confiable y disponible a sus usuarios finales. La principal tarea de la seguridad informática es la minimización de los riesgos, los cuales provienen de muchas partes, por ejemplo: de la entrada de datos, del medio que transporta la información, del hardware que es usado para transmitir y recibir, de los mismos usuarios y protocolos implementados [6].

Tiene como objetivo asegurar los pilares de seguridad de la información mediante la aplicación de una serie de medidas de control físico como el control de acceso y el cuarto de servidores. Los controles también se pueden crear de forma lógica mediante software y controles de acceso, como la clave de usuario y la biometría [7].

La seguridad de la información se ocupa de analizar y prevenir los riesgos, y debe encontrar rápidamente soluciones eliminarlos, si es necesario. Está se basa en 4 pilares fundamentales que se describen a continuación [8]:

Figura 1: Pilares de la seguridad de la información

Disponibilidad: Posibilidad de acceder a la información cuando se requiera, pero considerando la privacidad. Evitar "fallas" del sistema que permitan conexiones no autorizadas, de manera que se bloquee el acceso ilegítimo a los servidos (por ejemplo, al correo electrónico).

Confidencialidad: La información solo es accesible a personas autorizadas. La información no debe llegar a personas u organizaciones no autorizadas.

Coherencia: La información debe ser precisa sin cambios o errores no autorizados. Se refiere a la protección contra vulnerabilidades externas, o posibles errores humanos.

Autenticación: La información tienen como procedencia el usuario correcto, debiendo proveer mecanismos de validación de identidad, para garantizar que la fuente de datos sea correcta. [9]

2.1.2.1. Tipos de seguridad

Existen dos tipos de seguridad:

Activa: Comprende las medidas destinas a evitar o reducir los riesgos que amenazan a los sistemas.

Pasiva: Comprende las medidas destinadas a minimizar el daño una vez producido el incidente de seguridad [10].

2.1.3. Vulnerabilidad Informática

La vulnerabilidad de seguridad es un componente del código o del software que identifica los defectos de la seguridad de las aplicaciones, sistemas y redes para que los cibercriminales puedan beneficiarse de ellos [11]. Por lo tanto, una vulnerabilidad informática puede definirse como un fallo en un sistema que puede ser explotado por un atacante, situación que genera un riesgo para la organización o para el mismo sistema.

2.1.3.1. Tipos de vulnerabilidades

Existen dos tipos de vulnerabilidades informáticas: lógicas y físicas.

Vulnerabilidades Lógicas

Las vulnerabilidades lógicas son las que van a afectar directamente la infraestructura y el desarrollo de la operación de estos, estas pueden ser de [6]:

- *Configuración:* Configuraciones por defecto del sistema o incluso de algunas aplicaciones del servidor que se tenga expuesta.
- *Actualización*: Hay empresas que no actualizan sus sistemas, lo cual genera que surjan vulnerabilidades.
- *Desarrollo:* se puede mencionar, por ejemplo, las inyecciones de código en SQL, Cross Site Scripting; puede variar dependiendo del tipo de aplicación y la validación de los datos.

Vulnerabilidades Físicas

Las vulnerabilidades físicas son las que van a afectar a la infraestructura de la organización de manera física, como ejemplo se podría mencionar una vulnerabilidad alta de este tipo si se vive en una zona de alto riesgo de sismos, ya que puede presentarse una negación en el servicio, una afectación en la disponibilidad [6].

2.2. Hardening de Servidores web

Hardening (palabra en inglés que significa endurecimiento), en seguridad informática, es el proceso de asegurar un sistema mediante la reducción de vulnerabilidades en el mismo. Según [12], hardening es la protección de un sistema o conjunto de sistemas informáticos mediante la aplicación de configuraciones de seguridad específicas para prevenir ataques informáticos.

El aseguramiento de servidores (conocido también como "Server Hardening") se ocupa de revisar todos estos componentes que forman parte de una solución, así como de testear si están correctamente configurados en los aspectos de seguridad a fin de que no existan grietas de esa índole [13].

Los beneficios que nos brinda el hardening de servidores es reducir los riesgos asociados con fraude y error humano, facilita un despliegue de configuración más limpio y seguro, y certifica que los recursos críticos se encuentren con parches actualizados y sean capaces de defenderse contra vulnerabilidades conocidas.

2.2.1. Actividades del proceso de hardening

El proceso de hardening o endurecimiento de un servidor, se le conoce como “refuerzo del sistema operativo”. Entre las actividades que pueden considerarse dentro de este, de manera general, se encuentran: la eliminación de cuentas sin uso, el empleo de políticas de contraseñas, el cierre de puertos de red sin uso, la administración de privilegios de usuarios, la eliminación de servicios no deseados, y la administración de actualizaciones [14].

1) y procedimientos de administración de cuentas de usuario, grupos, TCBS (Truste Base Computing), módulos de autenticación agregables y relaciones de confianza.

Administrar los paquetes Las actividades de hardening específicas a sistemas operativos Linux, según Maldonado [15], son las siguientes:

1) Asegurar las herramientas de desarrollo y compiladores.

2) Instalar y configurar Firewalls, Kits de Seguridad (antivirus, antispyware, antimalware, anti hackers, anti banners).
3) Usar herramientas para Pen Testing y Monitoreo.
4) Configurar protocolos, puertos y servicios (solo los necesarios).
5) Implementar esquemas de seguridad, DMZ (Demilitarized Zone), Front End / Back End, Router apantallado, proxys, Firewalls.

2.3.Herramientas utilizadas

2.3.1. Pentest Tools

En un programa de scaneo que realiza evaluaciones de seguridad de red externa de caja negra, a su vez ofrece compromisos de pentesting exitosos con velocidad, consistencia y flexibilidad superiores.

Cubre todas las etapas de un compromiso, desde la recopilación de información hasta el escaneo del sitio web, el escaneo de la red, la explotación y la generación de informes. [16]

2.3.2. Centos

Es una distribución de GNU/LINUX completamente gratuita, se deriva del código de fuente que lanza Red Hat, es un sistema operativo de código abierto conocida por su consistencia, estabilidad, administración fácil de usar [17].

2.3.3. OpenVas

Sistema Abierto para la Evaluación de Vulnerabilidades, está compuesto por una serie de servicios y herramientas de escaneo y administración de vulnerabilidades; su arquitectura es robusta y completa, siendo su componente más importante el Escáner OpenVAS, el cual es altamente eficiencia en la ejecución de NVTs (pruebas de vulnerabilidad en redes), mismo que obtiene actualizaciones diarias a través de OpenVAS NVT Feed [18].

CAPÍTULO III

DISEÑO METODOLÓGICO

3.1. Enfoque de la investigación

Mixta: El presente proyecto tiene varios propósitos, por una parte, la mitigación de vulnerabilidades en el servidor web Nginx de la UNACH, y por otra la presentación de un manual de buenas prácticas de seguridad; para el efecto, y de manera transversal, se recopilará información cuantitativa sobre la detección de vulnerabilidades.

3.2. Alcance de la investigación

Descriptiva: "Se orienta a describir el fenómeno e identificar las características de su estado actual. Lleva a las caracterizaciones y diagnóstico descriptivos" [19]. El interés de la investigación se centra en describir los hechos o situaciones observadas, así como generar un elemento propositivo (manual) que resuma técnicamente la propuesta de la autora.

3.3. Tipo y diseño de la investigación

3.3.1. Según el objetivo de investigación

Investigación Aplicada:

La investigación es de tipo aplicativa ya que se pretende identificar y evaluar las vulnerabilidades del servidor web Nginx de la UNACH, mediante la aplicación de las técnicas.

3.3.2. Según la fuente de investigación

Investigación Bibliográfica: La investigación documental es un procedimiento científico, un proceso sistemático de indagación, recolección, organización, análisis e interpretación de información o datos en torno a un determinado tema [20].

Para el desarrollo del proyecto se realizará la recolección de información teórica de postulados relacionadas al tema de investigación con base en artículos científicos, libros y revistas que permitan sustentar la base teórica y aplicativa del mismo.

3.3.3. Según el método a utilizar

Método Inductivo: Con la aplicación de este método se pretende conocer la realidad actual de la seguridad informática en el servidor, realizando el proceso Hardening y sin éste.

3.3.4. Según el tipo de datos

Cuasiexperimental: La investigación es de tipo cuasiexperimental ya que se procederá a identificar las vulnerabilidades del servidor web Nginx de la UNACH, una vez identificadas las vulnerabilidades evaluar cada una de ellas y a continuación aplicar el proceso de hardening. Los resultados serán procesados estadísticamente para la comprobación de la hipótesis de investigación.

3.4. Unidad de análisis

Servidor Web Nginx de la UNACH.

3.5. Población y muestra

De acuerdo con el tipo de investigación planteada, se trata de una población infinita puesto que se obtendrán datos de diferentes mediciones al servidor web Nginx de la UNACH.

La muestra que se tomó fue durante 6 días en 3 horarios para efectos de comprobación de la hipótesis de investigación.

3.6. Técnicas de recolección de datos

Para la caracterización de la LAN y del Servidor Web Nginx de la UNACH (diagnóstico inicial), se aplicará una entrevista de investigación con el responsable de departamento de TIC de la institución (ver Anexo 1).

Posteriormente, evaluación del impacto del proceso hardening se empleará software de monitoreo de red, así como comandos propios del sistema operativo. Estos datos serán registrados en una matriz o ficha de información, para su posterior procesamiento estadístico.

3.7. Técnicas de Análisis e Interpretación de Información

Para mitigar las vulnerabilidades en el servidor web, se realizarán las siguientes actividades:

- Estudiar los mecanismos de seguridad y herramientas que permitan la detección de vulnerabilidades en el servidor web Ngnix.
- Identificar las vulnerabilidades existentes en el servidor web Ngnix.
- Evaluar las vulnerabilidades que ponen en riesgo el funcionamiento del servidor web.
- Aplicar el proceso hardening, y evaluar la incidencia en el servidor web.
- Interpretar y analizar los resultados
- Elaborar un manual de buenas prácticas de seguridad.

3.8. Variables de investigación

3.8.1. Variable dependiente

Mitigación de vulnerabilidades

3.8.2. Variable independiente

Proceso Hardening

3.9. Hipótesis

Ho: La implementación de un proceso hardening no permite la mitigación de vulnerabilidades del servidor web nginx de la UNACH.

Hi: La implementación de un proceso hardening permite la mitigación de vulnerabilidades del servidor web nginx de la UNACH.

3.10. Operacionalización de variables

Tabla 2*:* Operacionalización de variables

	Variable	Definición Conceptual	Dimensión	Indicadores
Independiente	Proceso Hardening	Hardening se define como la protección de un sistema o conjunto de sistemas informáticos mediante la aplicación de configuraciones de seguridad específicas para prevenir ataques informáticos. [21]	• Configuraciones de seguridad	• Configuración actual del servidor • Actividades de protección realizadas.
Dependiente	Mitigación de vulnerabilidades en el servidor web.	La vulnerabilidad de seguridad es un componente del código o del software que identifica los defectos de la seguridad de las aplicaciones, sistemas y redes para que los cibercriminales puedan beneficiarse de ellos [22]	• Escaneo de vulnerabilidades • Consumo de recursos.	• Porcentaje de ataques mitigados. • Uso de CPU del servidor • Tiempo de Respuesta del servidor

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. Proceso Hardening de servidores para la mitigación de ataques cibernéticos en entornos de redes LAN.

El proceso de hardening en servidores web NGINX ha sido documentado técnicamente en varias investigaciones publicadas en revistas o repositorios, a nivel nacional e internacional. Con el objeto de establecer un punto referencial sobre las actividades planteadas por dichos investigadores, tanto en el análisis de vulnerabilidades como en el proceso hardening, se procedió a efectuar la búsqueda mediante Google Académico, bajo los siguientes criterios de inclusión:

- *Cadena de búsqueda:* "hardening" and "servidor web" and "NGINX". Se empleó el operador AND para establecer, como requerimiento estricto, la inclusión de los tres términos en todos los posibles resultados de la búsqueda.
- *Temporalidad:* Publicado entre los años 2017 y 2021 (5 años).
- *Idioma:* Español.

Como resultado de la aplicación de estos criterios, se obtuvieron un total de 24 resultados coincidentes (ver Figura 2). Posteriormente, se procedió a revisar el contenido de cada una de las investigaciones, a fin de aplicar un segundo filtro bajo los siguientes criterios de exclusión:

- Investigaciones centradas en procesos de revisión documental/bibliográfica.
- Investigaciones cuyas variables de estudio no se centran en los términos de la cadena de búsqueda.

De la revisión de los contenidos de las investigaciones mencionadas, se seleccionaron finalmente un total de 5 investigaciones, las cuales se describen en la Tabla 3.

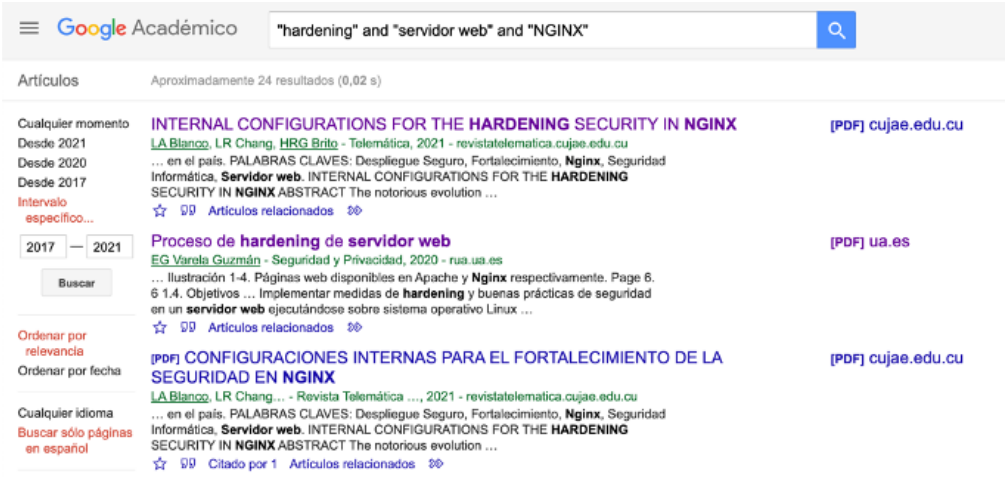

Figura 2: Resultados del proceso de revisión documental/bibliográfica

Como puede observarse en la Tabla 3, la evaluación de vulnerabilidades es un proceso que puede realizarse en dos fases de la investigación: previo al proceso de hardening (para establecer medidas correctivas a las configuraciones por defecto o modificadas del servidor) o posterior al proceso de hardening (a fin de evaluar su efectividad en la mitigación de vulnerabilidades). La hipótesis de la presente investigación requiere de la aplicación de ambos tipos.

Así también, los procesos de hardening planteados por los investigadores podrían categorizarse en:

- Configuraciones del Sistema Operativo
- Configuraciones de las peticiones http
- Configuraciones del servidor web (apache / Nginx)
- Configuraciones de base de datos

Estos procesos han de ser considerados dentro de la implementación del proceso hardening (objeto de estudio), y el desarrollo del correspondiente manual. En cuanto al software empleado en las fases pre y post hardening de estos estudios, se analizaron y compararon sus funcionalidades, a fin de identificar las mejores opciones para su aplicación en el presente proyecto.

Tabla 3: Investigaciones relacionadas al proceso hardening de servidores para la mitigación de ataques cibernéticos en entornos de redes LAN

Investigador (es) /año publicación	Evaluación de vulnerabilidades (pre hardening)		Proceso Hardening	Pruebas post hardening y resultados
Barboza y Delgado (2017)	Pentest Tools	Puertos abiertos y versiones de los servicios levantados.	– Actualización del sistema operativo – Creación de usuario con privilegios, para evitar el uso de root de conexiones remotas. – Instalación y configuración de *fail2ban* (control de ataques de fuerza bruta sobre servicios web primordiales, como: ssh, ftp, etc…) – Aseguramiento de SSH – Configuración de archivo httpd.conf – Configuración del archivo php.ini – Instalación y configuración de *portsentry* (protección contra ataques de barrido de puerto y ataques de DDoS) – Instalación y configuración firewall de aplicación modsecurity (protección contra ataques de fuerza bruta o ataques DoS)	*Pruebas:* mediante bWAPP (aplicación web vulnerable a nivel de OS Command Injection, XSS – Reflected, y PHP Code Injection) *Resultados:* "(…) el servidor está evitando ataques que pueden ocasionar denegación de servicios (…) bloqueando las conexiones remotas que intentan accesos por fuerza bruta" (Barboza y Delgado, 2017, p.42).
	Nessus	Análisis de vulnerabilidad aplicando políticas de Web Application Test.		
Aguilera et al. (2021)	Ninguna, el análisis de los autores en torno a las vulnerabilidades de estos entornos se basa en los reportes de cvedetails.com Ninguna		*Seguridad en la gestión de peticiones HTTP* – Denegar agentes de usuario automatizados – Desactivar los métodos HTTP no utilizados – Limitar el número de peticiones por direcciones IP – Establecer límites al buffer *Seguridad en la gestión de respuestas HTTP* – Habilitar la conexión segura con SSL/TLS – Eliminar los encabezados que exponen la versión de la tecnología base – Incorporar encabezados de respuesta HTTP de seguridad *Configuraciones bases del servidor web*	*Pruebas:* ninguna, solo expone una serie de medidas para incrementar la seguridad a bajo nivel en instalaciones de Nginx.

		– Gestionar los errores mediante páginas generales – Configurar el registro de los eventos – Crear usuario no privilegiado para los procesos – Restringir el acceso a la interfaz de administración – Denegar ejecución de scripts de directorios	
Guajala (2018)	No especificado (Revisión analítica de las vulnerabilidades del Owasp Top 10 -2013)	– Cargar los módulos security2_module, proxy_module, y proxy_http_module de apache (v. 2.2.15) – Parametrizar a apache como un proxy inverso – Habilitar el protocolo HTTPS para la comunicación entre el cliente y el proxy reverso, y el soporte del HSTS. – Instalar Mod_security y descargar mod_security CSR – Identificar el Directorio de las reglas genéricas del proyecto OWASP Mod_security CSR. – Analizar y parametrizar los archivos de configuración de Mod_security	*Pruebas:* mediante DVWA (aplicación web para la evaluación de vulnerabilidades, como sql injection, CAPTCHA inseguro, XSS, entre otros) y Wireshark (captura de paquetes HTTPS). *Resultados:* De la evaluación de vulnerabilidades, los autores concluyen que el proxy reverso Apache+Mod_security, es más efectivo que Mginx+Naxsi, ya que detecta el 90% de los ataques y contrarresta el 80%. Se propone, por lo tanto, el proceso hardening en una infraestructura de red basada en dicho proxy.
Varela (2020)	Las mismas pruebas que en la evaluación post hardening; proceso de diagnóstico inicial.	*Servidor web Apache* – Editando configuración por defecto de Apache – Verificando el usuario encargado de ejecutar apache – Configurando Headers de respuesta – Configurando TLS – Instalando y configurando mod_security *Servidor web Nginx* – Editando configuración por defecto de Nginx – Configurando Headers de respuesta – TLS – Instalando y configurando mod_security	*Configuración:* Simulación de la arquitectura de red mediante GNS3. Implementación de servidores Apache y Nginx en base a los reportes sobre popularidad de servidores web de w3techs.com *Pruebas:* mediante Arachni (escaneo de aplicaciones web y generación de reportes) y Nikto (detección de vulnerabilidades en sitios web, análisis de configuraciones *Resultados:* "menos vulnerabilidades encontradas, en comparación a un

	servidor que posee la configuración por defecto (…) los problemas encontrados en los procesos de escaneo de vulnerabilidades son consideradas vulnerabilidades menores" (Varela, 2020, pp.25-26)

4.2. Topología de la red del servidor de la UNACH

La figura 3, muestra la estructura de la UNACH en la cual se realizó el proceso de hardening con el fin de mitigar las vulnerabilidades**.**

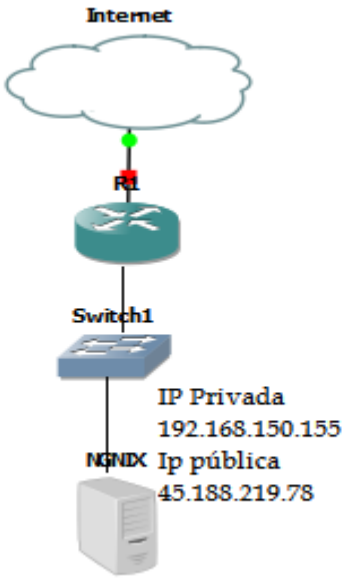

Figura 3: Estructura de la UNACH

Como se observa en el esquema de la figura 4, Nginx fue configurado como proxy de redireccionamiento SSL hacia el puerto 80 abierto a las peticiones http.

Como se puede visualizar en la figura 3 fue el escenario en donde se realiza el análisis de vulnerabilidades al servidor web NGNIX mediante la herramienta Pentest Tools se obtuvieron los siguientes resultados.

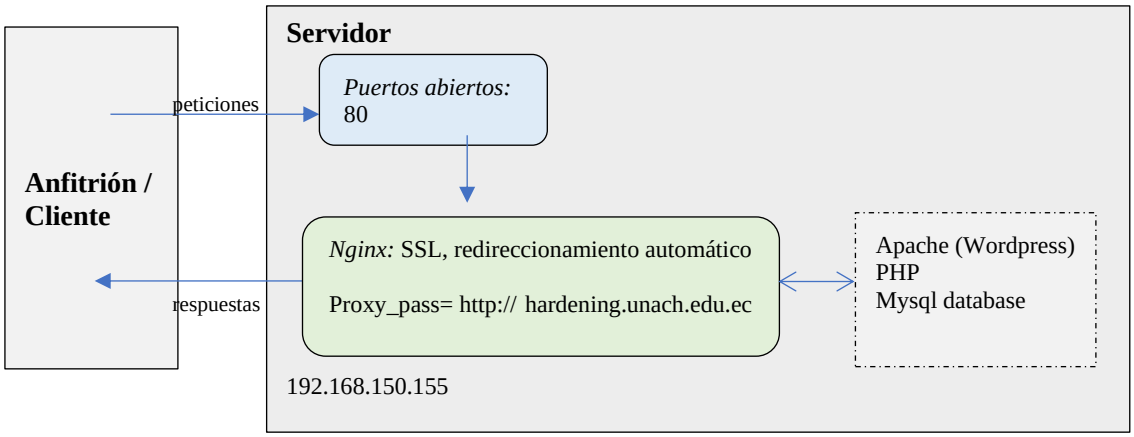

Figura 4:Esquema de configuración del entorno.

Tabla 4: Puertos abiertos

Puerto	Estado	Servicio
80/tcp	open	http
8008/tcp	open	http
9000/tcp	open	Cslistener

Se presenta una tabla con los resultados de Pentest tools sobre la detección de vulnerabilidades en el servidor web. La tabla 5 describe y explica cada una de las vulnerabilidades identificadas.

Tabla 5: Vulnerabilidades detectadas con Pentest Tools

Item	Vulnerabilidad	Descripción
1	CVE-2021-21703	Configuración de cookies insegura falta. El código JavaScript se ejecuta dentro de la página web puede acceder a ella lo que podría provocar un secuestro de la sesión. [23]
2	CVE-2021-21707	**Strict-Transport-Security.** Permite proteger la comunicación de un sitio, cuando el mismo emplea el protocolo https, así mismo dentro de sus propiedades indica el tiempo en que podrá ser accedido solo por esta vía, y redirige a los usuarios a la página actual y segura.
3	CVE-2021-21706	**Content-Security-Policy.** Para escribir un archivo fuera del directorio de destino puede ser engañoso lo que podría causar la sobreescritura de archivos, sujeto a los permisos del sistema.

Item	Vulnerabilidad	Descripción
4	CVE-2015-9251	**X-Frame-Options.** Permite identificar si una web puede renderizar una página, la propiedad deny indica que la página no podrá ser cargada en otros marcos desde fuera.
5	CVE-2019-11358	**X-XSS-Protection.** El encabezado HTTP X-XSS-Protection detiene la carga de una página si se detecta un ataque de tipo Cross-Site o XSS. [24]
6	CVE-2020-11022	**X-Content-Type-Options.** Permite controlar la carga de contenido, o MIME. Donde algunos usuarios puedan hacer pasar un archivo por otro.
7	CVE-2020-11023	**Referrer-Policy** Permite que un atacante no autenticado inyecte Javascript en la aplicación a través de vulnerabilidades de Cross-Site Scripting (XSS).

Pentest Tools categoriza las vulnerabilidades en las siguientes 5 escalas (de menor a mayor impacto): “Info”, “Low”, “Medium”, “High”, y “Critical”. Puede observarse en las estadísticas presentadas en el servidor hardening.unach.edu.ec se encontraron 7 vulnerabilidades medium

Dentro de la información reportada, destaca la detección de tipo “medium” denominada: “HTTP TRACE/TRACK methods allowed” (Permite los métodos HTTP TRACE / TRACK). Según en ranking de prioridad de vulnerabilidades (NPR) se encuentra en el top 10, lo cual significa que debe ser atendida con prioridad sobre el resto de las

detecciones. La importancia de su desactivación radica en que, mediante él, se puede ejecutar un ataque web tipo XSS (Cross-site-scripting), comúnmente encontrada en servidores web apache.

4.3. Escenario antes de realizar el proceso hardening.

La figura 3, muestra la topología de la red de la Unach con el fin de medir los tiempos empleados en dar respuesta el servidor web, el uso de RAM y CPU, estos datos se tomaron en los diferentes tiempos.

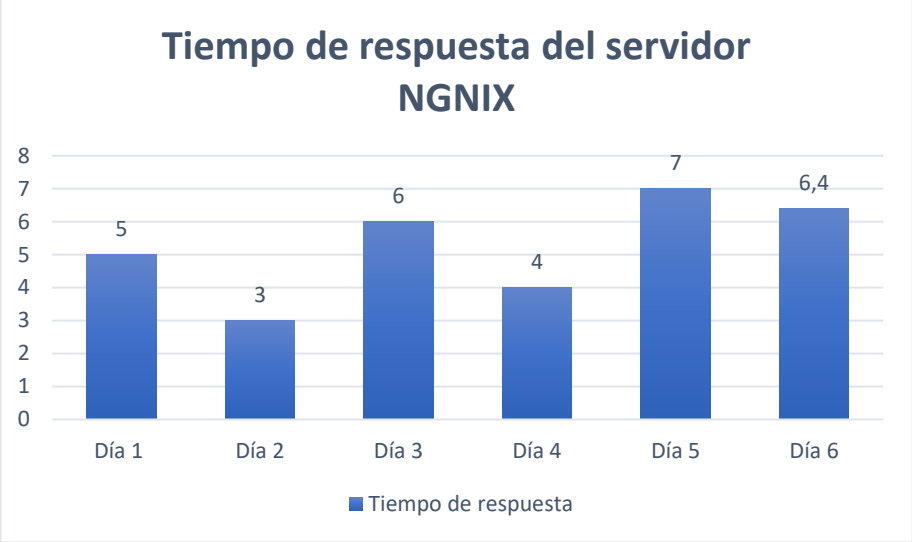

Figura 5:Tiempo de respuesta del servidor

La figura 5, muestra los tiempos de respuesta del servidor web Ngnix sin aplicar el proceso de hardening, obteniendo así un tiempo de respuesta del servidor de 5,23 segundos lo que se considera que los servicios del servidor web están lentos.

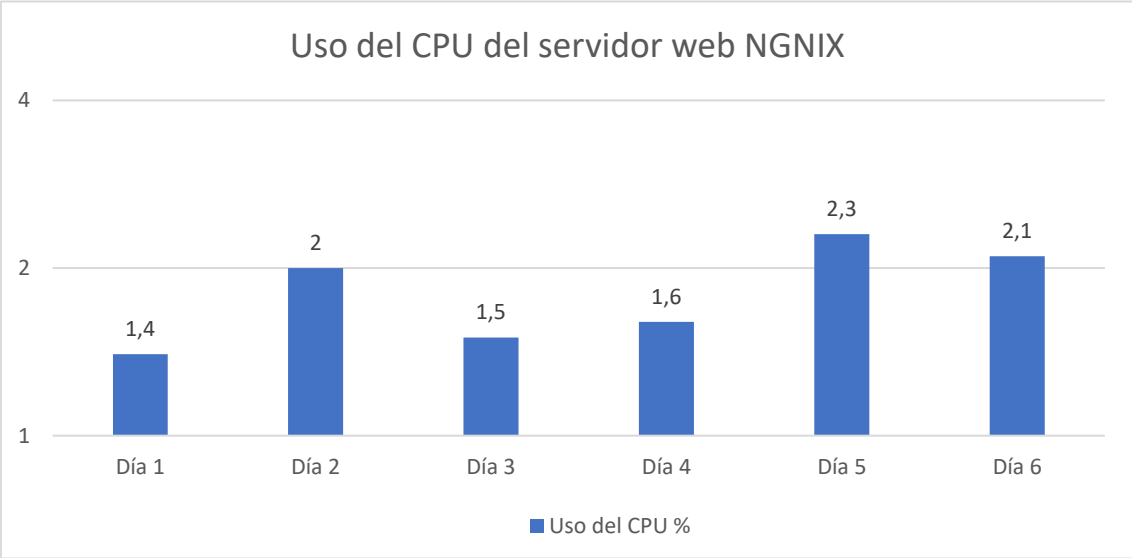

Figura 6:Uso del CPU del servidor.

En la figura 6, en el quinto día es donde el uso del CPU del servidor web Ngnix posee un valor más alto de 2,5 %.

4.4. Escenario realizado el proceso hardening.

En este escenario, se aplicó el proceso de hardening esto se implementó con el fin de mejorar la seguridad de la información en el servidor.

Como se observa en la Figura 7, aplicando los mecanismos de seguridad los tiempos de respuestas son más bajos siendo el más alto 1,4 segundos en el día, estos tiempos de respuestas ayudan a que el usuario que acceda al servidor web tenga una navegación de los servicios más agiles.

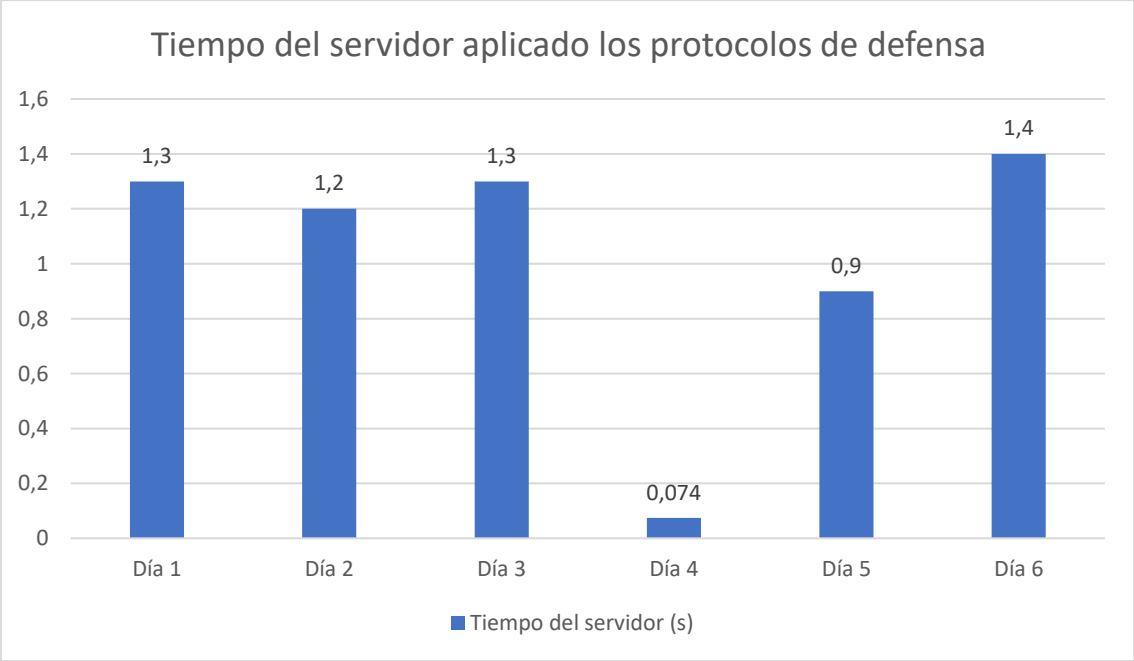

Figura 7:Tiempo de respuesta con mecanismos de seguridad

Se muestra en la figura 8, por parte del servidor web se puede evidenciar el comportamiento del CPU un menor siendo el pico más alto 0,09 %.

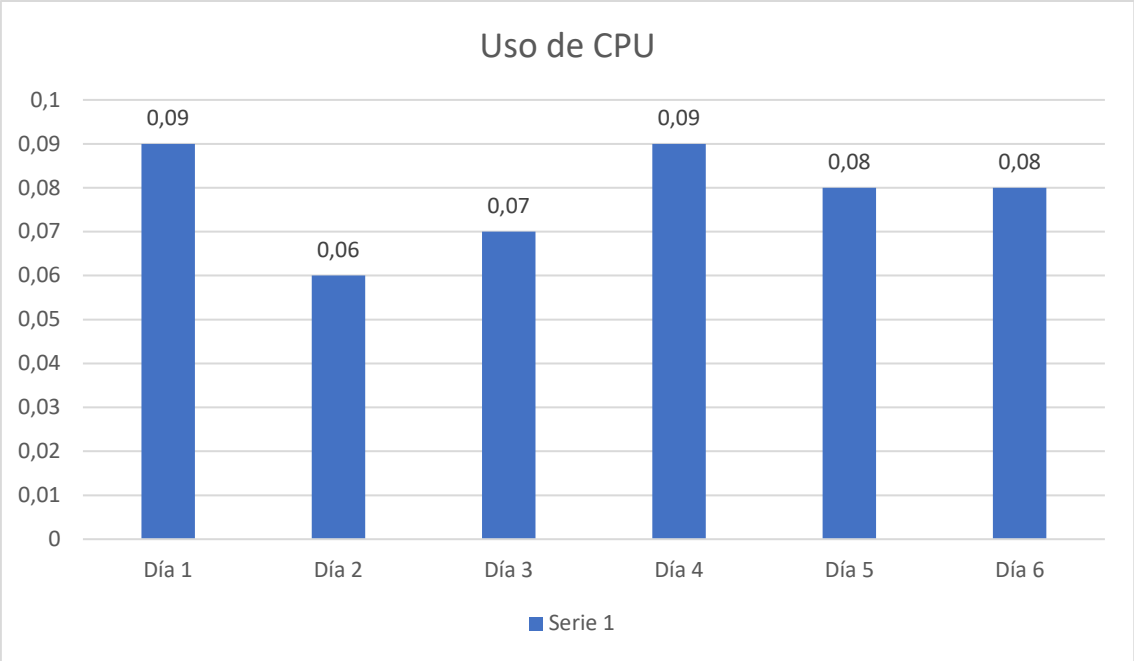

Figura 8:Uso del CPU con medidas de seguridad.

4.5. **Análisis de resultados**

Tabla 6: Cuadro comparativo de resultados

	Sin prototipo de seguridad		**Con proceso hardening**	
	Tiempo de respuesta (s)	Uso CPU (%)	Tiempo de respuesta (s)	Uso CPU (%)
Día 1	5	1,4	1,3	0,09
Día 2	3	2	1,2	0,06
Día 3	6	1,5	1,3	0,07
Día 4	4	1,6	0,074	0,09
Día 5	7	2,3	0,9	0,08
Día 6	6,4	2,1	1,4	0,08
Promedio	5,23	1,81	1,02	0,07

En la tabla 6 se puede visualizar un cuadro comparativo del tiempo de respuesta del servidor y del uso del CPU, cuando no se aplica el proceso de hardening y cuando si se aplica el proceso al servidor, en la figura se resumen estos datos.

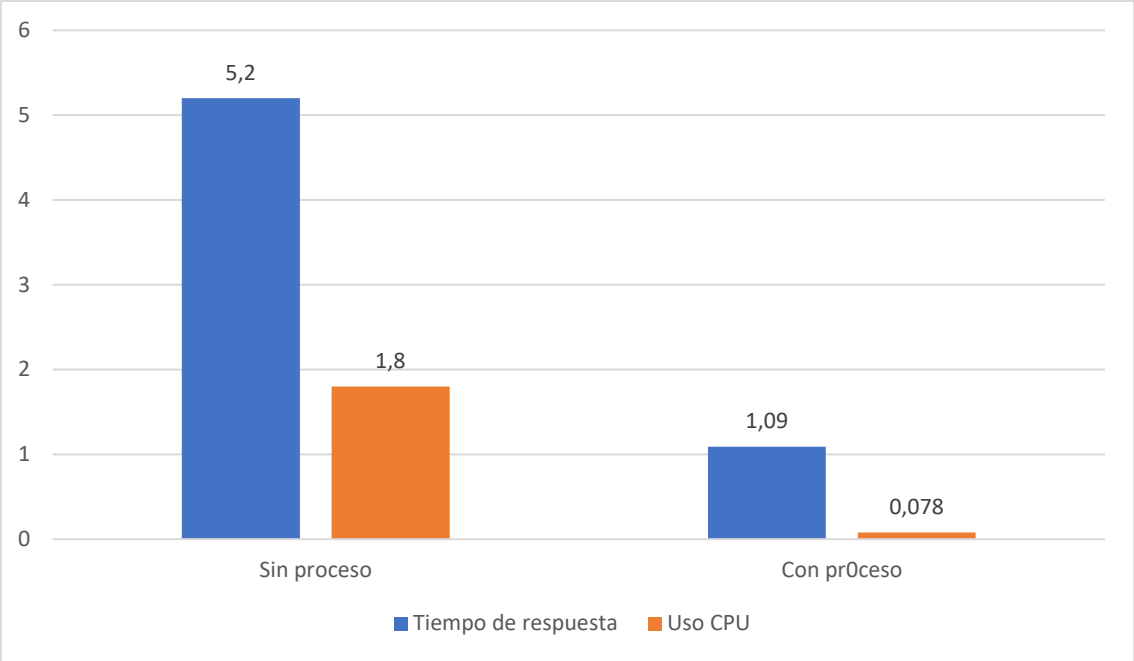

Figura 9:Comparación tiempos de respuesta del servidor y uso del CPU

CONCLUSIONES

- Al analizar las vulnerabilidades del servidor web Ngnix de la Universidad Nacional de Chimborazo se logró constatar que las vulnerabilidades más frecuentes son las de encabezado.
- Se evaluaron por medio de la simulación el antes y el después de la seguridad del servidor, así como la efectividad de las políticas aplicadas. También se evaluó la importancia de ajustar los parámetros que trae por defecto el servidor nginx demostrándose que es aconsejable elevar la seguridad de este.
- La implementación del proceso de hardening demostró que los tiempos de respuestas redujeron de 5,2 segundos a 1,9 segundos y también se redujo el uso del CPU de 1,8 a 0,078 lo que permite que los servicios del servidor sean más ágiles.
- La presente investigación permitió la conformación de un manual de hardening de servidor nginx para entornos lan/wan con la principal función de mejorar la seguridad de este, además, se demostró, por medio de la simulación, la eficacia de las mejoras introducidas en el proceso de hardening.

RECOMENDACIONES

- Se recomienda la implementación de un certificado SSL, permitiendo que los usuarios que accedan realicen sus operaciones y manipulen datos importantes con un protocolo de seguridad y privacidad idóneo.
- Se recomienda remover información o datos pertinentes que pueden sugerir al atacante características de hardware y software, vulnerabilidades que pueden ser aprovechadas por el mismo. A través de las siguientes líneas, o comandos puede ocultarse la información del servidor: **server_tokens off;**
- Prevenir el uso de anuncios o publicidad tipo malware mediante la agregación de esta línea: **add_header X-Frame-Options "SAMEORIGIN**". De esta manera el atacante no podrá sobrescribir los elementos gráficos o fronted como botones, haciendo que el usuario final de manera accidental presione el enlace oculto.

BIBLIOGRAFÍA

[1] R. Roque, Concientización y capacitación para incrementar la seguridad informática en estudiantes universitarios, Guadalajara: PAAKAT, 2018.

[2] M. Romero, INTRODUCCIÓN A LA SEGURIDAD INFORMÁTICA Y EL ANÁLISIS DE VULNERABILIDADES, Manabí: ÁREA DE INNOVACIÓN Y DESARROLLO, 2018.

[3] R. Acosta, La infraestructura de las tecnologías de la información y comunicación como mediadoras y el aprendizaje de la biología, Venezuela: Telos, 2014.

[4] F. d. A. L. Fuentes, Sistemas Distribuidos, México: Una Década, 2015.

[5] Á. D. León, «Qué es un servidor caracteristicas y tipos.,» 2019. https://blog.infranetworking.com/category/tutoriales-hosting/.

[6] M. Romero, Introducción a la seguridad informática y el análiss de vulnerabilidades, Manabí: Área de Innovación y Desarrollo, 2018.

[7] J. G. LONDOÑO, «ESTUDIO DEL ESTADO ACTUAL DE LA SEGURIDAD INFORMÁTICA EN LAS ORGANIZACIONES DE COLOMBIA,» 2020. https://repository.unad.edu.co/bitstream/handle/10596/36669/jgonzalezlon.pdf?sequence=1.

[8] «La Seguridad de la Información,» 2018. https://www.tecon.es/la-seguridad-de-la-informacion/.

[9] Tecon, «La Seguridad de la Información,» 2018. https://www.tecon.es/la-seguridad-de-la-informacion/.

[10] P. Palacio, Seguridad Informática, Paraninfo, SA., 2020.

[11] Ambit, «Tipos de Vulnerabilidades y Amenazas informáticas,» 2020. https://www.ambit-bst.com/blog/tipos-de-vulnerabilidades-y-amenazas-inform%C3%A1ticas.

[12] L. M. Rodríguez, «La importancia del bastionado de sistemas,» 18 05 2015. https://www.incibe-cert.es/blog/importancia-bastionado-sistemas.

[13] Arduinosecurity, «Aseguramiento de servidores,» 2016. https://arduinosecurity.com/es/services/hardening.

[14] A. I. Empresarial, «Aseguramiento de servidores hardening,» 2017. https://asesoriaitempresarial.com/index.php/aseguramiento-de-servidores-hardening/.

[15] M. E., «Seguridad Informática,» 2016. https://slideplayer.es/slide/9342436/ .

[16] M. Á. Camacho, «Pentest-Tools.com,» 2013. https://pentest-tools.com/.

[17] R. Demetrio, Diseño de un cluster de alta disponibilidad para un entorno educativo virtual universitario., Venezuela : INGENIERIA UC , 2018.

[18] A. E. C. Quezada, «Open Vulnerability Assessment,» 2016. [En línea]. Available: http://www.reydes.com/archivos/slides/webinars/AC_WG_OpenVAS_v2.pdf.

[19] H. S. Carlessi, «Manual de térnimos en investigación científica, tecnológica y humanística,» 2018. [En línea]. Available: https://www.urp.edu.pe/pdf/id/13350/n/libro-manual-de-terminos-en-investigacion.pdf.

[20] R. Maradiaga, «Técnicas de investigación documental,» 2015.. https://repositorio.unan.edu.ni/12168/1/100795.pdf.

[21] A. Martinez. https://www.moebio.uchile.cl/54/martinez.html.

[22] Ambit, «Ambit Technology,» https://ambit.com.mx/.

[23] INCIBE, «INCIBE-CERT,» 25 10 2021. https://www.incibe-cert.es/content/boletin-vulnerabilidades-7109.

[24] Hautes-Alpes, «ProHacKtive,» 20 04 2019.. https://kb.prohacktive.io/index.php?action=detail&id=CVE-2019-11358&lang=es.

[25] N. Yanza, «Entorno virtual de aprendizaje Estudios Sociales para fortalecer la Democracia y Participación en estudiantes de décimo año utilizando MOODLE,» 2020. http://repositorio.uisrael.edu.ec/bitstream/47000/2663/1/UISRAEL-EC-MASTER-EDUC-378.242-2020-137.pdf.

[26] P. Palacios, Seguridad informática, Paraninfo, SA., 2020.

[27] R. Demetrio, Diseño de un cluster de alta disponibilidad para un entorno educativo virtual universitario, Venezuela: INGENIERÍA UC 2018, 25(1), 2018.

[28] R. Roque, Concientización y capacitación para incrementar la seguridad informática en estudiantes universitarios, México: PAAKAT, 2018.

[29] M. Á. Camacho, «Pentest-Tools.com,» 2013..

ANEXOS

Anexo 1: Cuestionario guía de la entrevista de investigación.

UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

CUESTIONARIO GUÍA PARA ENTREVISTA DE INVESTIGACIÓN

Nombre del entrevistado: Ing. Diego Chicaiza

Tipología de red (equipos, conexiones, seguridades, LAN/WAN)

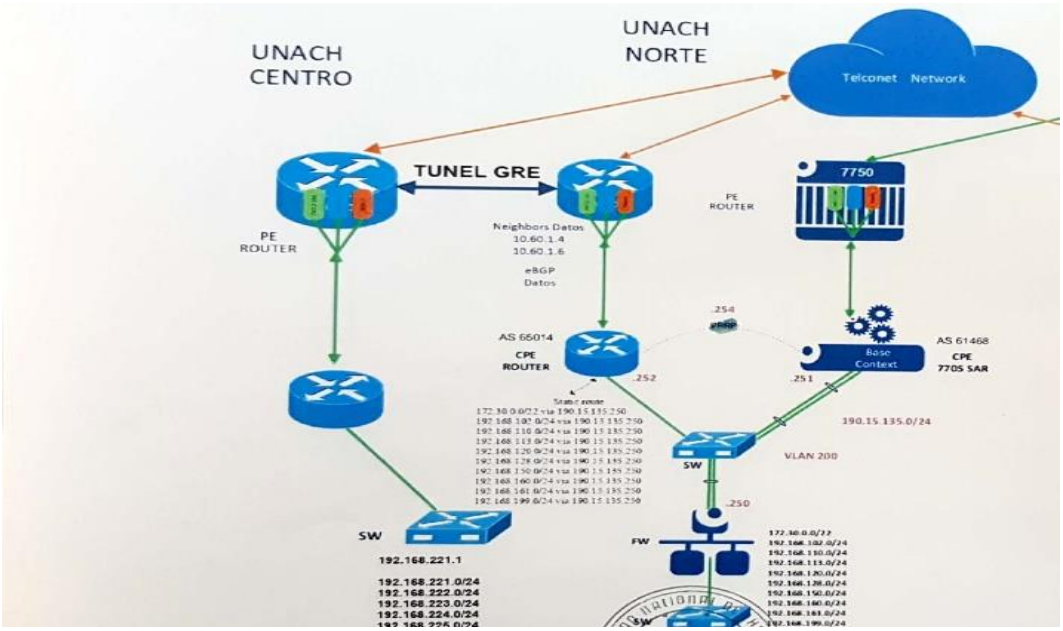

Figura 10: Diagrama Wan Institucional

Configuración del servidor web Ngnix (servicios, puertos abiertos/bloqueados,seguridad)

Puertos Abiertos
80 y 443

Anexo 2: Manual de implementación del proceso hardening.

UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

MANUAL DE IMPLEMENTACIÓN DE UN PROCESO HARDENING PARA MITIGAR VULNERABILIDADES EN EL SERVIDOR WEB NGINX DE LA UNACH.

Autora:

Karina Paola Pinduisaca Guashpa

Tutor:

Ing. Lorena Molina

RIOBAMBA- ECUADOR. 2022

Contenido

1. Requerimientos mínimos de Software y Hardware

En la Universidad Nacional de Chimborazo en el departamento de tecnologías de la información y comunicación se procedió a la instalación de un servidor virtualizado con las siguientes características:

1.1. Hardware

- RAM 2GB
- CPU 4
- Disco Duro 70 GB

1.2. Software

- Sistema Operativo Centos 7

2. Instalación del servidor Ngnix

Para instalar el servidor nginx y el sitio web de Wordpress, en el servidor de la universidad se implementó el uso de VPN, PuTTY y WinSCP, estos programas de terceros permitieron el acceso al servidor, así como la manipulación y transferencia de archivos.

Para acceder al Servidor de la Universidad, usar FortiClient VPN, así como las credenciales dadas por el departamento universitario.

2.1. Descarga de FortiClient

Para descargar FortiClient se puede realizar desde la siguiente página https://www.fortinet.com/lat/support/product-downloads y aparece la siguiente pantalla.

Figura 11:Pantalla de inicio de FortiClient

A continuación selecionar la plataforma como puede ser Windows 32-bit, Windows 64-bit MacOs. Presionamos click en el botón de descargar y empieza la descarga automaticámente.

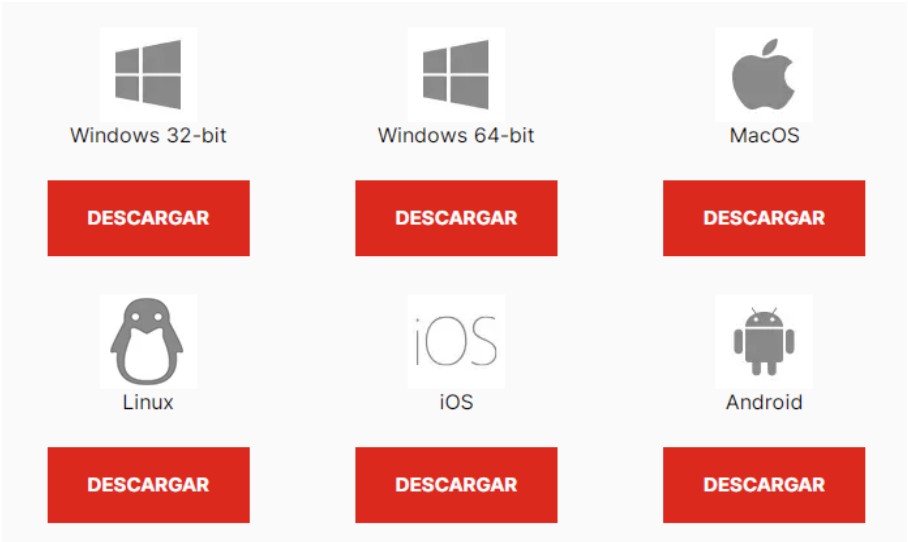

Figura 12: Pantalla de descarga del programa FortiClient

2.2. Instalación de FortiClient

Posteriormente se detalla paso a paso la instalación, una vez descargado el programa ejecutar como administrador.

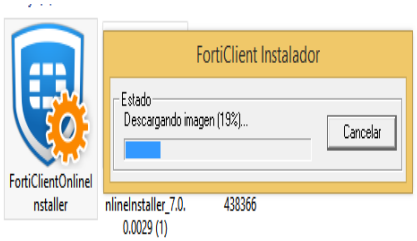

Figura 13: Programa a ejecutar.

Una vez descargado los componentes presionar en el botón yes para empezar la instalación.

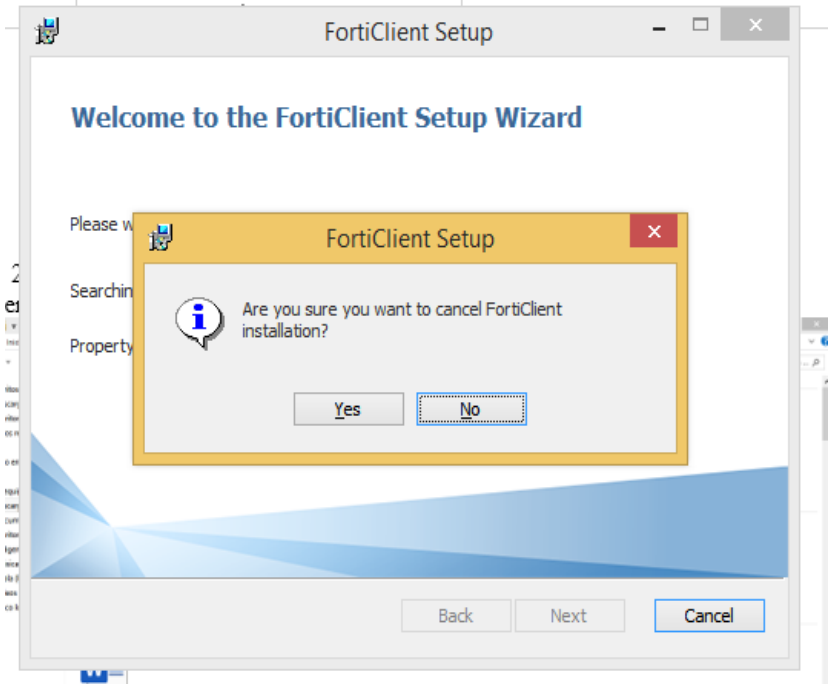

Figura 14: Pantalla de inicio de la instalación.

Posteriormente marcar con un clic en los términos y condiciones y presionar clic en siguiente.

Figura 15: imagen de término y condiciones.

Seleccionar la ubicación donde se instalará el programa y presionar en next.

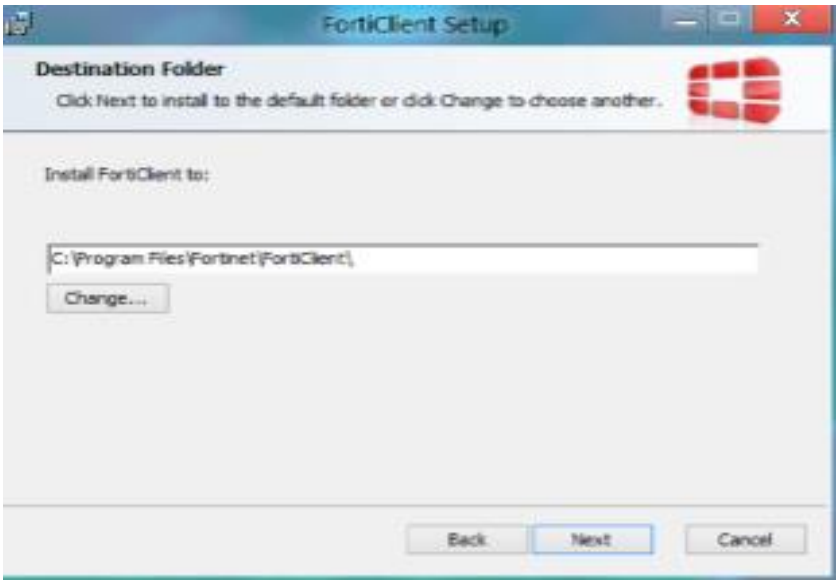

Figura 16: Pantalla de la ubicación del programa.

Posteriormente el programa ha sido instalado correctamente e iniciar la conexión.

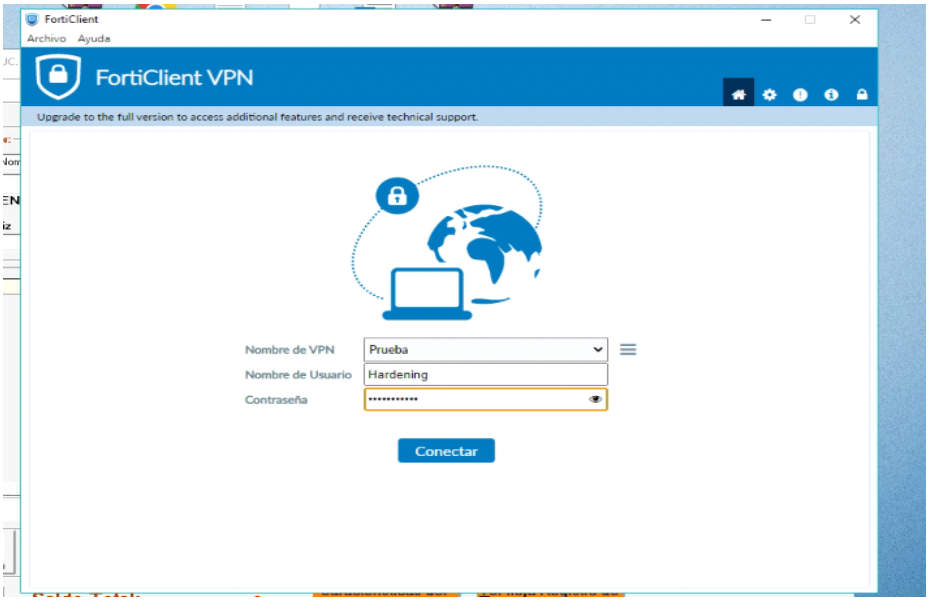

Figura 17: imagen de inicio con las respectivas credenciales.

Si las credenciales son correctas, se podrá acceder y tener conexión con el servidor, como indica la siguiente imagen.

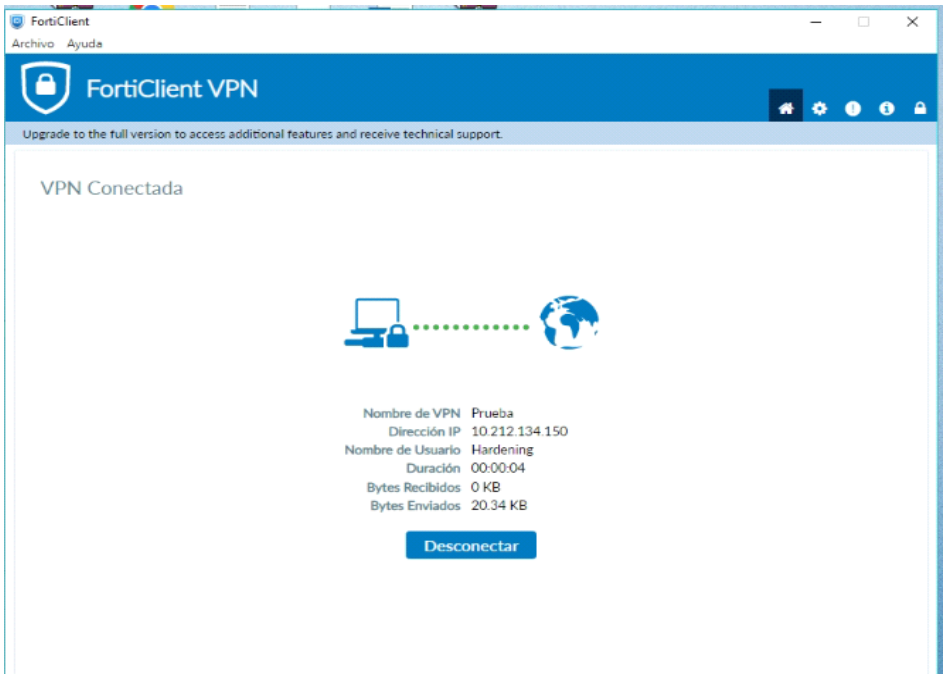

Figura 18: Conexión exitosa.

3. Instalación de Putty

Para la manipulación de archivos, y su correspondiente instalación, de la misma forma que con el programa FortiClient VPN, accedemos a instalar PuTTY.

Para la descarga de PuTTY descargar de la siguiente página https://www.usitility.com/es/putty/descargar-windows y presionars en descargar.

Figura 19:Pantalla de descarga del programa PuTTY

A Continuación, ejecutar el programa y presionar clic en next para el inicio de la instalación.

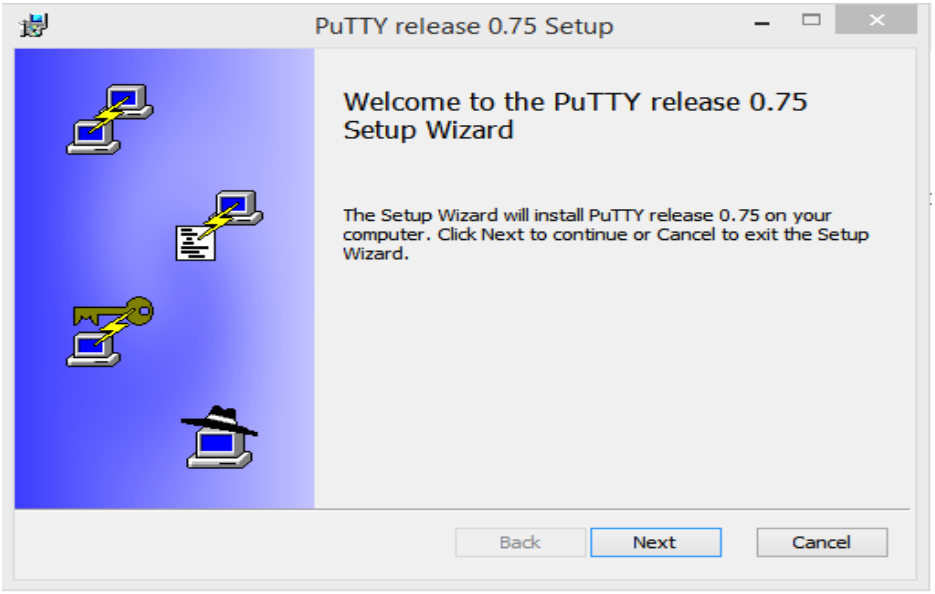

Figura 20: imagen del inicio de instalación de PuTTY.

Seleccionar la ubicación de la instalación del programa y presionar clic en next.

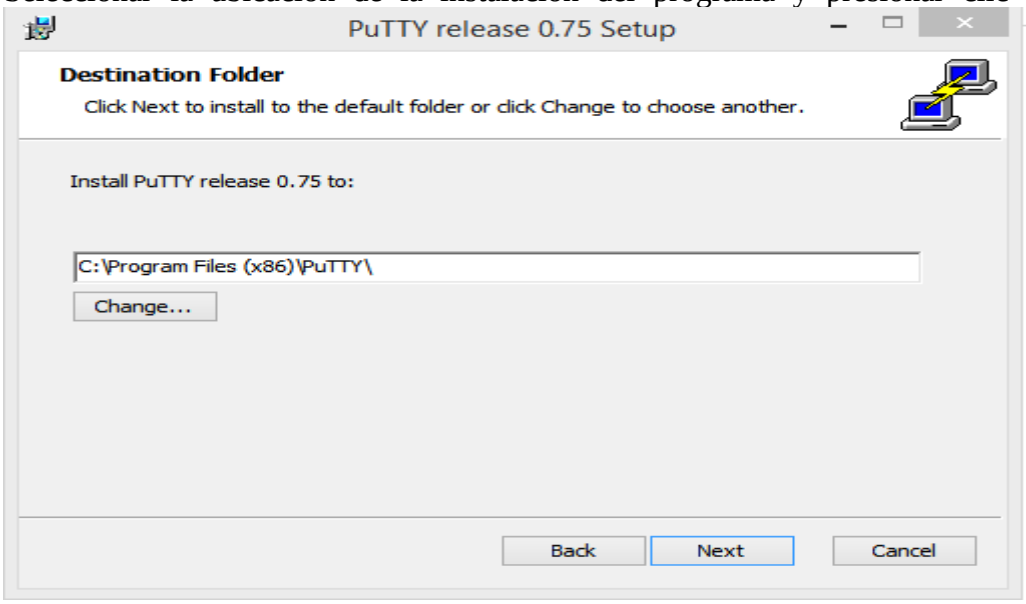

Figura 21: Ubicación del programa.

Finalmente, se ha concluido con la instalación del programa PuTTY.

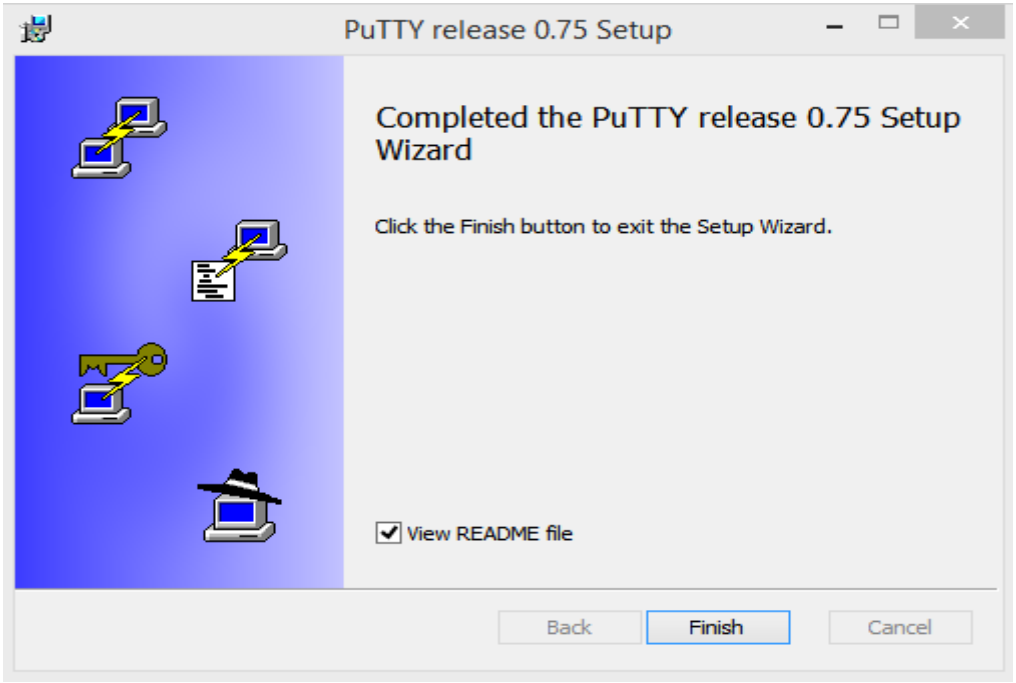

Figura 22: Imagen del programa instalado.

Posteriormente proceder con la conexión a la Ip 192.168.150.155

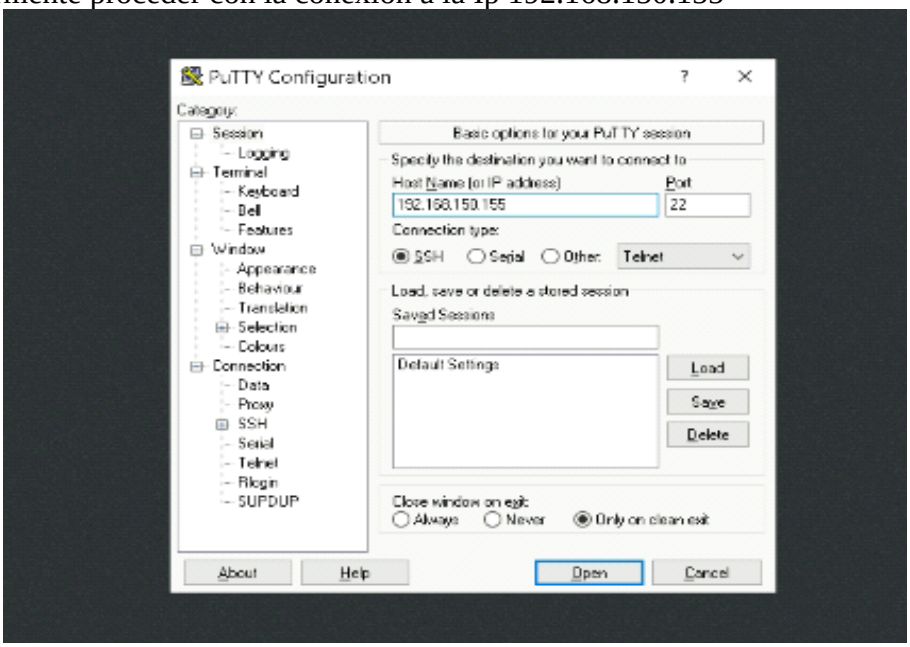

Figura 23: conexión a la Ip 192.168.150.155

4. Instalación de NGNIX

Nginx no está disponible por defecto en los repositorios de CentOS 7 está disponible en los repositorios EPEL. Para agregar los repositorios EPEL al sistema usar el siguiente comando: **yum install epel-release**

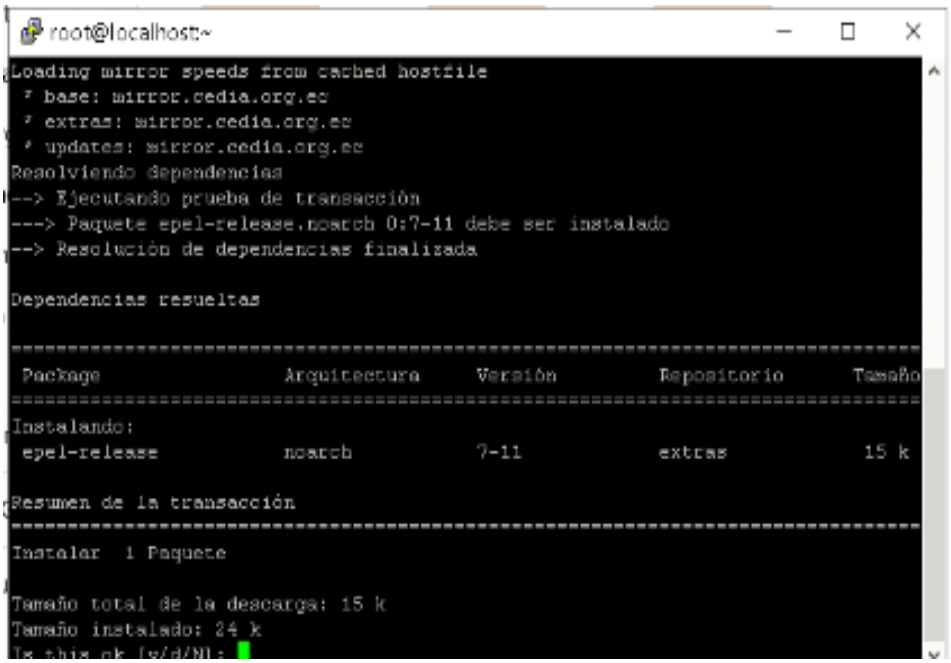

Figura 24: Repositorios EPEL

4.1. Inicio instalación Nginx

Ahora instalar Nginx con el siguiente comando **yum install nginx**

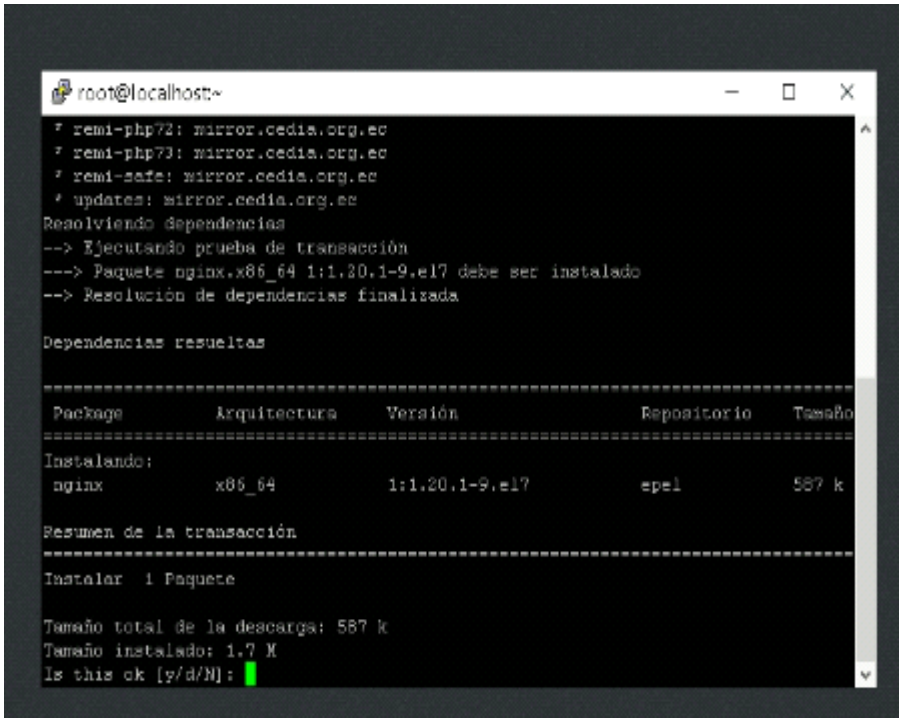

Figura 25: Instalación de Ngnix

Posteriormente iniciar nginx **systemctl start nginx** y activar **systemctl enable nginx**

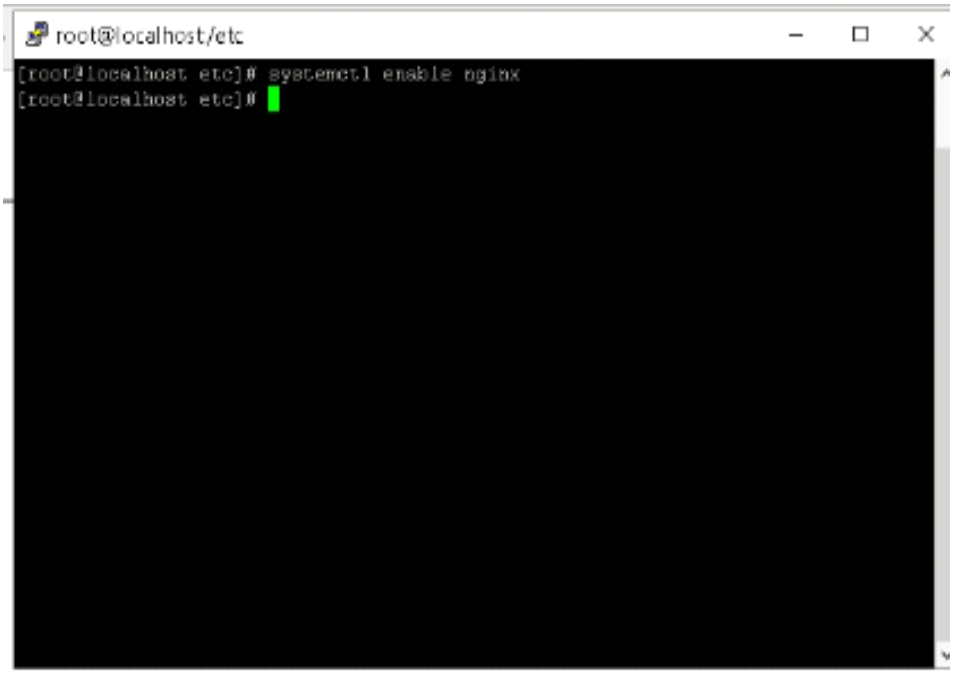

Figura 26: Inicio y activación de Nginx

Instalar un sistema de gestión de bases de datos derivado de MySQ como lo es **Mariadb**

yum install mariadb-server

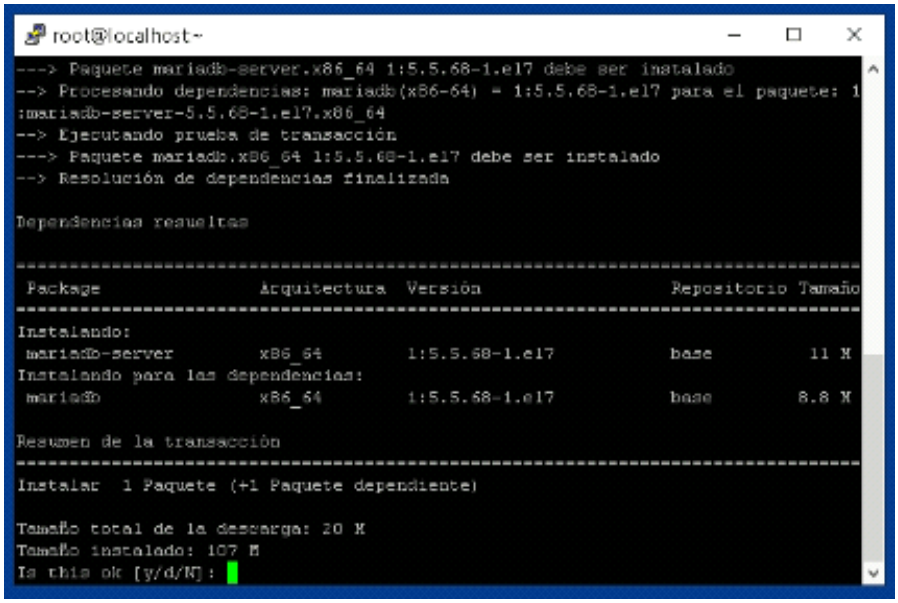

Figura 27: Instalación del sistema de base de datos.

Iniciar y activar la base de datos con los siguientes comandos.

systemctl start mariadb.service y **systemctl enable mariadb.service**

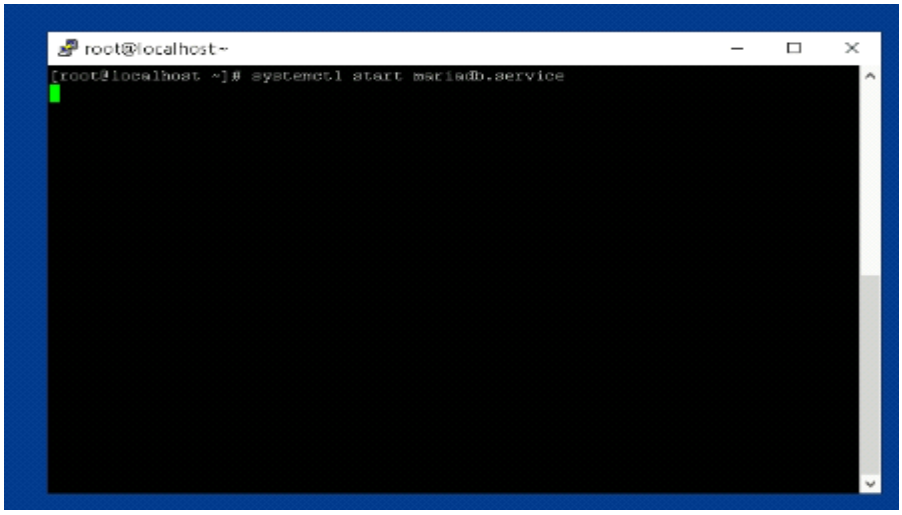

Figura 28: Inicio y activación de Mariadb-server.

CentOS 7 se envía con la versión 5.4 de PHP, que está en EOL durante bastante tiempo, por lo que usaremos el repositorio de Remi para instalar PHP.

Ejecute el siguiente comando para agregar el repositorio de Remi a su sistema: **yum install http://rpms.remirepo.net/enterprise/remi-release-7.rpm**

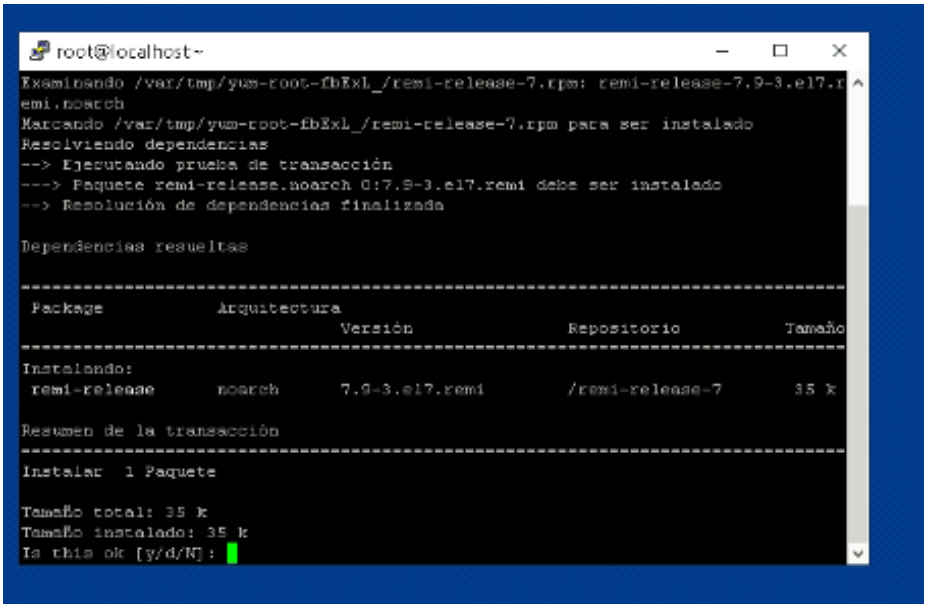

Figura 29: Instalación de PHP.

Una vez agregado, instalar el paquete yum-utils y habilite el repositorio remi-php72:

yum install yum-utils

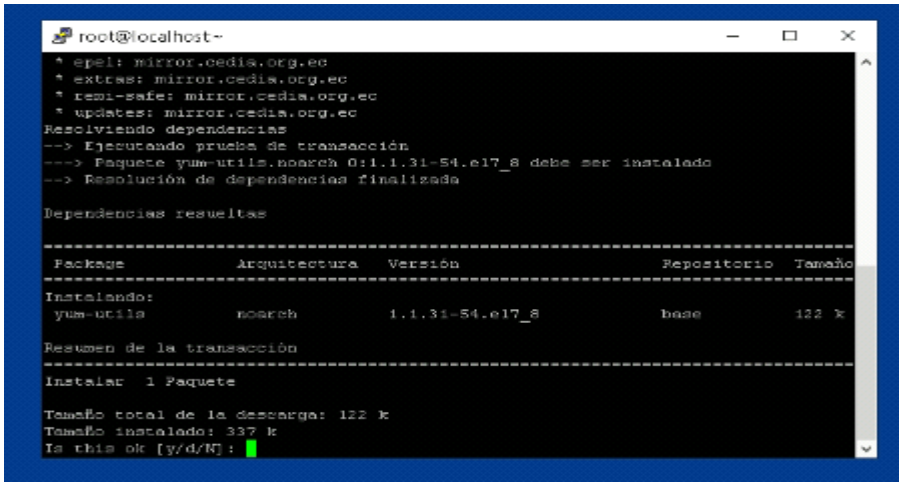

Figura 30: Instalación de paquetes yum-utils

Ahora habilitar el repositorio Remi, instalar PHP FPM y varios de los módulos PHP más comunes con:

yum install php-fpm php-opcache php-cli php-gd php-curl php-mysql

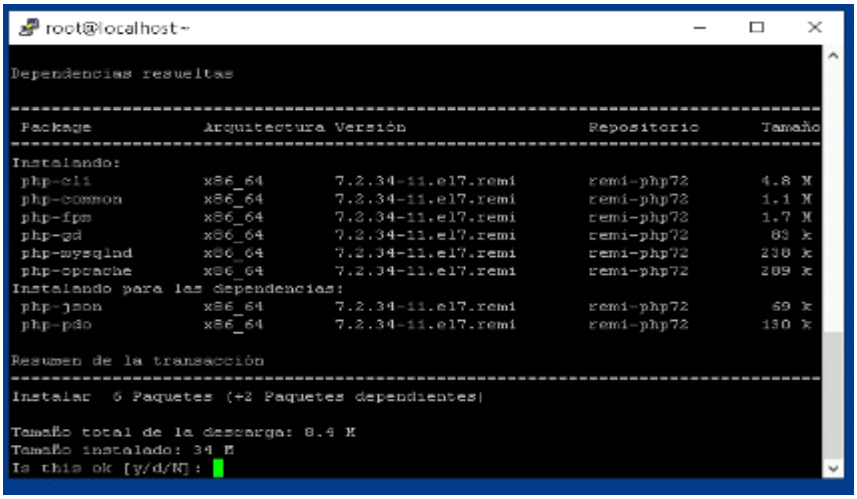

Figura 31:Instalación de módulos PHP.

De forma predeterminada, PHP FPM se ejecutará como usuario apache en el puerto 9000. Cambiar el usuario a nginx y cambiar del socket TCP al socket Unix. Para ello, editar las líneas:

nano /etc/php-fpm.d/www.conf, nano Is -la /etc/nginx/site-enable

user = nginx
group = nginx
listen = /run/php-fpm/www.sock
listen= 127.00.1:9000
listen.owner = nginx
listen.group = nginx

Asegurar de que el directorio /var/lib/php tenga los permisos correctos con el siguiente comando **chown -R root:nginx /var/lib/php**

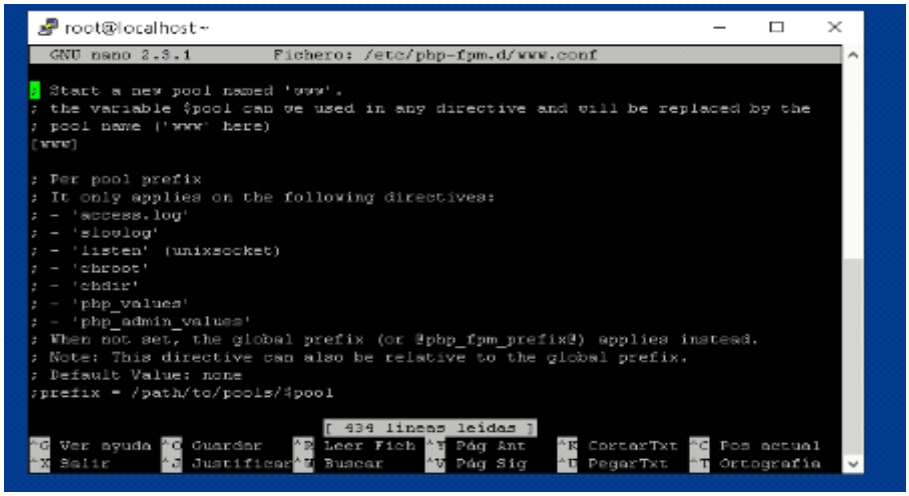

Figura 32:Comprobación de permisos.

Guardar el archivo, habilitar e iniciar el servicio PHP FPM con: **systemctl enable php-fpm, systemctl start php-fpm**

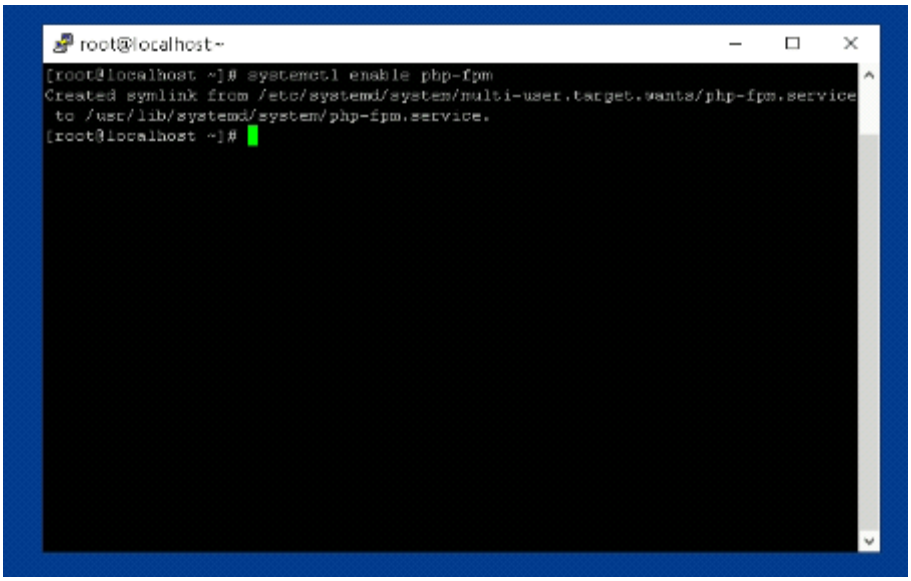

Figura 33:Inicio del servicio PHP FPM

4.2.Configuración de Nginx para procesar páginas PHP

Editar el archivo de configuración del bloque del servidor Nginx y agregar las siguientes líneas para que Nginx pueda procesar archivos PHP:

```
server {
  # other code
  location ~ \.php$ {
    try_files $uri =404;
    fastcgi_pass unix:/run/php-fpm/www.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
  }
}
```

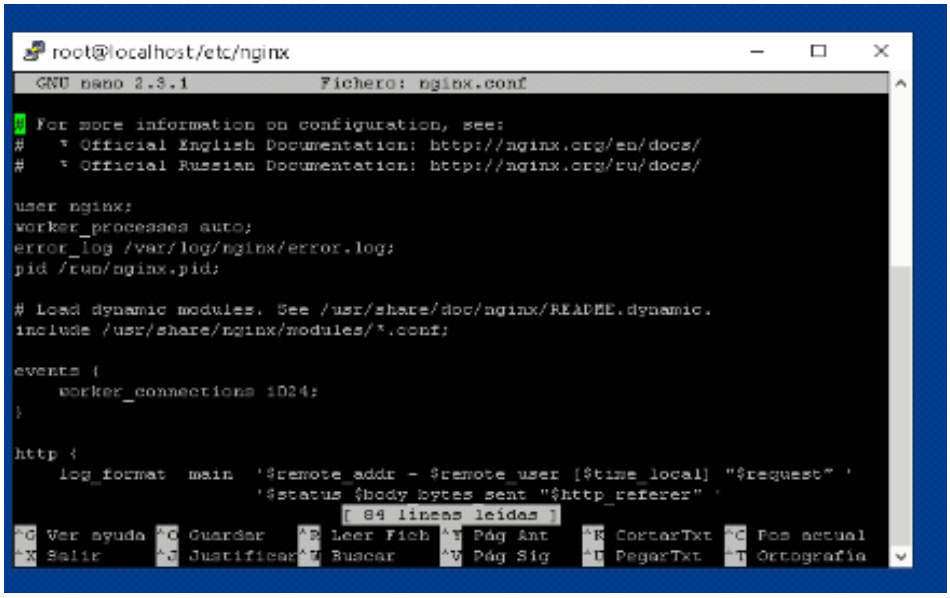

Figura 34:Archivo de configuración de Ngnix.

No olvidar iniciar el servicio Nginx: y verificar su estatus con las siguientes líneas de comandos.

systemctl start nginx, systemctl status nginx

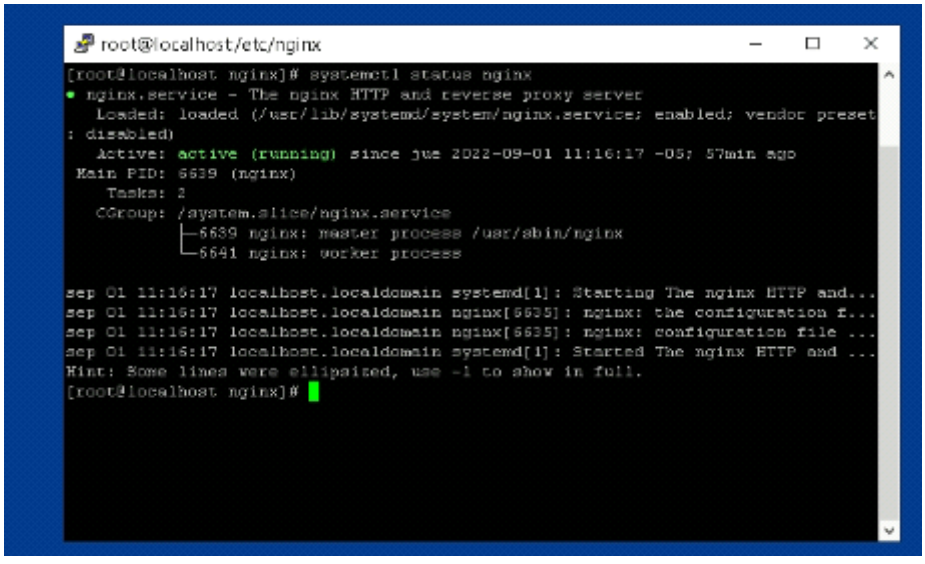

Figura 35: Inicio del servicio Ngnix.

4.3. Sitio web cargado en el servidor.

Como se puede evidenciar en las imágenes posteriores la página o sitio de tipo educativa, consta de las siguientes secciones: Inicio, Sobre Nosotros, Servicios y Contactos.

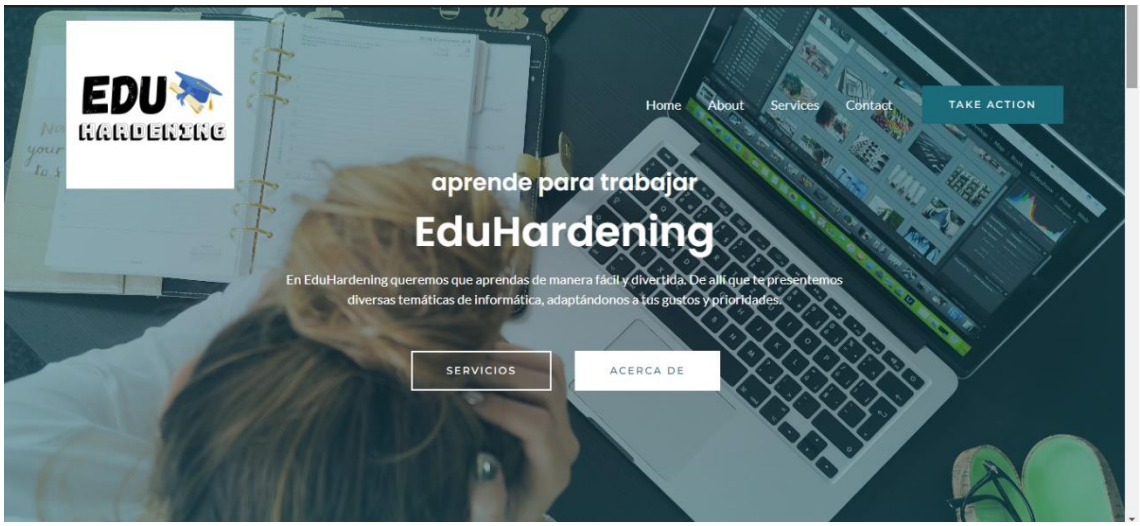

Figura 36: Pantalla principal del sitio web.

Figura 37: Pantalla servicios.

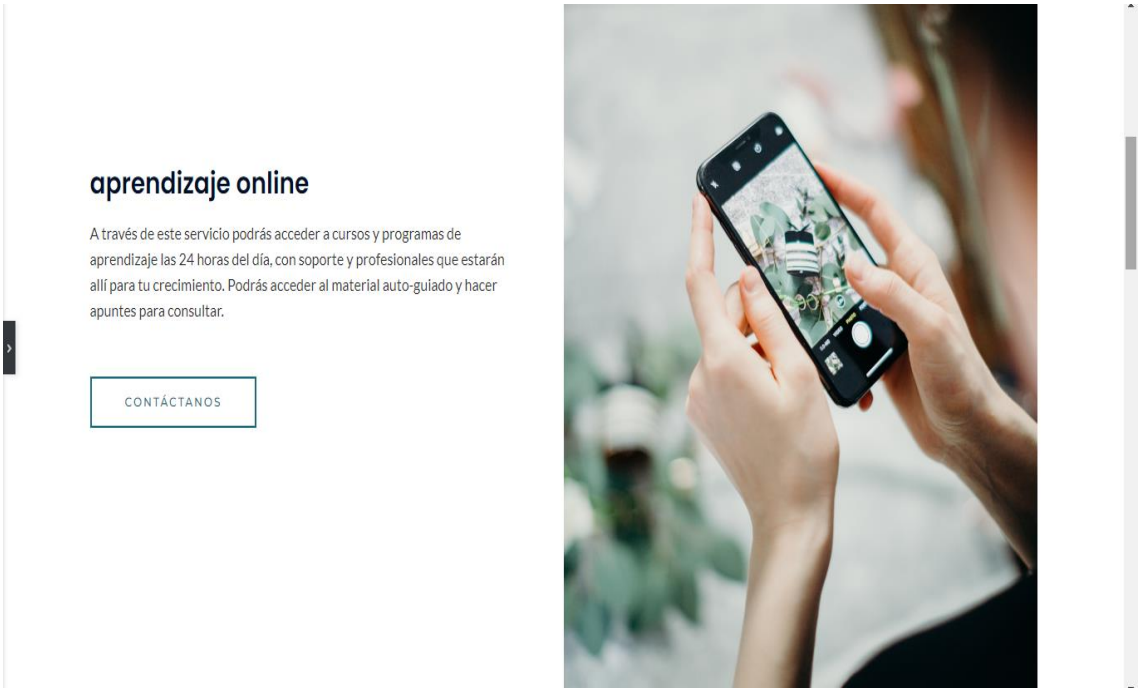

Figura 38: Clases en línea

Figura 39: Seguridad Informática

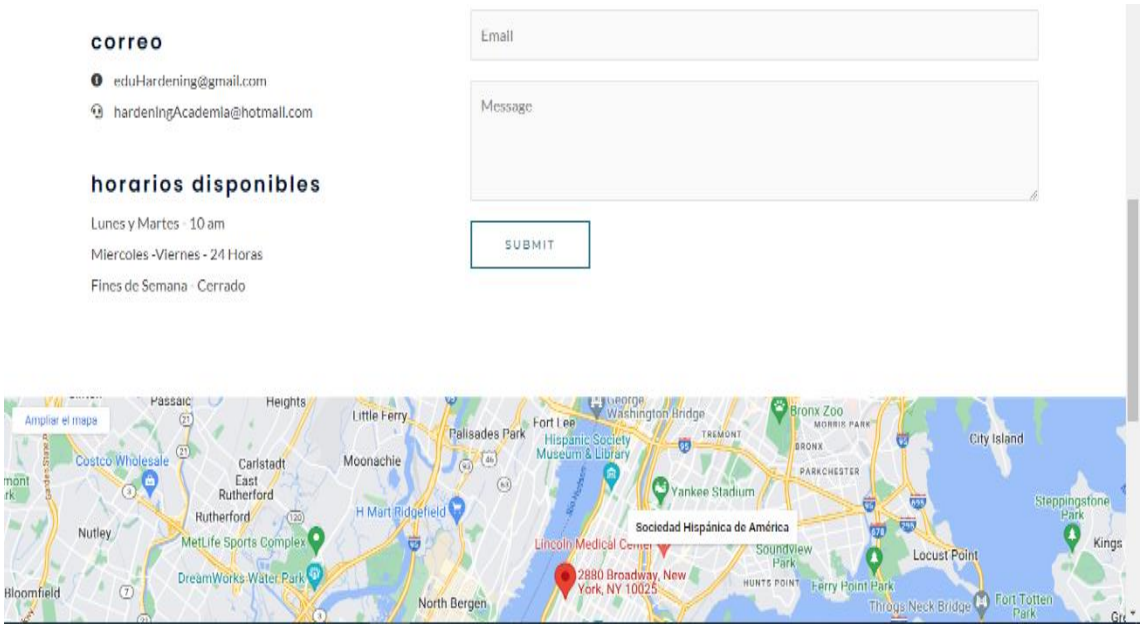

Figura 40: Captura de contactos

5. Proceso de Hardening del servidor web

El *Hardening* consiste en el endurecimiento del sistema, con el fin de reducir y evitar las amenazas y los peligros de este.

5.1. **Identificación y evaluación de las vulnerabilidades**

Realizar un scanner de vulnerabilidades desde el sitio

https://pentest-tools.com/website-vulnerability-scanning/website-scanner

Principales vulnerabilidades encontradas

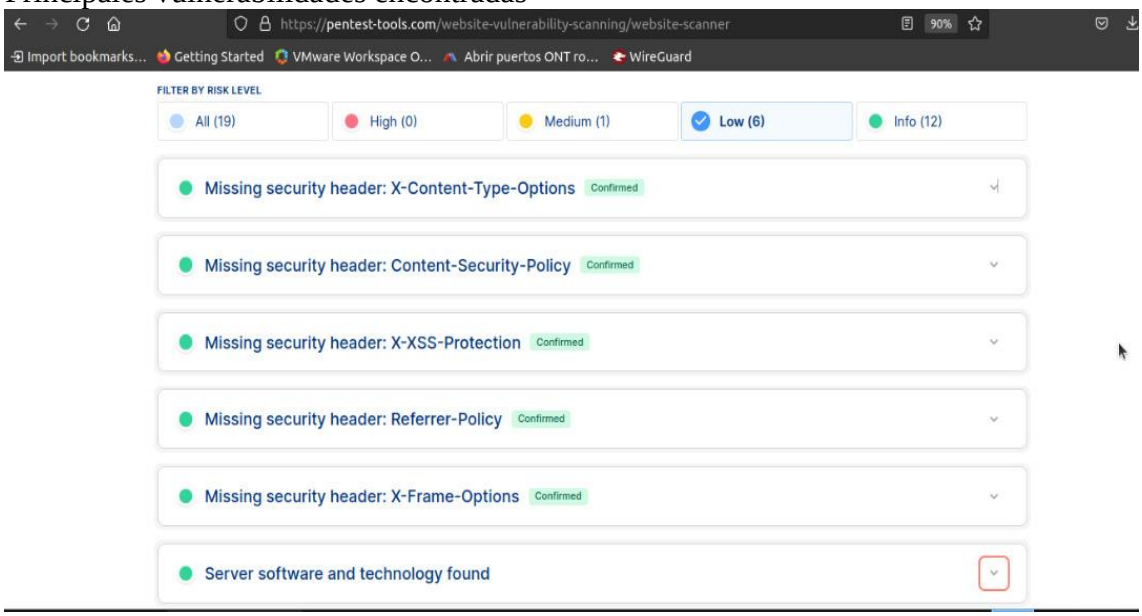

Figura 41: Vulnerabilidades encontradas.

6. Implementación del proceso de Hardening

Proceder a solucionar cada 1 de las vulnerabilidades encontradas.

En el sitio https://cve.mitre.org/ se encuentra información sobre las vulneravilidades encontradas y solucionar las mismas. CVE-2021-21703 CVE-2021-21707 CVE-2021-21706 CVE-2015-9251 CVE-2019-11358 CVE-2020-11022 CVE-2020-11023. Estas se solucionan con la actualización del sistema operativo que está corriendo en el servidor

-Las vulnerabilidades se solucionan agregando configuración al servidor

1 Insecure cookie setting: missing Secure flag

2 Missing security header: Strict-Transport-Security

3 Missing security header: Content-Security-Policy

4 Missing security header: X-Frame-Options

5 Missing security header: X-XSS-Protection

6 Missing security header: X-Content-Type-Options

7 Missing security header: Referrer-Policy

Agregar al fichero /etc/nginx/nginx.conf la siguiente linea.

```
add_header Set-Cookie "Path=/; HttpOnly; Secure";
```

Al comprobar con el comando nuevamente debe estar seguro.

Agregar las líneas en la configuración del virtual host del sitio se soluciona las vulnerabilidades del 2 al 7

```
server {
    add_header       Strict-Transport-Security       "max-age=63072000;
includeSubdomains;" always;
    add_header X-Frame-Options "deny" always;
    add_header X-XSS-Protection "1; mode=block" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header Content-Security-Policy "default-src 'self'" always;
    add_header    Referrer-Policy    "strict-origin-when-cross-origin"
always;
}
```

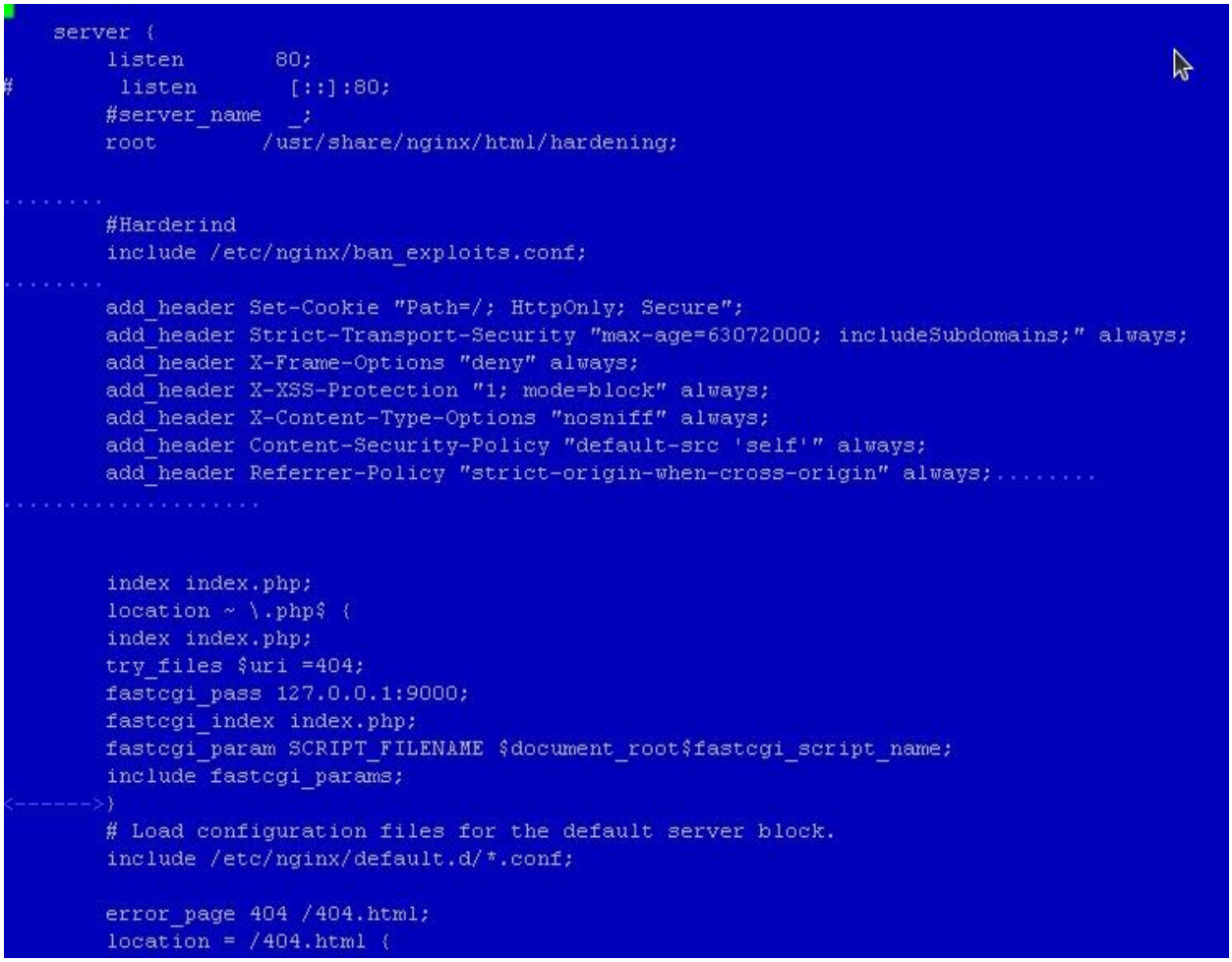

Figura 2: Hardening Implementado

7. Otras configuraciones para mejorar la seguridad

Ingresar al fichero nano /etc/nginx/ban_exploits.conf

El código a continuación permite bloquear los exploits, las inyecciones SQL, inyecciones de archivos, spam y los agentes de usuarios.

- Bloquear Inyecciones SQL.

```
set $block_sql_injections 0;
if ($query_string ~ "union.*select.*\(") {
    set $block_sql_injections 1;
}
if ($query_string ~ "union.*all.*select.*") {
    set $block_sql_injections 1;
}
if ($query_string ~ "concat.*\(") {
    set $block_sql_injections 1;
}
if ($block_sql_injections = 1) {
    return 403;
```

```
}
```

- Bloquear inyecciones de archivo.

```
set $block_file_injections 0;
if ($query_string ~ "[a-zA-Z0-9_]=http://") {
   set $block_file_injections 1;
}
if ($query_string ~ "[a-zA-Z0-9_]=(\.\.//?)+") {
   set $block_file_injections 1;
}
if ($query_string ~ "[a-zA-Z0-9_]=/([a-z0-9_.]//?)+") {
   set $block_file_injections 1;
}
if ($block_file_injections = 1) {
   return 403;
}
```

- Bloquear exploits communes.

```
set $block_common_exploits 0;
if ($query_string ~ "(<|%3C).*script.*(>|%3E)") {
   set $block_common_exploits 1;
}
if ($query_string ~ "GLOBALS(=|\[|\%[0-9A-Z]{0,2})") {
   set $block_common_exploits 1;
}
if ($query_string ~ "_REQUEST(=|\[|\%[0-9A-Z]{0,2})") {
   set $block_common_exploits 1;
}
if ($query_string ~ "proc/self/environ") {
   set $block_common_exploits 1;
}
if ($query_string ~ "mosConfig_[a-zA-Z_]{1,21}(=|\%3D)") {
   set $block_common_exploits 1;
}
if ($query_string ~ "base64_(en|de)code\(.*\)") {
   set $block_common_exploits 1;
}
if ($block_common_exploits = 1) {
   return 403;
}
```

- Bloquear spam

```
set $block_spam 0;
if ($query_string ~ "\b(ultram|unicauca|valium|viagra|vicodin|xanax|ypxaieo)\b") {
   set $block_spam 1;
}
if ($query_string ~ "\b(erections|hoodia|huronriveracres|impotence|levitra|libido)\b")
{
   set $block_spam 1;
}
```

```
    if ($query_string ~ "\b(ambien|blue\spill|cialis|cocaine|ejaculation|erectile)\b") {
       set $block_spam 1;
    }
    if ($query_string ~
"\b(lipitor|phentermin|pro[sz]ac|sandyauer|tramadol|troyhamby)\b") {
       set $block_spam 1;
    }
    if ($block_spam = 1) {
       return 403;
    }
```

- Bloquear agentes de usuarios.

```
    set $block_user_agents 0;
    if ($http_user_agent ~ "Indy Library") {
       set $block_user_agents 1;
    }
    if ($http_user_agent ~ "libwww-perl") {
       set $block_user_agents 1;
    }
    if ($http_user_agent ~ "GetRight") {
       set $block_user_agents 1;
    }
    if ($http_user_agent ~ "GetWeb!") {
       set $block_user_agents 1;
    }
    if ($http_user_agent ~ "Go!Zilla") {
       set $block_user_agents 1;
    }
    if ($http_user_agent ~ "Download Demon") {
       set $block_user_agents 1;
    }
    if ($http_user_agent ~ "Go-Ahead-Got-It") {
       set $block_user_agents 1;
    }
    if ($http_user_agent ~ "TurnitinBot") {
       set $block_user_agents 1;
    }
    if ($http_user_agent ~ "GrabNet") {
       set $block_user_agents 1;
    }

    if ($block_user_agents = 1) {
       return 403;
    }
```