

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACION DE *HARDENING* EN SISTEMAS OPERATIVOS DE SERVIDOR

IMPLEMENTACIÓN DE *HARDENING* EN FEDORA CON CIS

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN REDES Y TELECOMUNICACIONES**

GUANIN RODRIGUEZ STALIN RAMIRO

stalin.guanin@epn.edu.ec

DIRECTOR: GABRIELA KATHERINE CEVALLOS SALAZAR

gabriela.cevalloss@epn.edu.ec

DMQ, septiembre 2024

CERTIFICACIONES

Yo, STALIN RAMIRO GUANIN RODRIGUEZ declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

STALIN RAMIRO GUANIN RODRIGUEZ

stalin.guanin@epn.edu.ec

stalindm_90@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Stalin Ramiro Guanín Rodríguez, bajo mi supervisión.

GABRIELA KATHERINE CEVALLOS SALAZAR
DIRECTOR

gabriela.cevalloss@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

STALIN RAMIRO GUANIN RODRIGUEZ

DEDICATORIA

A mi amada familia,

Quiero expresar mi profundo agradecimiento por el apoyo principal en mi vida y en la realización de esta tesis. Especialmente a mis Amados padres Myrian y Ramiro, que son el pilar de mi existencia. Su unión, dedicación incansable, fortaleza, sabiduría y amor perpetuo han sido la brújula que me ha guiado a lo largo de esta travesía.

Dennys, ha sido una fuente constante de inspiración, su apoyo y confianza inquebrantables me han impulsado a superar obstáculos y alcanzar mis metas.

Pablo, mi fiel compañero, ha estado siempre a mi lado, brindándome apoyo y aliento incansables. Su confianza en mis capacidades ha sido un motor de impulso.

Aracely, ha demostrado una fe inigualable en mis habilidades desde el principio, su amor y apoyo persistentes me han dado la seguridad de avanzar con determinación.

Mercedes, siempre preocupada por mi bienestar, ha sido mi guía y mentora, su deseo de verme prosperar ha sido una luz brillante en mi camino.

Con su amor y apoyo han dejado una huella imborrable en mi corazón y en esta tesis. Cada logro refleja su apoyo y cariño. Agradezco profundamente por ser mi fuente de inspiración, mi fortaleza y mi razón para esforzarme al máximo.

Con sincero cariño y gratitud,

Stalin

AGRADECIMIENTO

Agradezco infinitamente a Dios, cuya gracia y guía constante han iluminado mi camino en este viaje académico. Sin su bendición, nada de esto habría sido posible.

A mi querida familia, por su infinito apoyo, amor y paciencia a lo largo de este arduo proceso. Sus sacrificios y aliento constante han sido mi mayor fuente de fortaleza.

A la Escuela de Formación de Tecnólogos de la Escuela Politécnica Nacional agradezco por brindarme la oportunidad de aprender, crecer y realizar esta investigación. La calidad de la educación que he recibido aquí ha sido fundamental para mi desarrollo académico.

Este logro es un tributo a la gracia de Dios, al apoyo incondicional de mi familia y a la calidad de educación de la Escuela Politécnica Nacional.

Gracias.

ÍNDICE DE CONTENIDOS

1	DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1	Objetivo general	1
1.2	Objetivos específicos.....	1
1.3	Alcance	1
1.4	Marco Teórico	2
	Sistema Operativo	2
	Servidor de correo	3
	Herramientas de escaneo de vulnerabilidades	5
	<i>Hardening</i>	6
	OVAL.....	6
	Marcos de referencia de seguridad.....	7
2	METODOLOGÍA.....	8
3	RESULTADOS	8
3.1	Identificación de las vulnerabilidades en Fedora <i>Server</i> sin políticas de seguridad	9
	Sistema operativo Fedora <i>Server</i> 39.....	9
	Servidor de correo Postfix.....	13
	Instalación de la herramienta de escaneo.....	21
3.2	Implementación de seguridad en un sistema operativo de servidor.....	29
3.3	Análisis de los reportes, resultado de la aplicación de la herramienta de escaneo.	39
3.4	Verificación del <i>Hardening</i> del sistema operativo en base a los elementos de la triada CIA	40
4	CONCLUSIONES.....	43
5	RECOMENDACIONES.....	45
6	REFERENCIAS BIBLIOGRÁFICAS.....	46
7	ANEXOS.....	49
	ANEXO I: Certificado de Originalidad	i
	ANEXO II: Enlaces	ii

RESUMEN

La actual investigación de titulación enfocó en el diseño y fortalecimiento de un entorno seguro, aplicando *hardening* en Fedora Server 39 con los lineamientos de CIS, centrándose en la configuración de un servidor de correo.

La primera sección conlleva la descripción del proyecto y los objetivos a conseguir. Posteriormente se tiene la metodología con los pasos a seguir para obtener el objetivo general.

La segunda sección consistió en el análisis y evidencias de vulnerabilidades. La tercera sección consistió en la instalación de un servidor de correo *Postfix* junto con *Dovecot*, *mailx*, y la herramienta de escaneo *OpenScap* el cual permitió evidenciar un primer reporte de vulnerabilidades. Se basó en el proceso de evaluar y remediar las vulnerabilidades que presenta el sistema operativo cuyas directrices son establecidas en su totalidad por CIS. Luego se estableció el proceso de remediación, fortaleciendo al sistema operativo de servidor Fedora Server 39.

Luego se generaron los reportes mediante la herramienta *OpenScap*, el cual exhibió un control exhaustivo de todas las vulnerabilidades inherentes al sistema, todo este proceso se documentó meticulosamente, proporcionando una visión comprensiva de las medidas correctivas aplicadas en cada etapa del proyecto.

La cuarta sección contiene las conclusiones y recomendaciones generados en el presente proyecto de titulación.

PALABRAS CLAVES: *Hardening, Fedora 39, Postfix, CIS, OpenScap*

ABSTRACT

The current degree research focused on the design and strengthening of a secure environment, applying hardening in Fedora Server 39 with the CIS guidelines, focusing on the configuration of a mail server.

The first section includes the description of the project and the objectives to be achieved. Subsequently, there is the methodology with the steps to follow to obtain the general objective.

The second section consisted of the analysis and evidence of vulnerabilities. The third section consisted of the installation of a Postfix mail server together with Dovecot, mailx, and the Openscap scanning tool, which allowed us to reveal a first report of vulnerabilities. It was based on the process of evaluating and remediating the vulnerabilities presented by the operating system whose guidelines are established in their entirety by CIS. The remediation process was then established, hardening the Fedora Server 39 server operating system.

The reports were then generated using the Openscap tool, which exhibited exhaustive control of all the vulnerabilities inherent to the system. This entire process was meticulously documented, providing a comprehensive view of the corrective measures applied at each stage of the project.

The fourth section contains the conclusions and recommendations generated in this degree project.

KEYWORDS: *Hardening, Fedora 39, Postfix, CIS, OpenSCAP*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El presente proyecto consiste en implementar un proceso de *hardening* en sistemas operativos de servidor basado en un marco de referencia. Con esto se asegura al sistema operativo de servidor, reduciendo significativamente la superficie de ataques, disminuyendo los puntos donde un atacante puede infiltrarse.

Se tiene un sistema operativo de servidor con un servidor de correo electrónico, se escaneará el mismo mediante una herramienta de escaneo de configuración y vulnerabilidades basada en el protocolo SCAP, donde se obtendrá un reporte inicial el cual será comparado con un reporte luego de endurecer al servidor. Este análisis determinará si se ha mejorado la seguridad del servidor.

Se enfoca en los parámetros que estipula en CIS los cuales permiten que todos los reportes obtenidos sean completamente generados en base a la documentación CIS.

1.1 Objetivo general

Implementar *hardening* en sistemas operativos de servidor.

1.2 Objetivos específicos

- Identificar las vulnerabilidades en un sistema operativo de servidor sin políticas de seguridad.
- Implementar seguridad en un sistema operativo de servidor.
- Analizar los reportes, resultado de la aplicación de la herramienta de escaneo.
- Verificar el *hardening* del sistema operativo en base a los elementos de la triada CIA.

1.3 Alcance

El presente proyector consiste en implementar *hardening* en el sistema operativo de servidor Fedora. En primera instancia se investigarán herramientas de

escaneo, de configuración y vulnerabilidades, basados en el protocolo SCAP. Partiendo de esto, se instalará un sistema operativo de servidor con un servidor de correo, sin ninguna política de seguridad; con la herramienta de escaneo se procede a obtener un primer informe de vulnerabilidades. Luego se endurecerá al servidor con el fin de obtener una mejora en el reporte de vulnerabilidades obtenido de la herramienta de escaneo. Se compararán los reportes para observar cuáles parámetros se han solventado según el manual de buenas prácticas de seguridad emitido por organizaciones de estandarización en esta área.

Se realizará una guía que resuma las mejores prácticas, con esto se implementa *hardening* en un servidor, reduciendo la superficie de ataques y por ende mitigando las debilidades que puedan ser aprovechadas por intrusos locales o remotos.

1.4 Marco Teórico

Sistema Operativo

Se lo denomina con las siglas SO, su funcionamiento se basa en la implementación de alcances en procesos y recuperación de información que sea visible al usuario final, en informática un *software* se clasifica de dos formas, los programas internos los cuales ejecutan las operaciones en el computador y programas de aplicación los cuales se enfocan en las actividades finales del usuario, el programa con mayor énfasis es el sistema operativo ya que permite el control de cada recurso de un computador y genera los parámetros que se van a realizar sobre dicha aplicación [1]. El sistema operativo controla totalmente cada programa que sea instalada y permite la interacción de interfaz gráfica entre usuario y el hardware del sistema [2].

Fedora 39

Establecido como sistema operativo de Linux escritorio, totalmente gratuito, permite implementar funciones como servidor de correo. Entre sus funciones a destacar es la navegación en internet, correo electrónico, reproducción multimedia e incluso programación, con Fedora 39 se destaca por ser un sistema operativo robusto en seguridad mediante las actualizaciones frecuentes que

presenta el *software*, cabe mencionar que Fedora es patrocinada por *Red Hat*, adoptando el entorno de escritorio a *GNOME* el cual es la estructura de escritorio enfocado a sistemas operativos *Linux*.

La seguridad en Fedora se basa en *SELinux (Security-Enhanced Linux)* es el cual es el módulo de seguridad que define los controles de acceso para los procesos desarrollados en el *software* [3].

Fedora se posiciona como un sistema operativo de mayor estabilidad, seguridad y acceso amplio de software a través de sus repositorios, su versión establece parámetros visuales de escritorio mejorando el *hardware* permitiendo una mejora continua en el espacio de trabajo para una interfaz gráfica mucho más accesible para el usuario, posee un rendimiento más eficiente ya que el tamaño de partición del sistema es de 500 (MB) coincidiendo con la configuración que maneja Microsoft [4].

Servidor de Fedora

Una computadora de trabajo se utiliza para conectarse a una red local o a Internet con el fin de obtener acceso a documentos o llevar a cabo tareas especializadas. En cambio, un servidor es un programa que responde a los servicios solicitados por un cliente. Ambos tipos de dispositivos informáticos cumplen funciones específicas y permiten el acceso a redes, ya sea local o en línea.

Fedora *Server* se presenta como un sistema operativo robusto y adaptable que incorpora las tecnologías más avanzadas para entornos de centros de datos.

Es aceptable emplear estaciones de trabajo en funciones de servidor, aunque suele reflejarse en la calidad conforme al costo. Además, es esencial verificar que el procesador designado como clase de servidor sea plenamente compatible con la placa base. Por lo general, al optar por fabricantes reconocidos como *Dell*, el *hardware* calificado como de clase de servidor incluye utilidades de gestión, entre otras características [12].

Servidor de correo

El servidor de correo desempeña un rol esencial al facilitar el intercambio de mensajes electrónicos entre usuarios dentro de una red, su función principal

radica en almacenar, recibir y transmitir correos electrónicos utilizando protocolos como *Simple Mail Transfer Protocol* (SMTP) para el envío de correos y *Internet Message Access Protocol* (IMAP) para permitir a los usuarios acceder y gestionar sus mensajes de manera remota [15].

Postfix

Es un programa de escritorio con interfaz gráfica focalizado especialmente en cubrir herramientas como servidor de correo o *Mail Transfer Agent* (MTA), el cual permite receptar y enviar mensajes de mail, conociendo su dominio, buzones de mensajería, se adapta a las versiones IPV4 e IPV6, conexión confiable *Transport Layer Security* (TLS), su configuración no es compleja y posee una robusta seguridad. Su caracteriza por su eficiencia ya que permite manejar varios correos y protocolos a la vez, como el caso del protocolo *SMTP* ayudando a la interoperabilidad.

Una de sus funciones principales es la versatilidad que lo caracteriza ya que permite desempeñarse en diversos entornos como en empresas pequeñas hasta servicios de correo en gran escala [5].

Dovecot

Es un servidor de correo que tiene código abierto, caracterizado por el envío y recepción segura y confiable de mensajes mediante los protocolos *IMAP*, Protocolo de oficina de correos (*POP3*).

Su seguridad alta y solida se basa en *TLS/SSL* es decir en los certificados seguros que permiten navegar de forma cifrada en las páginas web esto realizado en la capa de transporte. Permite realizar pruebas de análisis para el envío de correo electrónico conociendo sus falencias, analizando que parámetros se puede corregir o no de un servidor. Su función de emitir mensajes desde un localhost es muy importante ya que establece una barrera de seguridad para el servidor. Se empareja totalmente con *Postfix* para el endurecimiento y fortalecimiento de un sistema de correo seguro y completo.

Mientras que *Postfix* se maneja como servidor de transferencia de correo (MTA) encargado del enrutamiento eficiente de los correos, *Dovecot* actúa como el servidor de acceso (MDA) en donde el usuario puede gestionar e ingresar a los

correos electrónicos. Su combinación mutua establece una integración total para el acceso y almacenamiento de mensajes por parte del usuario final [6].

Herramientas de escaneo de vulnerabilidades

Las herramientas de escaneo de vulnerabilidades en Fedora Server 39 representan una piedra angular en la identificación y mitigación proactiva de posibles debilidades en la seguridad del sistema, las utilidades desempeñan un papel importante ya que permite examinar la infraestructura informática en busca de posibles vulnerabilidades, configuraciones incorrectas o puntos de acceso no autorizados los cuales pueden ser explotados por atacantes. Los reportes que emiten las herramientas de escaneo proporcionan una visión clara de las vulnerabilidades detectadas, sus niveles de riesgo y recomendaciones para su recomendación.

Openscap

Conocido también como protocolo de automatización de contenido de seguridad, es un conjunto de estándares los cuales permiten automatizar y gestionar políticas de seguridad, en Fedora se emplea *Openscap* para evaluar, escanear, y medir la conformidad con políticas de seguridad predefinidas.

Entre sus funciones se destaca la identificación de vulnerabilidades, el *Benchmarking* (estudio que se encarga de mejorar las buenas prácticas sobre un área determinada) de seguridad de información a través de perfiles estándar, la automatización de cumplimiento, la elaboración de reportes sobre el estado de seguridad ser servidor. Estas funciones son cruciales ya que se puede evidenciar qué solución se puede solventar y bajo qué parámetros.

Es una aplicación que permite escanear la configuración del sistema operativo en Fedora evidenciando que vulnerabilidades tiene en la red de manera local o remota. Con ello se generan informes técnicos proporcionando información clara de todo lo que se debe corregir para la obtención de un sistema totalmente seguro, facilita la administración centralizada de la seguridad en entornos distribuidos y asegurando la conformidad con estándares totalmente reconocidos como el caso del CIS [8].

Hardening

Proceso que permite fortalecer la seguridad de un sistema, aplicando parámetros específicos. Su objetivo se basa en minimizar las áreas vulnerables y mejorar la resistencia frente a amenazas cibernéticas. Involucra acciones como actualizar regularmente el software, desactivar servicios no esenciales, configurar cortafuegos, aplicar políticas de acceso y cifrar información. Se aplica a sistemas operativos, servidores y dispositivos de red, siguiendo el principio de "*Least Privilege*". Las auditorías periódicas evalúan la eficacia de las medidas implementadas, y la automatización y documentación son prácticas clave. La adaptabilidad es esencial para mantener la seguridad frente a cambios en la infraestructura y nuevas amenazas. Es importante para proteger sistemas y datos, fortaleciendo su seguridad a lo largo del tiempo [9].

En Fedora, el proceso de *hardening* implica fortalecer la seguridad del sistema operativo mediante acciones específicas. Esto incluye actualizaciones de las últimas correcciones de vulnerabilidades, configurar un cortafuegos permitiendo controlar el tráfico, aplicar políticas de acceso rigurosas, realizar auditorías periódicas para evaluar la seguridad, implementar configuraciones seguras para protocolos como *TLS/SSL*, asegurar que las aplicaciones sigan prácticas de "*hardening*" y utilizar *SELinux* para aplicar políticas de seguridad a nivel de sistema. Todas estas medidas buscan reducir las vulnerabilidades y fortalecer la resistencia del sistema contra posibles amenazas y ataques cibernéticos [10].

OVAL

Open Vulnerability and Assessment Language corresponde a Lenguaje de Evaluación y Vulnerabilidad Abierto, se establece como un estándar en el ámbito de la ciberseguridad para proporcionar una estructura uniforme en la descripción de información relacionada con vulnerabilidades y evaluaciones de seguridad. Supervisa este estándar. La finalidad principal de *OVAL* es normalizar la representación de datos sobre vulnerabilidades, permitiendo así la automatización de evaluaciones de seguridad y facilitando un intercambio coherente analizando la toda información informática, a su vez busca fomentar la colaboración eficaz y la gestión mejorada de vulnerabilidades y evaluaciones de seguridad [11].

Marcos de referencia de seguridad

Son conjuntos de estándares y mejores prácticas diseñados para brindar un sistema robusto a la información protegiendo contra amenazas cibernéticas, algunos de ellos son:

- NIST (*National Institute of Standards and Technology*): se centra en identificar, proteger, detectar, responder y recuperar información crítica, utilizando un enfoque basado en riesgos.
- CIS: (*Center for Internet Security*) o por sus siglas en español centro para la seguridad en internet, proporciona controles y mejores prácticas priorizados para mitigar riesgos y mejorar la seguridad cibernética, se enfoca a mejorar constantemente la ciberseguridad ofreciendo buenas prácticas de uso, dando iniciativas para un mejor control en software y hardware, las directrices que manejan se denominan *Critical Security Controls* (CSC) las cuales ayudan a las organizaciones a fortalecer las defensas contra los ciberataques y mitigar riesgos, permiten establecer parámetros claves para el cumplimiento de ataques maliciosos, controles orientados a la especificación de vulnerabilidades altas, medias y bajas en un sistema operativo. A su vez está direccionando a la seguridad de la información con el fin de proporcionar un marco integral y sus controles de seguridad ayudan como recursos fundamentales para impulsar la implementación de medidas de seguridad sólidas reduciendo el índice de vulnerabilidades ante amenazas [7].
- PCD DSS (*Payment Card Industry Data Security Standard*): elaborado por el PCI SSC, establece requisitos para la protección de sistema bancarios, concentrándose áreas como la protección del entorno, la gestión de acceso y la monitorización.

Estos marcos son herramientas esenciales para que las organizaciones implementen prácticas de seguridad efectivas y adapten sus defensas cibernéticas según sus necesidades específicas y el entorno operativo, garantizando así la protección de la información sensible [13] [14].

2 METODOLOGÍA

El proyecto actual de titulación está basado en la investigación experimental referente a las vulnerabilidades que presenta el sistema operativo de servidor Linux de Fedora 39.

El primer objetivo consta de la instalación del sistema operativo de servidor Fedora 39, en donde se levantó el servidor de correo *Postfix* sin tomar en cuenta ninguna política de seguridad. Mediante la herramienta de escaneo se obtuvo un primer reporte de vulnerabilidades.

Se ejecutó, sobre el servidor de correo *Postfix*, recomendaciones de seguridad en base a los lineamientos del marco de referencia elegido. Se obtiene un segundo reporte de vulnerabilidades con la herramienta de escaneo.

Se analizaron y compararon los reportes obtenidos, se observaron cuales parámetros se han solventado según el marco de referencia *CIS*.

Se analizó el reporte final, verificando el impacto que tienen las políticas implementadas sobre cada uno de los tres elementos del triángulo de seguridad informática: confiabilidad, integridad y disponibilidad. Además, se realizó una guía que resuma las mejores prácticas a tener en cuenta para implementar un servidor endurecido.

3 RESULTADOS

La implementación actual del proyecto de investigación necesitó el desarrollo continuo de los pasos a seguir enfocado en *hardening* en el sistema operativo Linux base Fedora, aplicándose medidas de protección basadas en la documentación CIS, el cual permitió obtener un servidor con mayor seguridad y robustez. Gracias a la herramienta de escaneo *OpenScap* se pudo solventar las incidencias detalladas en un informe individual.

3.1 Identificación de las vulnerabilidades en Fedora Server sin políticas de seguridad

Sistema operativo Fedora Server 39

Con el fin de evidenciar los reportes al aplicar *hardening* se procede a crear una máquina virtual de servidor en *Oracle VM Virtual Box*, en donde se instaló el servidor de correo y la herramienta de vulnerabilidad. Se debe estipular los siguientes puntos: creación de una máquina virtual dando click en la opción nuevo, donde se establece los siguientes parámetros: nombre de la máquina virtual a crear junto con la imagen ISO de almacenamiento, ver Figura 3.1.

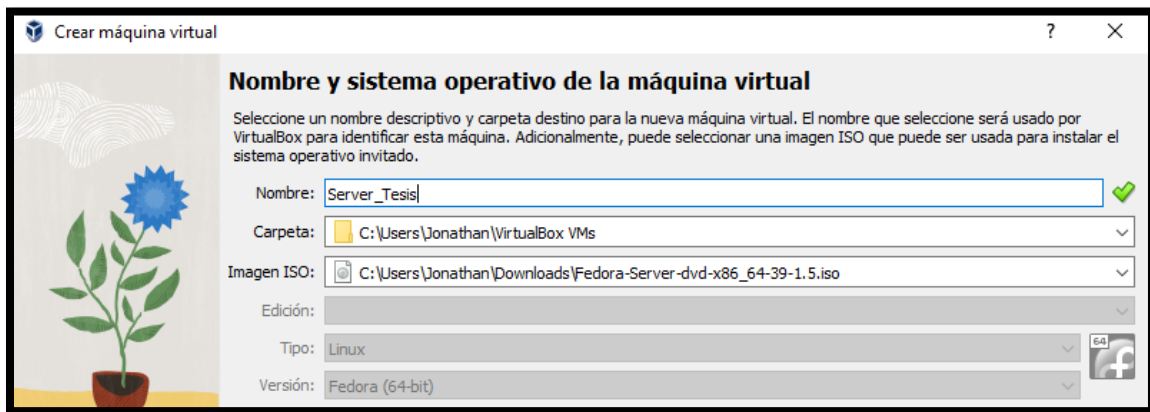


Figura 3.1 Creación de máquina virtual

Posterior a ello se genera una nueva ventana en donde se configura la memoria base de 4096 (MB) que tendrá el sistema operativo Fedora junto con 1 procesador de almacenamiento, ver Figura 3.2.

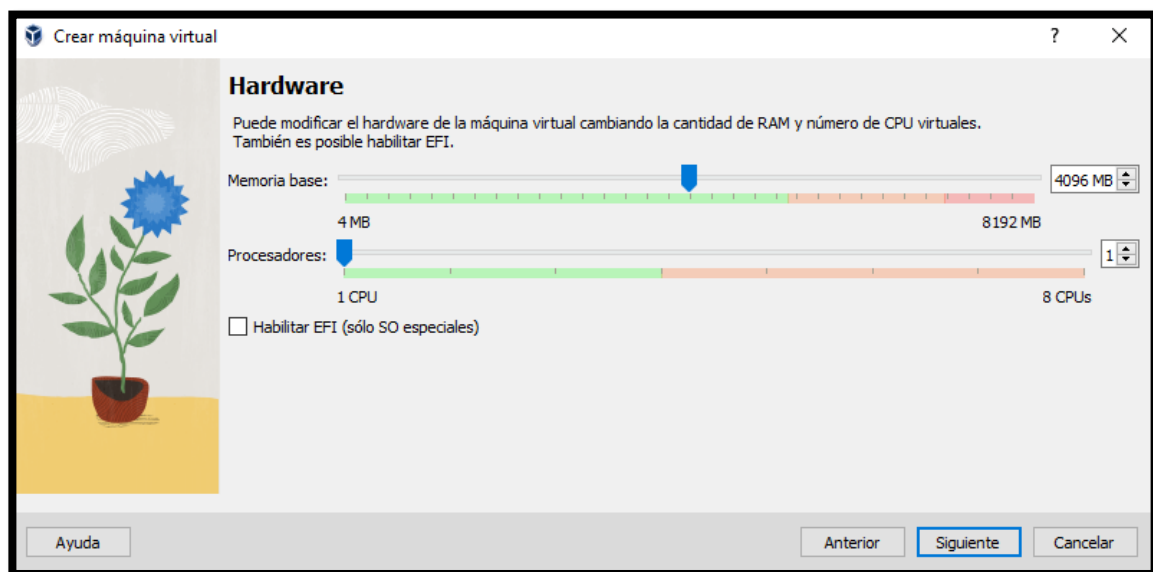


Figura 3.2 Capacidad de procesador

Tal como se establece en la Figura 3.3 la creación de la máquina virtual Fedora se debió otorgar un almacenamiento mínimo de 50 (Gb), para realizar todas las configuraciones el servidor de correo y la herramienta de vulnerabilidad.

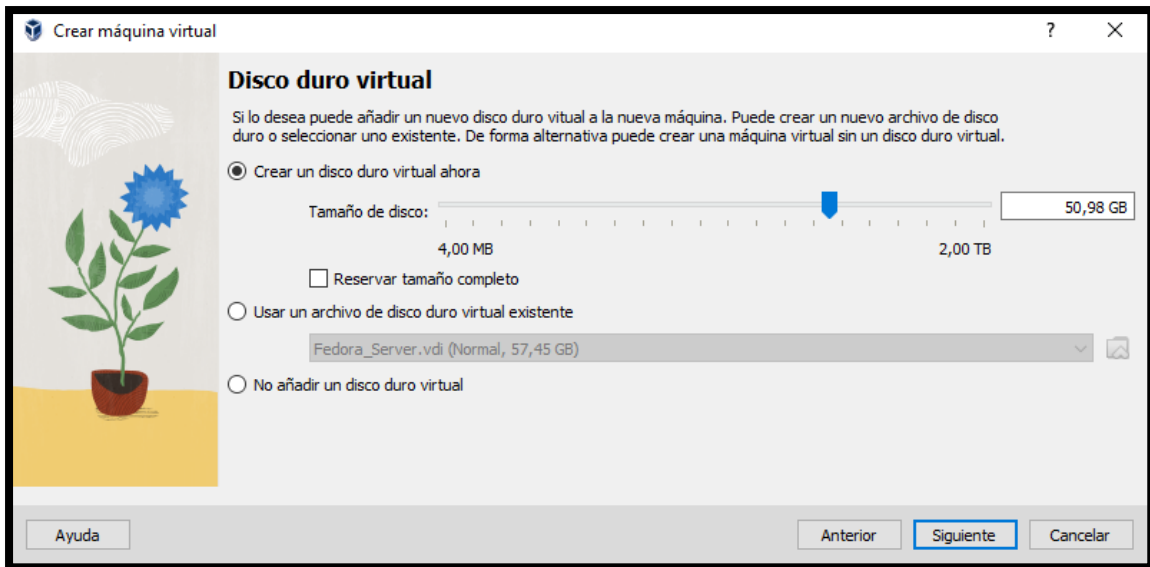


Figura 3.3 Almacenamiento de espacio Máquina Virtual

A continuación, se evidenció como la máquina virtual se la logrado configurar e instalar el sistema operativo Fedora Server 39, con todos lineamientos establecidos los cuales se evidencian en la Figura 3.4.

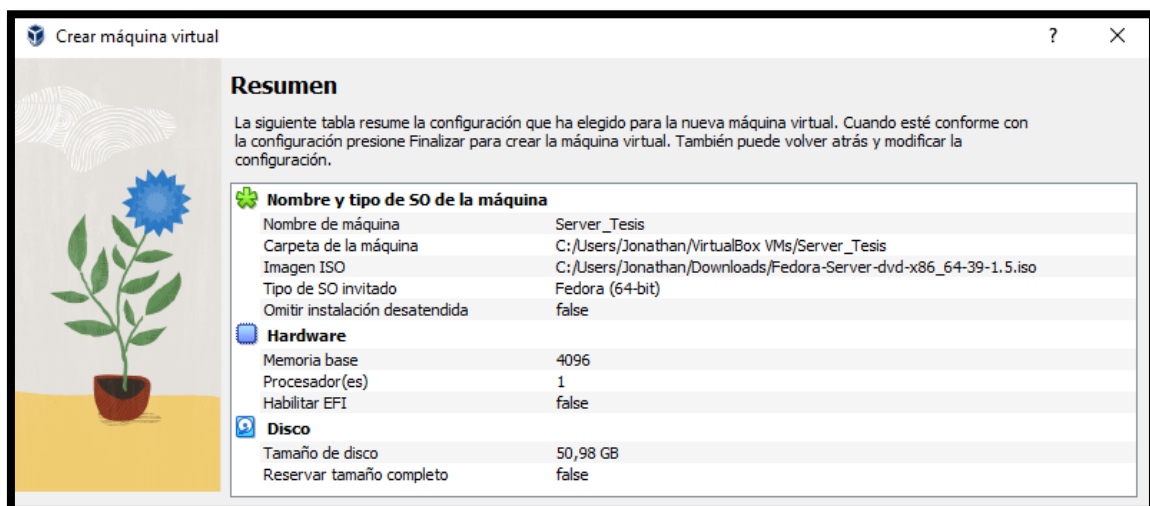


Figura 3.4 Resumen de máquina virtual

Luego se procedió a iniciar la máquina virtual creada, en donde se asignó el idioma de trabajo junto con el País respectivo, para este caso Ecuador, ver Figura 3.5

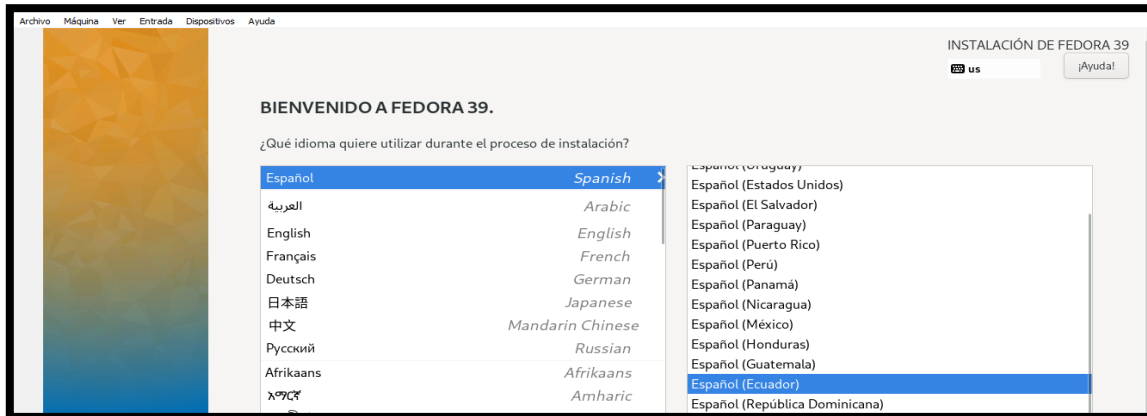


Figura 3.5 Idioma para el Fedora Server

Posterior a ello en la Figura 3.6 se procedió a dar el destino de la instalación seleccionando el disco local en donde se guardará todas las configuraciones establecidas en el sistema operativo.

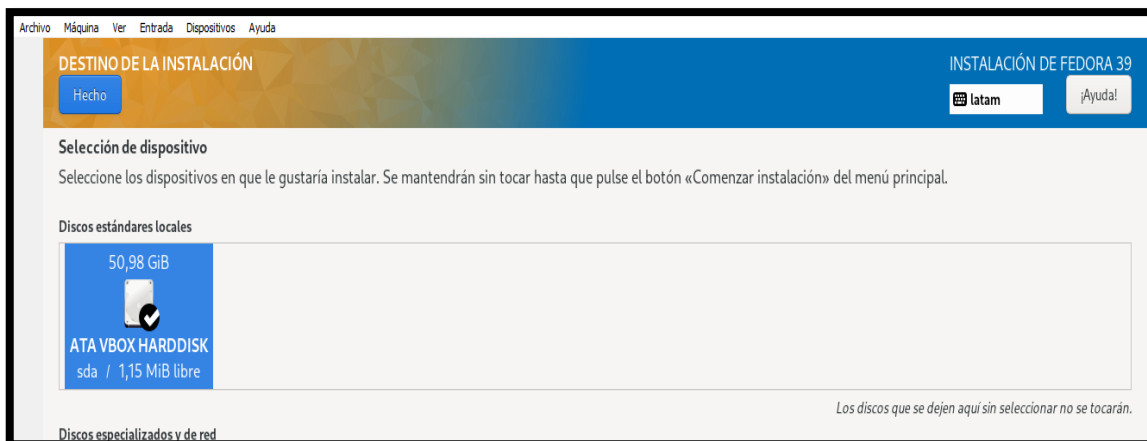


Figura 3.6 Lugar destino del disco local

Acto seguido se activó una cuenta root para respaldo de información ya que cada vez que se quiera ingresar al sistema se ingresa con usuario root y la contraseña asignada. Ver Figura 3.7.

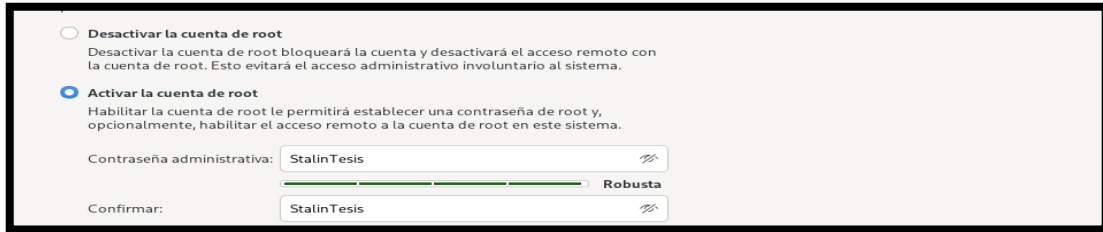


Figura 3.7 Creación de usuario Root

Luego comenzó el proceso de instalación, como se muestra en la Figura 3.8.



Figura 3.8 Proceso inicial de instalación

El programa automáticamente realizó la instalación de Fedora Server 39, esto se evidencia en la Figura 3.9.

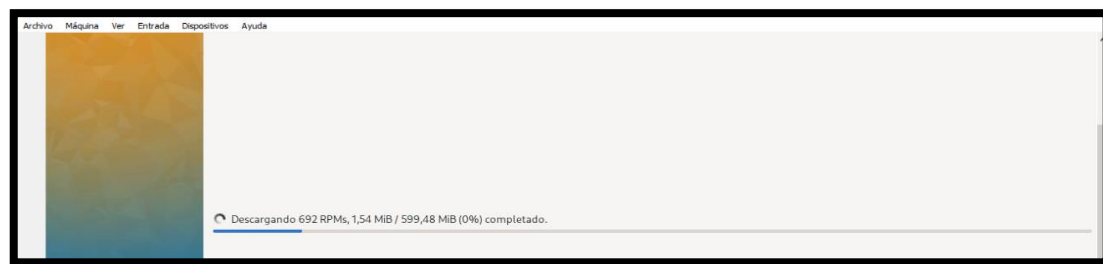


Figura 3.9 Máquina Fedora Server instalada.

Terminada la instalación Fedora Server 39 como se constata en la Figura 3.10 generó el mensaje de bienvenida mediante línea de consola.

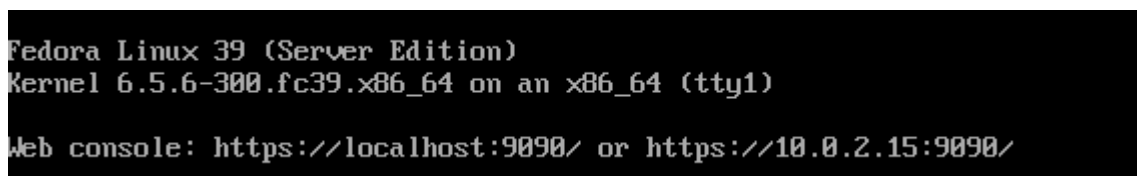


Figura 3.10 Saludo inicial Fedora Server 39

Servidor de correo Postfix

Se procedió a ejecutar el comando `dnf -y install postfix` para instalar el servidor de correo, véase en la Figura 3.11.

```
[root@localhost ~]# dnf -y install postfix
```

Figura 3.11 Ejecución de servidor de correo Postfix

Después se ejecutó el proceso de instalación, verificando que el servidor de correo este activo correctamente, ver Figura 3.12.

```
Dependencias resueltas.
=====
Paquete                Arquitectura      Versión           Repositorio      Tam.
-----
Instalando:
postfix                x86_64           2:3.0.4-1.fc39   updates          1.5 M
```

Figura 3.12 Servidor de correo Postfix activo

De forma inmediata mediante el comando `vim etc/postfix/main.cf` como se evidencia en la Figura 3.13 se ingresa al `main.cf`, en el cual se configuraron todos los parámetros para que el servidor de correo funcione correctamente.

```
# Global Postfix configuration file. This file lists only a subset
# of all parameters. For the syntax, and for a complete parameter
# list, see the postconf(5) manual page (command: "man 5 postconf").
#
```

Figura 3.13. Archivo principal de Postfix

Después se estableció el *hostname*, es este caso es `stalintesis.com`, obsérvese en la Figura 3.14.

```
myhostname = stalintesis.com
```

Figura 3.14. Nombre del host en Postfix

Se definió el nombre de dominio para el envío y recepción de los correos, ver Figura 3.15.

```
mydomain = stalintesis.com
```

Figura 3.15 Nombre del dominio en Postfix

Se descomenta la línea `myorigin` como se establece en la Figura 3.16, el cual constituye al origen de los correos donde se indica que se establecerá como dominio a `stalintesis.com`.

```
myorigin = $mydomain
```

Figura 3.16 Origen de dominio

A continuación, en la siguiente línea de comando de la Figura 3.17 se añade \$mydomain en donde se especifican las direcciones locales a las que el servidor de correo entregará los mails, donde incluye el nombre del servidor, direcciones locales, y el dominio establecido en el servidor.

```
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
```

Figura 3.17 Especificación de direcciones locales

Después en la Figura 3.18 se establecieron las redes, de las cuales Postfix recibirá las conexiones.

```
mynetworks = 127.0.0.0/8, 10.0.0.0/24
```

Figura 3.18 Red pública y privada

Se desenmarcó la línea home_mailbox, que indica que los buzones de correo de los usuarios estarán configurados en el formato Maildir en los directorios home. Esto se observa en la Figura 3.19.

```
home_mailbox = Maildir/
```

Figura 3.19 Especificación de los buzones de correo

Luego se estableció el banner que Postfix mostrará al recibir las conexiones SMTP, ver Figura 3.20.

```
smtpd_banner = $myhostname ESMTIP
```

Figura 3.20. Estructura de Banner

Finalmente, en el servidor de correo se establece las siguientes líneas de comandos al final del archivo main.cf. Los cambios se visualizan en la Figura 3.21.

- Se deshabilitó el comando vrfy, permitiendo verificar la existencia de una dirección de correo.
- Se configuró Postfix para aceptar que los clientes SMTP proporcionen un saludo HELO, al establecer la conexión, permitiendo disminuir la cantidad de correos no deseados al verificar la autenticidad de los clientes.

- Se estableció un tamaño límite para los mensajes entrantes es decir en 10 megabytes.
- Se mencionó que *postfix* utilizara *dovecot* como el tipo de autenticación *Simple Authenticated and Security Layer* (SASL) para mejorar la autenticación de los clientes SMTP.
- Se implantó la ubicación del socket entre *Postfix* y *dovecot* para la autenticación SSL, se utiliza el directorio *private/auth*.
- Se habilitó la autenticación SASL, permitiendo a los usuarios autenticarse antes de realizar un envío de mail.
- Se especificó que se rechazara conexiones de clientes SMTP que intenten autenticarse de manera anónima.
- Se creó el dominio local para la autenticación SASL, el dominio se configura con el nombre del host.
- Se definió las restricciones para aceptar o rechazar destinatarios de correos electrónicos, esto ayuda a controlar quien puede enviar correos electrónicos a través del servidor postfix.

```

disable_vrf_command = yes
smtpd_helo_required = yes
message_size_limit = 1048576
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $myhostname
smtpd_recipient_restrictions = permit_mynetworks, permit_auth_destination, permit_sasl_authenticated, reject

```

Figura 3.21 Configuración final del servidor de correo

Se guardó el archivo *main.cf* con todas las configuraciones mencionadas y realizando click en el botón Esc junto con *:wq* se regresa al usuario root del sistema. Figura 3.22.

```

:wq

```

Figura 3.22 Guardar archivo.cf

A continuación, se procedió a reiniciar el sistema Postfix, haciendo que todos los cambios que se realizaron sean guardados de forma correcta, ver Figura 3.23

```

[root@localhost ~]# systemctl enable --now postfix
Created symlink /etc/systemd/system/multi-user.target.wants/postfix.service → /usr/lib/systemd/system/postfix.service.

```

Figura 3.23 Servidor de correo Postfix reiniciado.

Luego, mediante el comando de la Figura 3.24, se agregó la regla de servicio para SMTP al *firewall*, permitiendo el tráfico a través del puerto SMTP.

```
[root@localhost ~]# firewall-cmd --add-service=smtp
success
```

Figura 3.24 Regla al *Firewall*

Después se aplicaron los cambios realizados en tiempo de ejecución al archivo de configuración permanente del *firewall*, como se establece en la Figura 3.25.

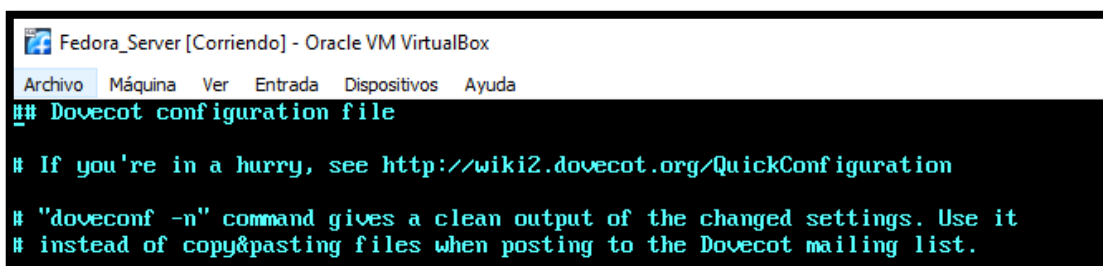
```
[root@localhost ~]# firewall-cmd --runtime-to-permanent
success
```

Figura 3.25 Configuración en el *firewall*

Posteriormente se procedió a instalar Dovecot como se muestra en la Figura 3.26. Luego se ingresó al archivo *dovecot.conf* para configurar los lineamientos que se requieren para su funcionamiento, el cual se puede observar en la Figura 3.27.

```
[root@localhost ~]# dnf -y install dovecot
```

Figura 3.26 Instalación Dovecot



```
Fedora_Server [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
## Dovecot configuration file
# If you're in a hurry, see http://wiki2.dovecot.org/QuickConfiguration
# "doveconf -n" command gives a clean output of the changed settings. Use it
# instead of copy&pasting files when posting to the Dovecot mailing list.
```

Figura 3.27 Configuración Dovecot

Luego se descomentó la siguiente línea de comando ya que Dovecot mostrará conexiones en todas las direcciones IPv4 e IPv6, esto se muestra en la Figura 3.28.

```
listen = *, ::
```

Figura 3.28 Direccionamiento IP en Dovecot

Posteriormente se ingresó al archivo `10-auth.conf` para la autenticación de Dovecot en donde de igual forma se descomentaron dos líneas de comando, ver Figura 3.29.

```
# Disable LOGIN command and all other plaintext authentications unless
# SSL/TLS is used (LOGINDISABLED capability). Note that if the remote IP
# matches the local IP (ie. you're connecting from the same computer), the
# connection is considered secure and plaintext authentication is allowed.
# See also ssl=required setting.
#disable_plaintext_auth = yes
```

Figura 3.29 Archivo *Auth* Dovecot.

En la Figura 3.30 se indicó que Dovecot permitirá la transmisión de contraseña en texto claro y en la Figura 3.31 se especifica el método de autenticación para la transmisión de credenciales de usuario.

```
disable_plaintext_auth = no
```

Figura 3.30 Autenticación en Dovecot

```
auth_mechanisms = plain login
```

Figura 3.31 Método de autenticación en Dovecot.

Posterior a ello se ingresó al archivo *mail*, el cual consiste en la configuración específica para el almacenamiento de correo en Dovecot; se evidencia en la Figura 3.32 en donde puede realizar los ajustes necesarios relacionado con el almacenamiento de correo.

```
# If you're using mbox, giving a path to the INBOX file (eg. /var/mail/%u)
# isn't enough. You'll also need to tell Dovecot where the other mailboxes are
# kept. This is called the "root mail directory", and it must be the first
# path given in the mail_location setting.
```

Figura 3.32 Archivo *mail* en Dovecot

Como se muestra en la Figura 3.33. se descomenta la línea `mail_location` para utilizar el formato Maildir y se especifica la ubicación del directorio de correo de los usuarios creados.

```
mail_location = maildir:~/Maildir
```

Figura 3.33 Formato para directorio Maildir en Dovecot

A continuación, se abre el archivo SSL el cual maneja la configuración para ajustar configuraciones SSL/TLS de Dovecot, ver Figura 3.34.

```
## SSL settings
##
# SSL/TLS support: yes, no, required. <doc/wiki/SSL.txt>
# disable plain pop3 and imap, allowed are only pop3+TLS, pop3s, imap+TLS and imaps
# plain imap and pop3 are still allowed for local connections
ssl = required
```

Figura 3.34 Archivo SSL Dovecot

En la Figura 3.35, se desenmarcó la línea SSL, especificando la habilitación del uso de *Secure Sockets Layer* en Dovecot SSL, es decir las conexiones entre cliente - servidor, teniendo un canal cifrado para el aseguramiento de información de datos.

```
ssl = yes
```

Figura 3.35 Habilitación de SSL en Dovecot

A continuación, mediante la Figura 3.36 se habilitó y activó el servicio Dovecot.

```
[root@localhost ~]# systemctl enable --now dovecot
Created symlink /etc/systemd/system/multi-user.target.wants/dovecot.service → /usr/lib/systemd/system/dovecot.service.
```

Figura 3.36 Activación Dovecot

Como se evidencia en la Figura 3.37 se añadieron las reglas para POP3, puerto 110 e IMAP, puerto 143 al *firewall*, esto para poder permitir el tráfico para los protocolos de acceso a los correos electrónicos.

```
[root@localhost ~]# firewall-cmd --add-service={pop3,imap}
success
```

Figura 3.37 Acceso a puertos 110 y 143

La siguiente línea de comando realizó los cambios respectivos al *firewall*, es decir que las modificaciones se mantengan después de reiniciar el sistema, ver Figura 3.328.

```
[root@localhost ~]# firewall-cmd --runtime-to-permanent
success
```

Figura 3.38 Cambios al firewall en Dovecot

Después se procedió a instalar *Mailx*, para realizar las pruebas de correos electrónicos directamente desde la línea de comandos, ver Figura 3.39.

```
[root@localhost ~]# dnf -y install mailx
```

Figura 3.39 Instalación *mailx*

Como se evidenció la Figura 3.40 se genera la variable para el entorno MAIL con la ubicación del directorio *Maildir*.

```
[root@localhost ~]# echo 'export MAIL=$HOME/Maildir' >> /etc/profile.d/mail.sh
```

Figura 3.40 Directorio *Maildir*

Se continuó a generar usuarios en el servidor de correo los cuales son los siguientes: Ramiro, Myrian, para las pruebas de correos con las contraseñas: 1234. Se evidencia en las figuras Figura 3.41 y Figura 3.42.

```
[root@localhost ~]# useradd ramiro
[root@localhost ~]# passwd ramiro
Cambiando la contraseña del usuario ramiro.
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres
Vuelva a escribir la nueva contraseña:
passwd: todos los tokens de autenticación se actualizaron exitosamente.
```

Figura 3.41 Usuario Ramiro

```
[root@localhost ~]# passwd myrian
Cambiando la contraseña del usuario myrian.
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres
Vuelva a escribir la nueva contraseña:
passwd: todos los tokens de autenticación se actualizaron exitosamente.
```

Figura 3.42 Usuario Myriam

Luego se realizaron las pruebas de mails correspondientes, es decir se envían mensajes de pruebas para establecer la recepción de cada mail. Ver Figura 3.43 y Figura 3.44.

```
[root@localhost ~]# mail ramiro@stalintesis.com
Subject: Invitacion de Graduacion
Se comunica que la invitacion esta pendiente para su incorporacion.
Saludos C.
Buen dia.
```

Figura 3.43 Envío de correo

```
[root@localhost ~]# mail myrian@stalintesis.com
Subject: Prueba de correo 2.0
El servidor de correo se levanto con exito.
Saludos C.
Buen dia.
```

Figura 3.44 Prueba de mensaje

A continuación, para evidenciar que los mensajes llegaron con normalidad se estableció el siguiente comando, véase en la Figura 3.45.

```
[root@localhost ~]# cd /home/ramiro/Maildir/new/
```

Figura 3.45 Buzón usuario

Luego mediante el comando `ls -la` se va a desplegar la siguiente opción en donde se copió el ID del mensaje `local-host-live` para evidenciar que los mensajes se establecieron con total normalidad, ver Figura 3.46.

```
[root@localhost new]# ls -la
total 4
drwx-----. 2 ramiro ramiro 67 feb  8 01:51 .
drwx-----. 5 ramiro ramiro 39 feb  8 01:51 ..
-rw-----. 1 ramiro ramiro 640 feb  8 01:51 1707375096.Ufd00I18fd533M536840.localhost.localdomain
```

Figura 3.46 ID de mensaje

Para ejecutar el mensaje se copió el ID anteponiendo el comando `vim`, como se evidencia en la Figura 3.47.

```
[root@localhost new]# vim 1707375208.Ufd00I18fd535M766723.localhost.localdomain
```

Figura 3.47 Mensaje del usuario

Automáticamente el mensaje se lo visualiza de la siguiente manera, ver Figura 3.48 y Figura 3.49.

```
Return-Path: <root@stalintesis.com>
X-Original-To: myrian@stalintesis.com
Delivered-To: myrian@stalintesis.com
Received: by stalintesis.com (Postfix, from userid 0)
        id B2A9010A3C8E; Thu,  8 Feb 2024 01:53:28 -0500 (-05)
Date: Thu, 08 Feb 2024 01:53:28 -0500
To: myrian@stalintesis.com
Subject: Prueba de correo 2.0
User-Agent: Heirloom mailx 12.5 7/5/10
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Message-Id: <20240208065328.B2A9010A3C8E@stalintesis.com>
From: Super User <root@stalintesis.com>

El servidor de correo se levanto con exito.
Saludos C.
Buen dia.
~
```

Figura 3.48 Visualización mensaje enviado

```
Return-Path: <root@stalintesis.com>
X-Original-To: ramiro@stalintesis.com
Delivered-To: ramiro@stalintesis.com
Received: by stalintesis.com (Postfix, from userid 0)
        id 78C5F10A3C8E; Thu,  8 Feb 2024 01:51:36 -0500 (-05)
Date: Thu, 08 Feb 2024 01:51:36 -0500
To: ramiro@stalintesis.com
Subject: Invitacion de Graduacion
User-Agent: Heirloom mailx 12.5 7/5/10
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Message-Id: <20240208065136.78C5F10A3C8E@stalintesis.com>
From: Super User <root@stalintesis.com>

Se comunica que la invitacion esta pendiente para su incorporacion.
Saludos C.
Buen dia.
~
```

Figura 3.49 Mensaje enviado con éxito

Instalación de la herramienta de escaneo

Se procede a instalar la herramienta de vulnerabilidad OpenScap, ver la Figura 3.50.

```
[root@localhost ~]# dnf -y install openscap-scanner
```

Figura 3.50. Instalación de OpenScap

Se instala el siguiente paquete de forma automática el cual contiene perfiles de seguridad de SCAP. Esto se observa en la Figura 3.51.

```
[root@localhost ~]# dnf -y install scap-security-guide
```

Figura 3.51 Instalación de paquete Scap security guide

Se ingresó directamente a la página oficial OVAL, como se observa en la Figura 3.52, en donde detalla cómo tiene la variedad de productos los cuales hace énfasis al entorno de CIS, véase Figura 3.53.

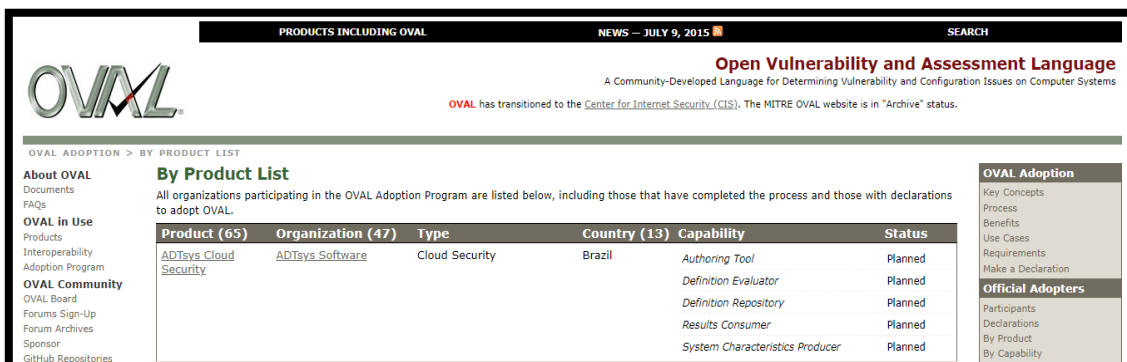


Figura 3.52 Interfaz de Oval

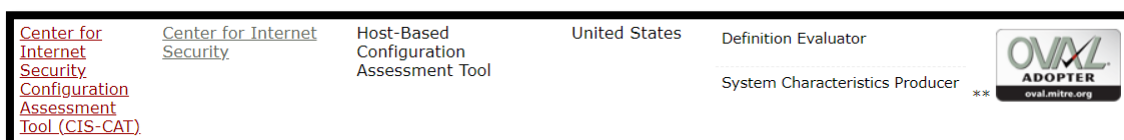


Figura 3.53 Interfaz CIS

Luego en la página web de OpenScap, Figura 3.54, se muestran los perfiles que brinda para poder implementar las buenas prácticas de seguridad en base al perfil elegido. Se puede constatar en la Figura 3.55.

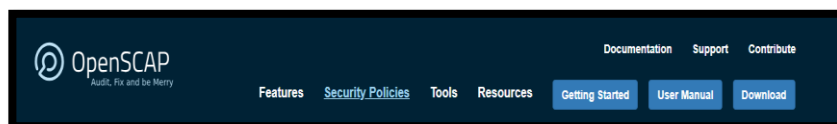


Figura 3.54 Página Oficial de OpenScap

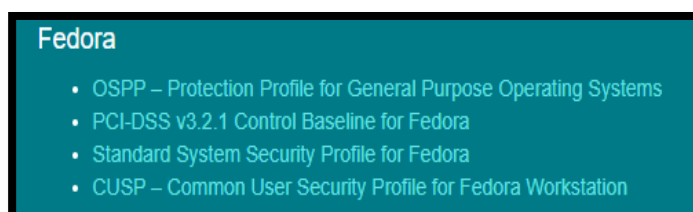


Figura 3.55 Perfiles para establecidos para realizar pruebas de vulnerabilidades

Se procedió a elegir el perfil CUSP – *Common User Security Profile for Fedora Workstation*, debido a que se alinea según la guía de CIS, ver Figura 3.56.



Figura 3.56 Perfil CUSP

En la Figura 3.57 se pudo ejecutar el siguiente comando para evaluar los perfiles de OpenScap.

```
[root@localhost ~]# oscap info /usr/share/xml/scap/ssg/content/ssg-fedora-ds.xml
Document type: Source Data Stream
Imported: 2023-12-18T19:00:00

Stream: scap_org.open-scap_datastream_from_xccdf_ssg-fedora-xccdf.xml
Generated: (null)
Version: 1.3
Checklists:
  Ref-Id: scap_org.open-scap_cref_ssg-fedora-xccdf.xml
  Status: draft
  Generated: 2023-12-20
  Resolved: true
  Profiles:
    Title: CUSP - Common User Security Profile for Fedora Workstation
    Id: xccdf_org.ssgproject.content_profile_cusp_fedora
    Title: OSPP - Protection Profile for General Purpose Operating Systems
    Id: xccdf_org.ssgproject.content_profile_ospp
    Title: PCI-DSS v3.2.1 Control Baseline for Fedora
    Id: xccdf_org.ssgproject.content_profile_pci-dss
    Title: Standard System Security Profile for Fedora
    Id: xccdf_org.ssgproject.content_profile_standard
  Referenced check files:
    ssg-fedora-oval.xml
    system: http://oval.mitre.org/XMLSchema/oval-definitions-5
    ssg-fedora-ocil.xml
    system: http://scap.nist.gov/schema/ocil/2
```

Figura 3.57 Perfil CUSP desde la terminal de comandos

A continuación, se evidencia la razón por la cual no se establece los 3 perfiles restantes en Fedora.

El primer perfil establecido que detalla la guía de OpenScap es OSPP – *Profile for General Purpose Operative Systems*, véase la Figura 3.58.

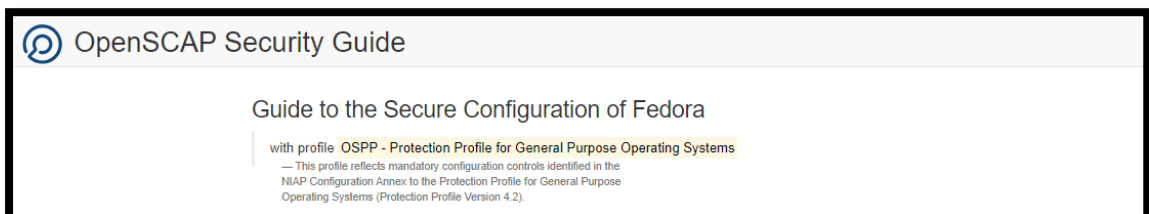


Figura 3.58 Perfil OSPP en OpenScap

A continuación, en la Figura 3.59, se detalla la documentación que tiene el perfil OSPP ya que se establece bajo los controles obligatorios de configuración establecidos en la configuración NIAP el cual evalúa y certifica productos para cumplir con los estándares preestablecidos, véase el Anexo II.

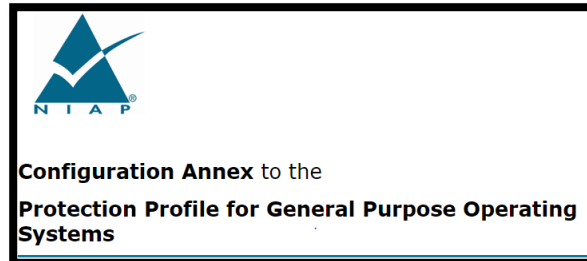


Figura 3.59 Configuración NIAP

El segundo perfil que se establece en la guía de OpenScap es el PCI-DSS que se menciona en la Figura 3.60.

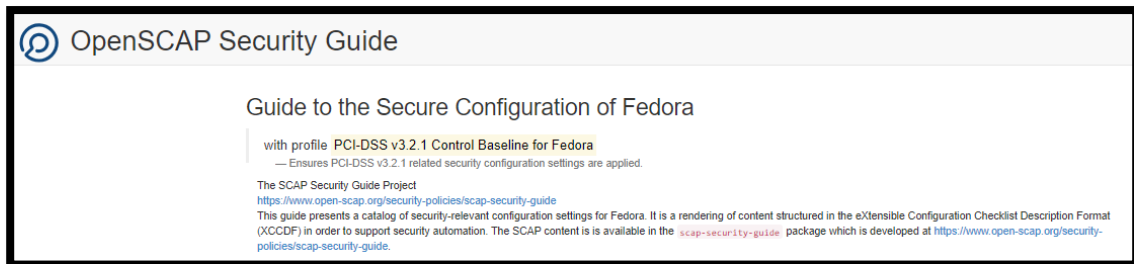


Figura 3.60 Perfil PCI-DSS

A continuación, se evidenció en la Figura 3.61. el detalle específico de los lineamientos del perfil en donde se destaca que el perfil se basa en las vulnerabilidades de tarjetas de pago para entidades bancarias, véase el Anexo 2.



Figura 3.61 Perfil PCI DSS

El tercer perfil que se menciona en la guía de OpenScap es el *Standard System Security Profile for Fedora* el cual detalla las características en la Figura 3.62.

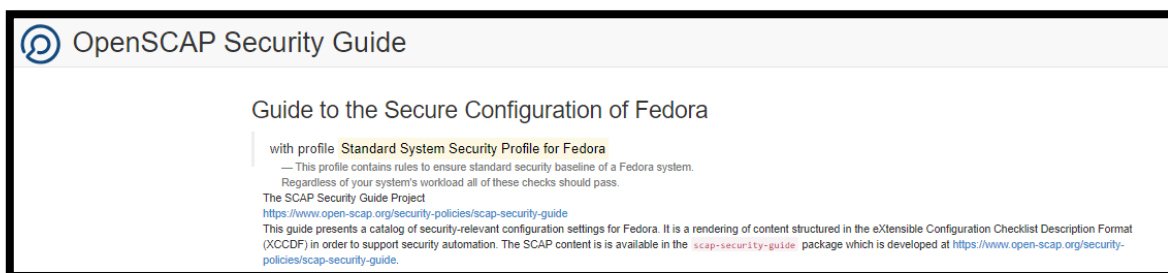


Figura 3.62 Perfil Estándar Fedora

Como se muestra en la Figura 3.63, Anexo 2, se evidencia la descripción que conlleva para la configuración y descripción del perfil, el cual sigue los lineamientos y estándares diferentes a los lineamientos de CIS.

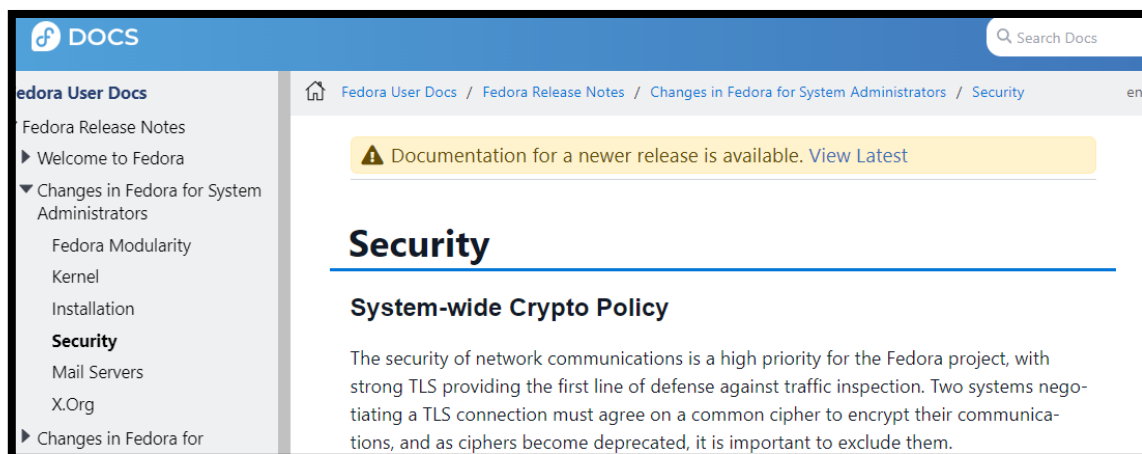


Figura 3.63 Perfil estándar de Fedora

Los 3 perfiles no cumplen con los lineamientos de CIS porque están basados en el concepto de los otros productos establecidos en OVAL ADOPTER.

Oval es una parte integral de OpenScap, utiliza para describir vulnerabilidades y configuraciones seguras, es decir inserta productos de seguridad a cada uno de sus perfiles, pero no todos tienen las mismas características, para el caso de estudio el perfil CUSP si está relacionado con los lineamientos de CIS, los tres perfiles restantes están enfocados de manera genérica, y relacionados con otros productos que se mencionan en la página del OpenScap

Al utilizar el perfil CUSP el cual le pertenece a la herramienta de OpenScap se lo ejecuta en Fedora directamente.

Primer reporte de vulnerabilidades

A continuación, se realizó la ejecución del primer reporte de vulnerabilidades sin realizar ninguna medida de vulnerabilidad, mediante el comando de la Figura 3.64.

```
[root@localhost ~]# oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_cusp_fedora --results reporte1.xml /usr/share/xml/scap/ssg/content/ssg-fedora-ds.xml
```

Figura 3.64 Reporte Inicial de OpenScap sin ninguna medida de vulnerabilidad

A continuación, mediante el comando de la Figura 3.65, genera un informe HTML a partir de los resultados obtenidos con las directrices de CIS.

```
[root@localhost ~]# oscap xccdf generate report reporte1.xml > reporte1.html
```

Figura 3.65 Generación de archivo html reporte inicial

Luego para poder evidenciar el reporte en formato html se procede a utilizar los programas MINGW64, *Visual Studio Code*, *FileZilla*. Previa a esto, se evidencia que la conexión de red de la máquina virtual esté conectada en adaptador puente para que se enlace la máquina virtual con el entorno de escritorio, véase la Figura. 3.66.

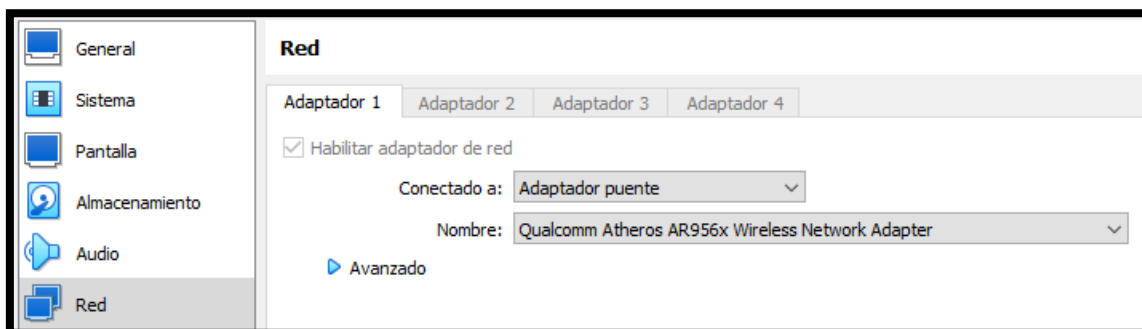


Figura 3.66 Conexión de red máquina virtual

Posterior a ello se evidencia que la IP 192.168.100.84/24 del servidor se genera para lograr la conexión remota mediante *FileZilla*, véase en la Figura 3.67.

```
root@localhost ~]# ip a
: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
  inet6 ::1/128 scope host noprefixroute
    valid_lft forever preferred_lft forever
: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
  link/ether 08:00:27:62:22:43 brd ff:ff:ff:ff:ff:ff
  inet 192.168.100.84/24 brd 192.168.100.255 scope global dynamic noprefixroute enp0s3
```

Figura 3.67. IP del servidor de correo

Luego se ingresó al programa *FileZilla* para lograr transmitir los archivos de información en donde se configura la IP del servidor, el protocolo que se va a utilizar, el puerto, el usuario y la contraseña generada para Fedora Server 39 y se pudo observar los reportes de vulnerabilidad generados, véase en la Figura 3.68.

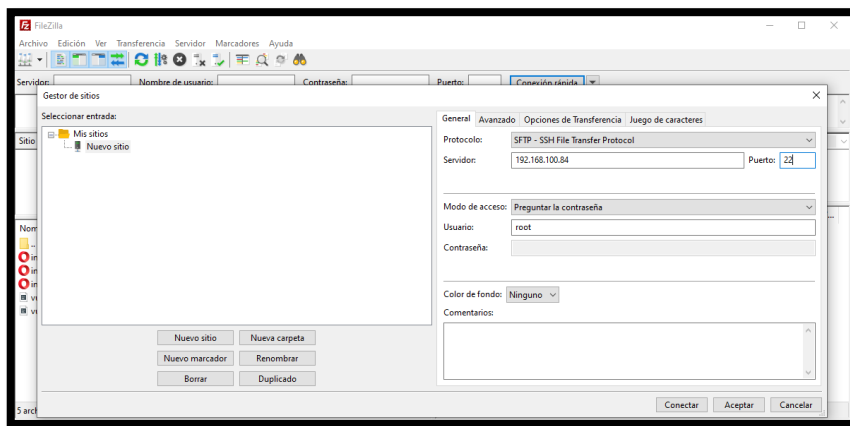


Figura 3.68 Configuración del servidor de correo para la visualización de vulnerabilidades

Después se generó una pestaña en donde se debe dar click en la opción confiar en el sitio, para lograr acceder a la carpeta, como se menciona en la Figura 3.69.

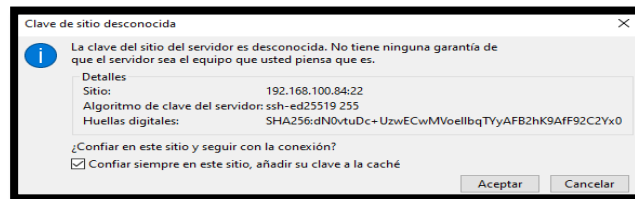


Figura 3.69 Aprobación para conectar Filezilla

Automáticamente se evidenció como se ingresa al usuario root y se observa los archivos que se pueden enviar a la carpeta de escritorio, como se establece en la Figura 3.70.

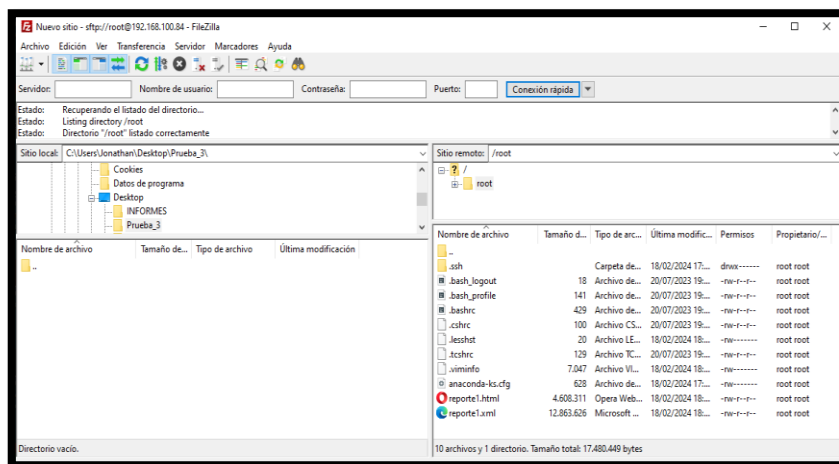


Figura 3.70 Visualización archivos copiados

Se procedió a seleccionar el reporte1.html y se lo copia en la carpeta de *Windows* creada, establecida en la Figura 3.71.

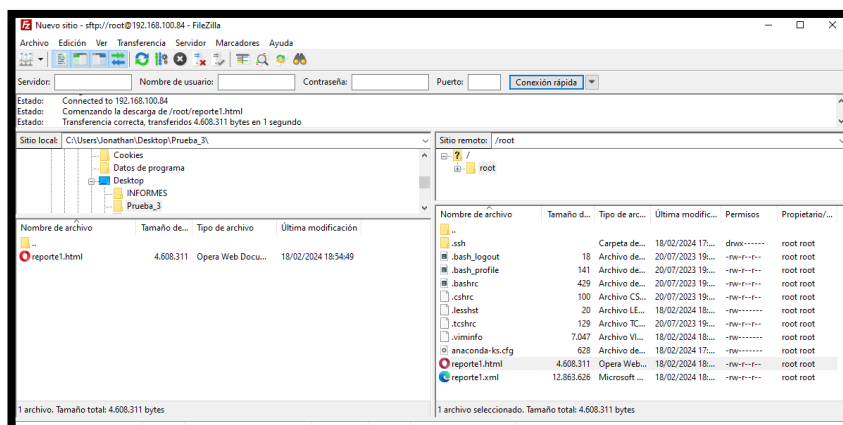


Figura 3.72 Archivos enviados desde FileZilla a *Windows*

De inmediato se evidenció que el reporte1.html se envió correctamente, permitiendo observar las vulnerabilidades que presenta el servidor, véase en la Figura 3.73.

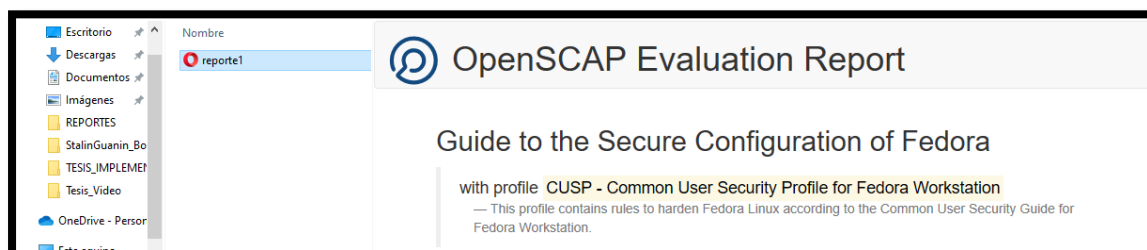


Figura 3.73 Reporte visualizado en carpeta de escritorio *Windows*

Como se muestra en la Figura 3.74. los resultados en el primer reporte ejecutado, es decir evidenciando 125 vulnerabilidades aprobadas y 136 fallidas teniendo un porcentaje del 69.85%.

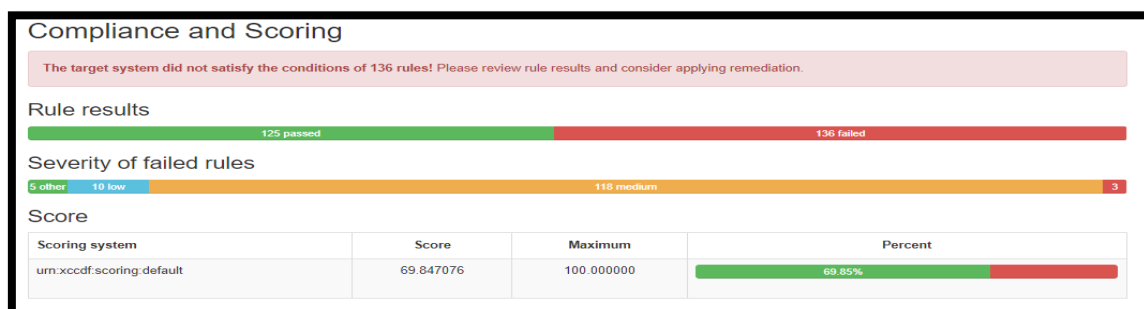


Figura 3.74 Resultados de primer reporte sin ninguna medida de vulnerabilidad

3.2 Implementación de seguridad en un sistema operativo de servidor

Después de verificar el primer reporte de vulnerabilidades, se visualizó que Postfix funciona de forma correcta, a continuación, para la realización del segundo reporte de vulnerabilidades se establecen 4 perfiles las cuales van a ser solventadas en base a CIS, para ello se obtiene la documentación que se la puede descargar directamente en la página web CIS *Benchmarks* Anexo 4, en donde se descarga toda la documentación de CIS, como se identifica en la Figura 3.75



Figura 3.75 Documentación CIS.

En la documentación se establecieron los siguientes parámetros que son asignados los cuales permiten que el reporte generado por la herramienta se alinea en su totalidad con los estándares de CIS. El reporte de vulnerabilidades se encuentra generado en formato html el cual detalla los puntos los cuales es como agrupa las reglas en cómo están asociadas mediante el perfil OVAL, las reglas se las puede ver de forma individual sin problema alguno, se utiliza el grupo de reglas por default, véase Figura 3.76

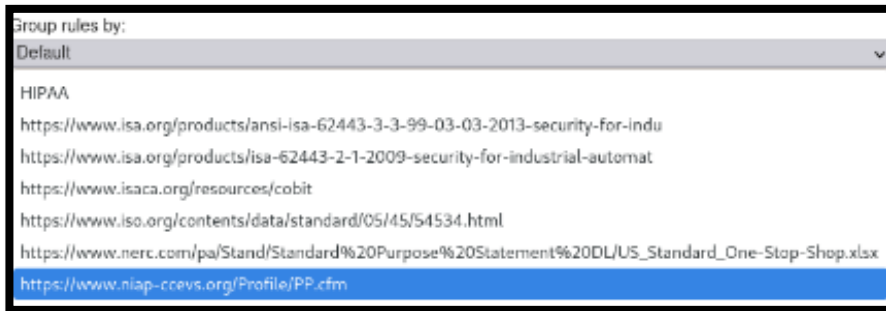


Figura 3.76 Puntos de referencia

A continuación, se procedió a corregir la primera vulnerabilidad las cuales forman parte de los lineamientos del CIS, se selecciona la vulnerabilidad *Uninstall dovecot Package* en donde detalla que se puede solventar haciendo click en *Remediation Shell script*, como se menciona en la Figura 3.77.

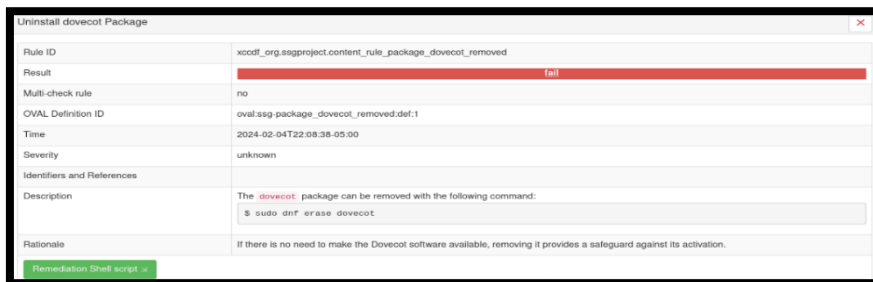


Figura 3.77 Primera vulnerabilidad basado en CIS

Luego mediante la aplicación MINGW64, se procede a crear los archivos en *Windows* los cuales permitirán enlazar para poder ejecutar con la extensión *.sh* en la máquina virtual, véase en la Figura 3.78.

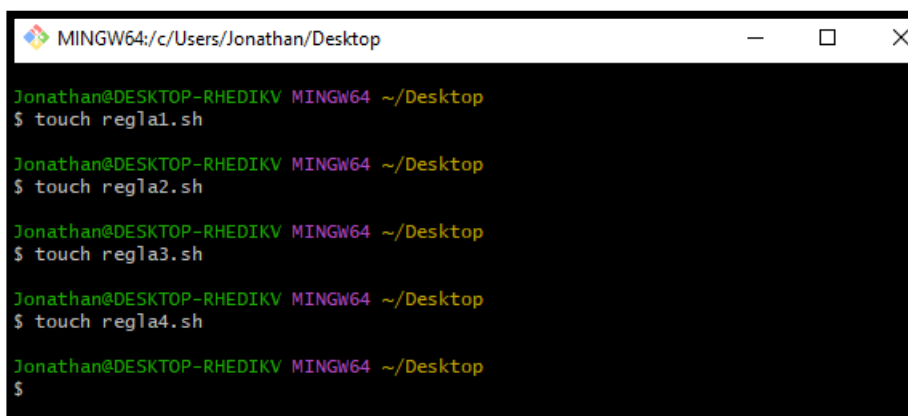


Figura 3.78 Archivos *touch*

A continuación, se ingresó a la remediación que va a ser ejecutada, como se muestra en la Figura 3.79.

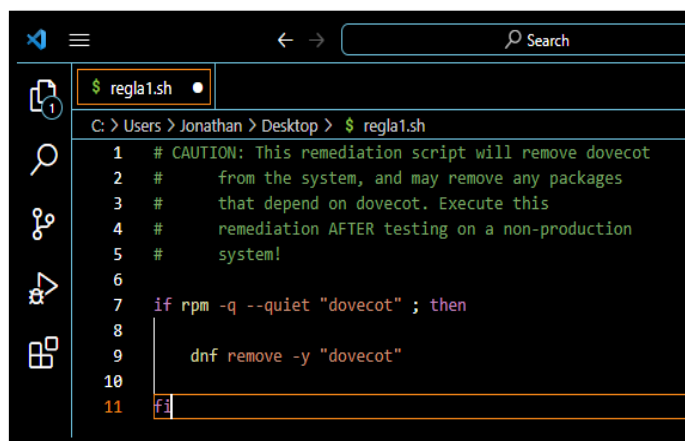
```
Remediation Shell script ↘
Complexity:
Disruption:
Strategy:

# CAUTION: This remediation script will remove dovecot
#           from the system, and may remove any packages
#           that depend on dovecot. Execute this
#           remediation AFTER testing on a non-production
#           system!

if rpm -q --quiet "dovecot" ; then
    dnf remove -y "dovecot"
fi
```

Figura 3.79 Script de remediación regla1.sh

Luego se copió el Script de la Figura 3.79 en *visual studio code* el cual permitirá generar un archivo con extensión .sh para que se pueda ejecutar en Fedora Server. Evidenciado en la Figura 3.80.



```
$ regla1.sh
C: > Users > Jonathan > Desktop > $ regla1.sh
1 # CAUTION: This remediation script will remove dovecot
2 #           from the system, and may remove any packages
3 #           that depend on dovecot. Execute this
4 #           remediation AFTER testing on a non-production
5 #           system!
6
7 if rpm -q --quiet "dovecot" ; then
8
9     dnf remove -y "dovecot"
10
11 fi
```

Figura 3.80 Archivo generado en *Visual Studio Code* regla1.sh

Después se procedió a guardar el archivo, este proceso se realiza con las 3 vulnerabilidades restantes. Luego se toma la segunda regla de vulnerabilidad *Uninstall httpd package*, como se establece en la Figura 3.81.

Set SSH Client Alive Count Max to zero	
Rule ID	xccdf_org.ssgproject.content_rule.sshd_set_keepalive_0
Result	fail
Multi-check rule	no
OVAL Definition ID	oval:ssg-sshd_set_keepalive_0:def:1
Time	2024-02-18T18:34:41-05:00
Severity	medium
Identifiers and References	<p>Identifiers 1, 12, 13, 14, 15, 16, 18, 3, 5, 7, 8, 5.5.6, APO13.01, BA03.01, BA03.02, BA03.03, DSS01.03, DSS03.05, DSS05.04, DSS05.05, DSS05.07, DSS05.10, DSS06.03, DSS06.10, 3.1.11, CCI-000879, CCI-001133, CCI-002361, 164.308(a)(4)(i), 164.308(b)(1), 164.308(b)(3), 164.310(b), 164.312(e)(1), 164.312(e)(2)(i), 4.3.3.2.2, 4.3.3.5.1, 4.3.3.5.2, 4.3.3.6.1, 4.3.3.6.2, 4.3.3.6.3, 4.3.3.6.4, 4.3.3.6.5, 4.3.3.6.6, 4.3.3.6.7, 4.3.3.6.8, 4.3.3.6.9, 4.3.3.7.2, 4.3.3.7.3, 4.3.3.7.4, 4.3.4.3.3, SR 1.1, SR 1.10, SR 1.2, SR 1.3, SR 1.4, SR 1.5, SR 1.7, SR 1.8, SR 1.9, SR 2.1, SR 2.2, A.12.4.1, A.12.4.3, A.14.1.1, A.14.2.1, A.14.2.5, A.18.1.4, A.6.1.2, A.6.1.5, A.7.1.1, A.9.1.2, A.9.2.1, A.9.2.2, A.9.2.3, A.9.2.4, A.9.2.6, A.9.3.1, A.9.4.1, A.9.4.2, A.9.4.3, A.9.4.4, A.9.4.5, CIP-004.6 R2.2.3, CIP-007.3 R5.1, CIP-007.3 R5.2, CIP-007.3 R5.3.1, CIP-007.3 R5.3.2, CIP-007.3 R5.3.3, AC-2(5), AC-12, AC-17(a), SC-10, CM-6(a), DE-CM-1, DE-CM-3, PRAC-1, PRAC-4, PRAC-5, PRAC-7, PR-IP-2, Req-8.1.8, SRG-OS-000126-GPOS-00066, SRG-OS-000163-GPOS-00072, SRG-OS-000279-GPOS-00109</p>
Description	The SSH server sends at most <code>ClientAliveCountMax</code> messages during a SSH session and waits for a response from the SSH client. The option <code>ClientAliveInterval</code> configures timeout after each <code>ClientAliveCountMax</code> message. If the SSH server does not receive a response from the client, then the connection is considered unresponsive and terminated. To ensure the SSH timeout occurs precisely when the <code>ClientAliveInterval</code> is set, set the <code>ClientAliveCountMax</code> to value of 0 in <code>/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf</code>

Figura 3.81 Segunda vulnerabilidad a corregir

Como se ejecuta en la Figura 3.82, se procedió a copiar el script de remediación.

```
# Remediation is applicable only in certain platforms
if [ ! -f /.dockerenv ] && [ ! -f /run/.containerenv ]; then

mkdir -p /etc/ssh/sshd_config.d
touch /etc/ssh/sshd_config.d/00-compliancecode-hardening.conf

LC_ALL=C sed -i "/^\s*ClientAliveCountMax\s\+/Id" "/etc/ssh/sshd_config"
LC_ALL=C sed -i "/^\s*ClientAliveCountMax\s\+/Id" "/etc/ssh/sshd_config.d/"*.conf
if [ -e "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf" ]; then

    LC_ALL=C sed -i "/^\s*ClientAliveCountMax\s\+/Id" "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf"
else
    touch "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf"
fi
# make sure file has newline at the end
sed -i -e '$a\' "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf"

cp "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf" "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf.bak"
# Insert at the beginning of the file
printf '%sn' "ClientAliveCountMax 0" > "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf"
cat "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf.bak" >> "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf"
# Clean up after ourselves.
rm "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf.bak"

else
    >&2 echo "Remediation is not applicable, nothing was done"
fi
```

Figura 3.82 Segunda remediación regla2.sh

A continuación, como se evidencia en la Figura 3.83 el script se lo ejecutó en *visual studio code*.

```
C:\Users\Jonathan\Desktop> $ regla2.sh
1 # Remediation is applicable only in certain platforms
2 if [ ! -f /.dockerenv ] && [ ! -f /run/.containerenv ]; then
3
4 mkdir -p /etc/ssh/sshd_config.d
5 touch /etc/ssh/sshd_config.d/00-compliancecode-hardening.conf
6
7 LC_ALL=C sed -i "/^\s*ClientAliveCountMax\s\+/Id" "/etc/ssh/sshd_config"
8 LC_ALL=C sed -i "/^\s*ClientAliveCountMax\s\+/Id" "/etc/ssh/sshd_config.d/"*.conf
9 if [ -e "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf" ]; then
10
11     LC_ALL=C sed -i "/^\s*ClientAliveCountMax\s\+/Id" "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf"
12 else
13     touch "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf"
14 fi
15 # make sure file has newline at the end
16 sed -i -e '$a\' "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf"
17
18 cp "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf" "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf.bak"
19 # Insert at the beginning of the file
20 printf '%sn' "ClientAliveCountMax 0" > "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf"
21 cat "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf.bak" >> "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf"
22 # Clean up after ourselves.
23 rm "/etc/ssh/sshd_config.d/00-compliancecode-hardening.conf.bak"
24
25 else
26     >&2 echo "Remediation is not applicable, nothing was done"
27 fi
```

Figura 3.83 Regla2.sh en *visual studio code*

Después en la Figura 3.84 se estableció la tercera vulnerabilidad *Ensure SSH MaxStartups is configured*.

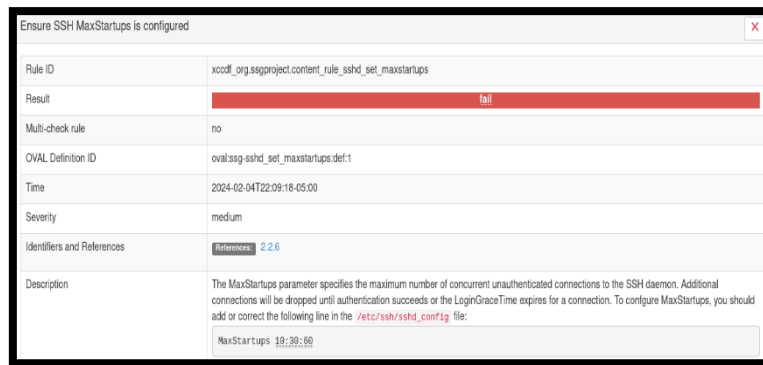


Figura 3.84 Tercera vulnerabilidad

Luego se procede a copiar el script de remediación, como se establece en la Figura 3.85.

```
# Remediation is applicable only in certain platforms
if [ ! -f /.dockerenv ] && [ ! -f /run/.containerenv ]; then
var_sshd_set_maxstartups='10:30:60'

if [ -e "/etc/ssh/sshd_config" ]; then
    LC_ALL=C sed -i "/^\s*MaxStartups\s\+/Id" "/etc/ssh/sshd_config"
else
    touch "/etc/ssh/sshd_config"
fi
# make sure file has newline at the end
sed -i -e '$a\' "/etc/ssh/sshd_config"

cp "/etc/ssh/sshd_config" "/etc/ssh/sshd_config.bak"
# Insert at the beginning of the file
printf '%s\n' "MaxStartups $var_sshd_set_maxstartups" > "/etc/ssh/sshd_config"
cat "/etc/ssh/sshd_config.bak" >> "/etc/ssh/sshd_config"
# Clean up after ourselves.
rm "/etc/ssh/sshd_config.bak"

else
    >&2 echo 'Remediation is not applicable, nothing was done'
fi
```

Figura 3.85 remediación regla3.sh

Después se ejecutó el archivo y se copió la remediación en *Visual Studio Code*, como se ejecuta en la Figura 3.86.

The screenshot shows the Visual Studio Code editor with a terminal window open. The terminal shows the execution of the script `regla3.sh`. The script content is the same as in Figure 3.85. The terminal output shows the script being executed successfully.

Figura 3.86 Script regla3.sh en visual studio code

A continuación, se ejecuta la cuarta vulnerabilidad denominada *Set SSH Daemon LogLevel to VERBOSE*, véase en la Figura 3.87

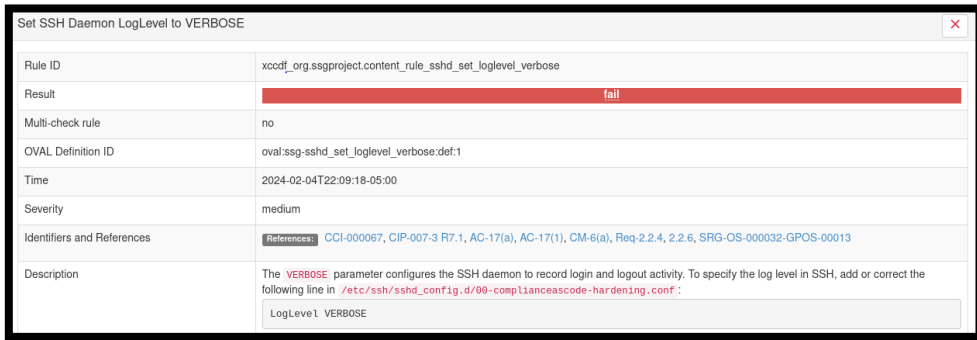


Figura 3.87 Cuarta Vulnerabilidad

Después como se genera en la Figura 3.88 se establece el *script* de la vulnerabilidad.

```
# Remediation is applicable only in certain platforms
if [ ! -f /.dockerenv ] && [ ! -f /run/.containerenv ]; then
mkdir -p /etc/ssh/ssh_config.d
touch /etc/ssh/ssh_config.d/00-compliancecode-hardening.conf
LC_ALL=C sed -i "/^\s*LogLevel\s\+/Id" "/etc/ssh/ssh_config"
LC_ALL=C sed -i "/^\s*LogLevel\s\+/Id" "/etc/ssh/ssh_config.d/*/.conf"
if [ -e "/etc/ssh/ssh_config.d/00-compliancecode-hardening.conf" ]; then
LC_ALL=C sed -i "/^\s*LogLevel\s\+/Id" "/etc/ssh/ssh_config.d/00-compliancecode-hardening.conf"
else
touch "/etc/ssh/ssh_config.d/00-compliancecode-hardening.conf"
fi
# make sure file has newline at the end
sed -i -e '$a\' "/etc/ssh/ssh_config.d/00-compliancecode-hardening.conf"
cp "/etc/ssh/ssh_config.d/00-compliancecode-hardening.conf" "/etc/ssh/ssh_config.d/00-compliancecode-hardening.conf.bak"
# Insert at the beginning of the file
printf "%s\n" "LogLevel VERBOSE" > "/etc/ssh/ssh_config.d/00-compliancecode-hardening.conf"
cat "/etc/ssh/ssh_config.d/00-compliancecode-hardening.conf.bak" >> "/etc/ssh/ssh_config.d/00-compliancecode-hardening.conf"
# Clean up after ourselves.
rm "/etc/ssh/ssh_config.d/00-compliancecode-hardening.conf.bak"
else
>&2 echo 'Remediation is not applicable, nothing was done'
fi
```

Figura 3.88 Script de remediación vulnerabilidad 4

Luego se copió el *script* en *visual studio code*, como se observa en la Figura 3.89.

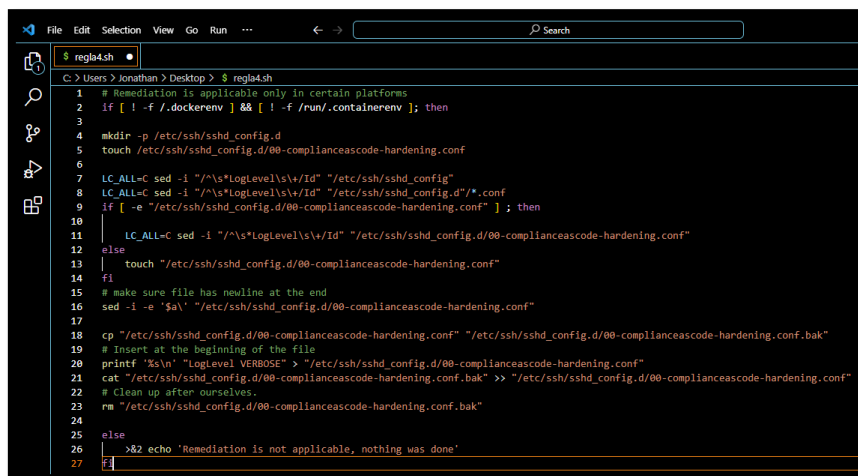


Figura 3.89 Script en visual studio code

Se guardaron los archivos generados e inmediatamente se procede a enviar desde la máquina de escritorio hasta el servidor, como se puede evidenciar en la Figura 3.90.

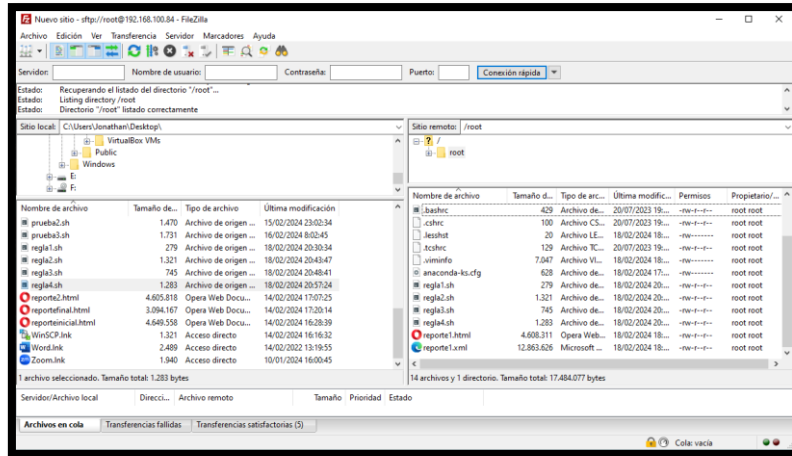


Figura 3.90 Archivos .sh enviados a máquina virtual

Se procedió a ingresar al servidor en donde se establece los comandos chmod +x para otorgar los permisos de ejecución de los archivos regla.sh y finalmente se utiliza la abreviación ./ para que se pueda ejecutar el script sin ningún problema, véase en la Figura 3.91.

```
[root@localhost ~]# chmod +x regla2.sh
[root@localhost ~]# ./regla2.sh
[root@localhost ~]#
[root@localhost ~]# chmod +x regla3.sh
[root@localhost ~]# ./regla3.sh
[root@localhost ~]#
[root@localhost ~]# chmod +x regla4.sh
[root@localhost ~]# ./regla4.sh
[root@localhost ~]#
[root@localhost ~]# chmod +x regla1.sh
[root@localhost ~]# ./regla1.sh
```

Figura 3.91 Permisos para ejecución de archivos con extensión .sh

Para evidenciar que se hayan cumplido los lineamientos de CIS, se ingresa a la documentación para evidenciar que la regla1.sh consta totalmente en CIS, como se establece en la Figura 3.92.

2.2.11 Ensure IMAP and POP3 server is not installed (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

dovecot is an open source IMAP and POP3 server for Linux based systems.

Rationale:

Unless POP3 and/or IMAP servers are to be provided by this system, it is recommended that the package be removed to reduce the potential attack surface.

Note: Several IMAP/POP3 servers exist and can use other service names. These should also be audited and the packages removed if not required.

Audit:

Run the following command to verify dovecot and cyrus-imapd are not installed:

```
# rpm -q dovecot cyrus-imapd
package dovecot is not installed
package cyrus-imapd is not installed
```

Remediation:

Run the following command to remove dovecot and cyrus-imapd:

```
# dnf remove dovecot cyrus-imapd
```

Figura 3.92 Norma extraída de la documentación CIS

A continuación, se puede evidencia en la Figura 3.93 que la segunda vulnerabilidad también conste en la documentación de CIS.

5.2.17 Ensure SSH MaxStartups is configured (Automated)

Profile Applicability:

- Level 1 - Server
- Level 1 - Workstation

Description:

The MaxStartups parameter specifies the maximum number of concurrent unauthenticated connections to the SSH daemon.

Rationale:

To protect a system from denial of service due to a large number of pending authentication connection attempts, use the rate limiting function of MaxStartups to protect availability of sshd logins and prevent overwhelming the daemon.

Audit:

Run the following command and verify that output MaxStartups is 10:30:60 or more restrictive:

```
# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -i maxstartups
maxstartups 10:30:60
```

Run the following command and verify the output:

```
# grep -Ei "\s*maxstartups\s*(([1-9]|[1-9][0-9][0-9]+):([0-9]+)|([0-9]+):([1-9]|[4-9][0-9]|[1-9][0-9][0-9]+):([0-9]+)|([1-9]|[7-9][0-9]|[1-9][0-9][0-9]+))' /etc/ssh/sshd_config
Nothing should be returned
```

Remediation:

Edit the /etc/ssh/sshd_config file to set the parameter as follows:

```
maxstartups 10:30:60
```

Figura 3.93 Segunda Regla establecida en CIS.

Tercera vulnerabilidad basada en los lineamientos de CIS, véase en la Figura 3.94.

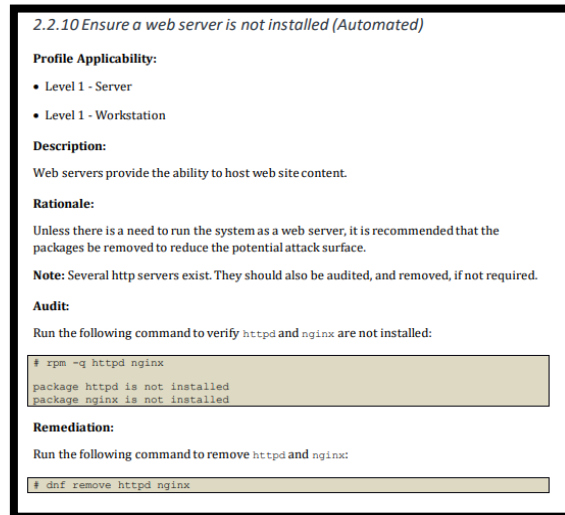


Figura 3.94 Tercera vulnerabilidad basada en CIS

A continuación, en la Figura 3.95, se evidenció que la regla se basa en la documentación CIS.

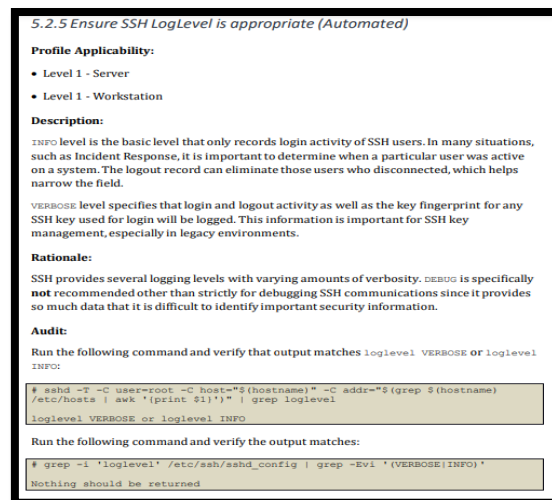


Figura 3.95 Cuarta vulnerabilidad basada en documentación CIS

A continuación, se generó un segundo reporte. Ver en las figuras Figura 3.96 y Figura 3.97.

```
[root@localhost ~]# oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_cusp_fedora --results reporte2.xml /usr/share/xml/scap/ssg/content/ssg-fedora-ds.xml
```

Figura 3.96 Segundo Reporte basado en CIS

```
[root@localhost ~]# oscap xccdf generate report reporte2.xml > reporte2.html
```

Figura 3.97 Segundo reporte formato html

A continuación, el reporte generado con el nombre reporte2.xml se lo visualiza en el formato html y se lo envía a la máquina de escritorio para visualizarlo. Como se ejecuta en la Figura 3.98.

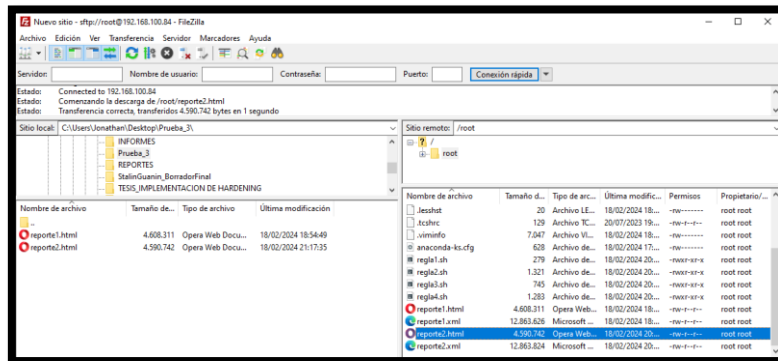


Figura 3.98 Reporte2.html

Finalmente se genera un reporte final el cual permite que todo el sistema se remedie en su totalidad, véase en la Figura 3.99.

```
[root@localhost ~]# oscap xccdf eval --remediate --profile xccdf_org.ssgproject.content_profile_cusp_fedora --results reporte3final.xml /usr/share/xml/ocap/ssg/content/ssg-fedora-ds.xml
```

Figura 3.99 Reporte final

Se procede a transformar el archivo reporte3final.xml al formato html. Como se observa Figura 3.100.

```
[root@localhost ~]# oscap xccdf generate report reporte3final.xml > reporte3final.html
```

Figura 3.100 Reporte Final transformado de xml a html

Método para enviar el archivo reppte3final.html a la carpeta de escritorio. Ver Figura 3.101.

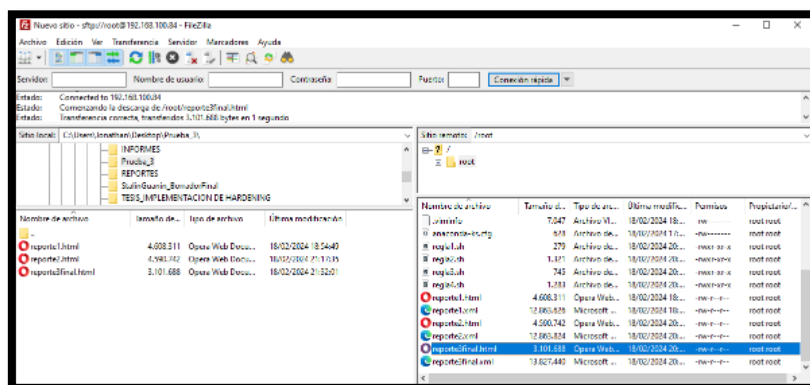


Figura 3.101 Visualización Reporte final.

3.3 Análisis de los reportes, resultado de la aplicación de la herramienta de escaneo

El primer informe se puede evidenciar un porcentaje del 69.85%, luego en el segundo reporte se puede evidenciar las reglas se han corregido y se tiene un sistema mucho más reforzado, en donde se han corregido 130 reglas son aprobadas y 131 fallidas, teniendo un porcentaje del 71.74%, tal como se evidencia en la Figura 3.102.

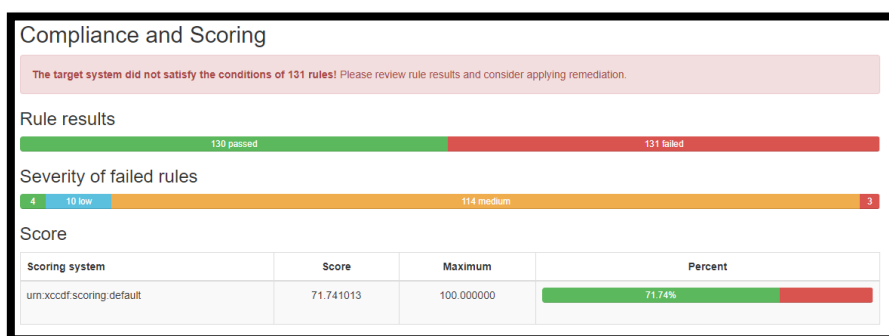


Figura 3.102 Segundo reporte basado en CIS

El resultado final luego de remediar todo el sistema, es el siguiente establecido en la Figura 3.103, en donde muestra como 259 vulnerabilidades aprobadas teniendo un porcentaje del 99.09% implementado *Hardening* para un sistema totalmente reforzado.

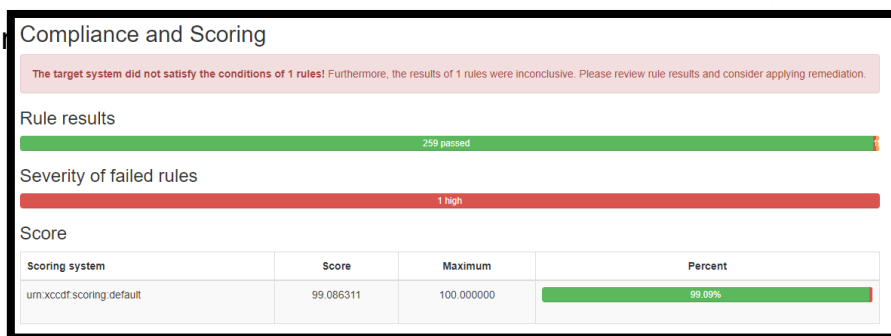


Figura 3.103. Vulnerabilidades corregidas en su totalidad.

Se mantiene el perfil establecido ya que la herramienta me permite elegir debido a tiempos y optimización, se permite utilizar al mismo el comando ejecutable. Gracias a la herramienta permite realizar el comando que corrige totalmente el sistema Fedora Server.

3.4 Verificación del *Hardening* del sistema operativo en base a los elementos de la triada CIA

Para los lineamientos de *Hardening* se elige la documentación basada en CIS, Para lo cual se tomaron los parámetros que se evidencian en la documentación de cis. Los puntos establecidos están en el Anexo II.

Los elementos de la triada CIA dependen entre si ya que al no tener confidencialidad es muy probable que se traspase la integridad de los datos, archivos, recursos, etc., de información es decir que si un usuario no tiene autorización de evidenciar la información la puede alterar, trabaja directamente con la ciberseguridad.

Cuando se modifica la integridad de información todas las claves se verán en riesgo ya que no se podrá tener el servicio esperado dando lugar a la disponibilidad.

Evaluación de normas relacionadas con la tríada CIA con un nivel alto, medio y bajo de gravedad

A continuación, se establece la verificación que enfoca en tres principios claves para un sistema totalmente robusto, es decir examinar y demostrar la implementación de medidas de seguridad.

En la Tabla 3.1 se lleva a cabo la verificación y determinación del impacto basado en la tríada CIA, las cuales presentan vulnerabilidades solventadas, especificando las políticas de seguridad según el marco de referencia CIS.

Tabla 3.1. Vulnerabilidades realizadas

VULNERABILIDAD	MEDIDA	ESTADO
Uninstall dovecot Package	INKNOWN	PASS
Uninstall httpd package	MEDIA	PASS
Ensure SSH MaxStartups is configured	MEDIA	PASS
Set SSH Daemon LogLevel to VERBOSE	MEDIA	PASS

La evaluación de estándares vinculados a la triada de seguridad informática desempeña un rol esencial en la determinación del nivel de gravedad asociado a posibles vulnerabilidades y amenazas en un entorno digital. En este contexto, se categorizan los estándares y controles según un sistema de niveles de gravedad que aborda la criticidad de cada aspecto de la triada CIA.

A un nivel elevado de gravedad, se identifican las normativas que afectan directamente la confidencialidad, integridad y disponibilidad críticas de los datos y sistemas. Estas normativas suelen estar vinculadas con políticas sólidas de acceso, cifrado avanzado y redundancias para asegurar la continuidad operativa.

A nivel intermedio, se evalúan las normativas que, aunque relevantes, pueden no tener un impacto inmediato en la seguridad crítica del sistema. Estas pueden incluir prácticas de respaldo regulares, controles de acceso más estándar y medidas de cifrado convencionales.

En el nivel bajo de gravedad, se consideran las normativas que, aunque importantes, podrían no representar una amenaza inminente para la triada CIA. Aquí, se incluyen prácticas de seguridad menos críticas, pero aún esenciales, como actualizaciones regulares de *software* y controles de acceso básicos.

La evaluación jerárquica de normativas según su impacto en la triada CIA ofrece una visión estratégica que permite priorizar eficientemente los esfuerzos y recursos de seguridad, abordando las vulnerabilidades proporcionalmente a su gravedad y asegurando una postura de seguridad sólida y equilibrada

Como se observa en la Tabla 3.1 se estableció 4 vulnerabilidades las cuales fueron sustentadas en su totalidad.

Parámetros de buenas prácticas para fortalecer la seguridad en un sistema operativo de servidor.

Reforzar la seguridad de un sistema de tipo servidor implica incorporar prácticas esenciales para mitigar vulnerabilidades potenciales. Es por ello por lo que se presentan algunos elementos clave:

1. Mantenimiento actualizado: asegurarse de que el servidor y cada aplicación estén actualizados para abordar posibles vulnerabilidades.

2. Configuración segura: ajustar las configuraciones del servidor para adherirse a principios de privilegio mínimo, desactiva servicios no esenciales y aplica configuraciones seguras.
3. Firewall: establecer un firewall para gestionar el tráfico de red y autorizar únicamente conexiones necesarias.
4. Acceso seguro: implementación de las medidas de autenticación sólidas, ya sea contraseñas robustas y restringir el acceso a usuarios debidamente autorizados.
5. Auditoría de registros: activar la auditoría de registros para supervisar actividades y detectar posibles intrusiones o comportamientos inusuales.
6. Cifrado de datos: cifrar para proteger la información confidencial.
7. Respaldos periódicos: ejecutar respaldos de forma regular y verifica su integridad para garantizar una recuperación eficaz en caso de pérdida de datos.
8. Monitoreo de seguridad: utilizar herramientas de monitoreo de seguridad para identificar y abordar eventos no habituales o intentos de intrusión.

Directrices para optimizar la seguridad en un servidor de correo electrónico resistente, evitando repeticiones innecesarias.

Así como es esencial fortalecer el sistema operativo del servidor, aplicar normas y reglas conforme al marco de referencia CIS también se debe mitigar vulnerabilidades en el servidor de correo. Es por ello por lo que se proporciona una guía de buenas prácticas destinada a mantener segura de posibles ataques al servidor.

1. Mantenimiento actualizado constantemente: aplicar de manera consistente las actualizaciones en el *software* asociado al servidor de correo.
2. Configuración mínima y segura establecida: establecer configuraciones seguras, aplicando el principio de privilegio mínimo y desactivando servicios no esenciales.
3. *Firewall* configurado para ser eficiente: establecer un firewall que permita únicamente conexiones esenciales al servidor de correo electrónico.

4. Implementación de autenticación robusta: adoptar medidas de autenticación sólidas, limitando el acceso a usuarios autorizados.
5. Auditoría de registros habilitada: activar la auditoría de registros para identificar de manera temprana actividades inusuales o intentos de intrusión.
6. Utilización de cifrado integral: emplear cifrado para asegurar la confidencialidad de los datos tanto en reposo como durante su transmisión.
7. Implementación de respaldo y recuperación eficientes: realizar respaldos periódicos y verifica su integridad para garantizar una pronta recuperación en caso de pérdida de datos.
8. Monitoreo proactivo mediante herramientas especializadas: usar herramientas de monitoreo para identificar y abordar de manera proactiva amenazas y violaciones de seguridad.
9. Gestión ágil de parches de seguridad: establecer un proceso eficiente para manejar y aplicar de forma rápida los parches de seguridad.
10. Documentación clara: desarrollar un plan de respuesta a incidentes, describiendo cada procedimiento a seguir frente a posibles violaciones de seguridad.
11. Evaluar la importancia del servidor de correo y su proceso de fortalecimiento.

4 CONCLUSIONES

- Postfix en Fedora como servidor de correo proporciona diversas ventajas, es decir la obtención de una seguridad robusta, un sistema muy eficiente y en la gestión de correos electrónicos muy flexible. Asegurando una comunicación fiable contribuyendo al servidor de correo
- Gracias al sistema operativo Fedora se puede establecer los parámetros de seguridad los cuales son basados en CIS cumpliendo cada estándar establecido de forma correcta.
- Las medidas de seguridad del perfil CUSP y las directrices del CIS converge en fortalecer la seguridad informática, mediante configuraciones

seguras y controles efectivos, contribuyendo así a la mitigación de riesgos y a la protección integral de la infraestructura tecnológica.

- La implementación de prácticas de seguridad alineadas con CUSP y de CIS mejoran la resistencia ante amenazas, estableciendo una postura sólida que salvaguarda la integridad y confidencialidad de los sistemas y datos.
- La aplicabilidad de integrar los preceptos delineados por CIS como estrategia de seguridad sobresale gracias a su enfoque integral y actualizado. Al adherirse a estas directrices, se instaura un marco robusto que aborda diversas amenazas cibernéticas y posibles vulnerabilidades. La vigencia constante de los estándares CIS, respaldada por su flexibilidad ante las cambiantes dinámicas de seguridad, refuerza la pertinencia de su adopción. Su enfoque completo es mitigar riesgos de manera proactiva y fortalecer la postura general de seguridad, presentándolos como una alternativa práctica y eficiente para resguardar los sistemas informáticos.
- Implementar en *Fedora Server 39* políticas de seguridad implica reconocer la función vital que desempeñan en la protección de la infraestructura tecnológica. Al establecer medidas proactivas, se establece una barrera robusta contra amenazas cibernéticas, contribuyendo significativamente a la estabilidad y funcionamiento seguro del entorno de servidor.
- Elegir *OpenScap* como la herramienta principal, para llevar a cabo el escaneo de vulnerabilidades en un entorno *Fedora Server*, se posiciona como una elección estratégica que impulsa la robustez y la seguridad del sistema de manera significativa. No solo se limita a la identificación de posibles puntos débiles en la configuración del servidor, sino que también ofrece la capacidad única de implementar y mantener estándares de seguridad ampliamente reconocidos.
- La verificación de endurecimiento a través de *OpenScap* en *Fedora Server 39* emerge como un componente esencial para consolidar la seguridad del sistema. Este procedimiento no solo se limita a la identificación y corrección de posibles vulnerabilidades en la configuración

del servidor, sino que también garantiza la conformidad del sistema con estándares de seguridad reconocidos. Los beneficios derivados abarcan desde la creación de un entorno más resistente frente a las amenazas cibernéticas hasta el fortalecimiento de la confianza en la integridad de los datos y servicios, contribuyendo en última instancia a una postura de seguridad más robusta.

- La aplicación de medidas de fortalecimiento en Fedora Server 39 se demuestra como una estrategia válida y esencial para mejorar la robustez del sistema operativo. Al implementar acciones específicas de fortalecimiento, como adoptar principios de privilegios mínimos y emplear tecnologías para mitigar vulnerabilidades conocidas, se crea un entorno más resistente ante posibles amenazas cibernéticas. Disminuyendo las áreas expuestas a ataques, restringir el impacto de potenciales vulnerabilidades y consolidar la postura global de seguridad en todo el sistema operativo, contribuyendo efectivamente a protección de sistemas informáticos, garantizando la integridad de información.

5 RECOMENDACIONES

- Tener las actualizaciones de forma periódica del software ya que fortalecerá las funciones del sistema operativo en este caso de fedora.
- Al momento de establecer la máquina virtual dar el almacenamiento adecuado ya que caso contrario no se podrá instalar las aplicaciones o herramientas para la ejecución del proyecto.
- Se recomienda encarecidamente mantener un enfoque proactivo en la gestión de la seguridad. Esto implica la actualización constante de las políticas de seguridad para adaptarse a las nuevas amenazas y vulnerabilidades emergentes. Además, la realización regular de auditorías de seguridad es importante para identificar las políticas implementadas y para identificar posibles puntos de mejora. Fomentar la conciencia sobre las prácticas seguras entre los usuarios y administradores se convierte en una estrategia clave para fortalecer la postura de seguridad de manera continua, su implementación y mantenimiento diligente de políticas de

seguridad en Fedora Server son cruciales para garantizar un entorno tecnológico resistente y protegido contra las amenazas actuales y futuras.

- Incorporar *OpenScap* de manera regular dentro de la rutina de seguridad del servidor. La realización de escaneos sistemáticos y recurrentes permite detectar de manera proactiva y abordar las vulnerabilidades potenciales antes de que puedan ser explotadas. Además, aprovechar las funcionalidades de automatización para implementar y mantener directrices de seguridad estándar simplifica la gestión y asegura un cumplimiento constante de las mejores prácticas de seguridad. Integrar de manera integral esta herramienta en la estrategia de seguridad en Fedora Server refuerza la postura defensiva del sistema, garantizando un entorno más seguro y mejor preparado ante posibles riesgos y amenazas.
- Agregar de manera sistemática la verificación de endurecimiento a través de *OpenScap* en la gestión de la seguridad de Fedora Server. La realización periódica de auditorías asegura que el sistema se mantenga en sintonía con los estándares de seguridad vigentes y pueda adaptarse a las cambiantes dinámicas del panorama de amenazas. La capacidad de automatización facilitada y agiliza este proceso, permitiendo una gestión eficiente y garantizando la uniformidad en la seguridad, contribuyendo a una protección proactiva y continua del servidor, generando beneficios tangibles en términos de estabilidad, confiabilidad y resiliencia ante potenciales riesgos de seguridad.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] «01introduccion.pdf,» [En línea]. Available: <https://wcruzy.pe/so/01introduccion.pdf>. [Último acceso: 29 12 2023].
- [2] «Microsoft Word - FOLLETO-SISTEMAS-OPERATIVO.docx,» [En línea]. Available:

https://ridda2.utp.ac.pa/bitstream/handle/123456789/5074/folleto_sistemas_operativos.pdf?sequence=3&isAllowed=y. [Último acceso: 29 12 2023].

- [3] «Somos Libres,» [En línea]. Available: <https://www.somoslibres.org/index.php/30-nieuws/fedora/12298-lanzamiento-de-fedora-39-con-gnome-45-y-linux-6-5>. [Último acceso: 29 12 2023].
- [4] «Desde Linux,» [En línea]. Available: <https://blog.desdelinux.net/fedora-39-ya-fue-liberado-y-estas-son-sus-novedades/>. [Último acceso: 29 12 2023].
- [5] «arsys,» [En línea]. Available: <https://www.arsys.es/blog/postfix-caracteristicas-mejores-practicas>. [Último acceso: 29 12 2023].
- [6] «PUIG CASTELLAR,» [En línea]. Available: <https://elpuig.xeill.net/Members/vcarceler/articulos/introduccion-a-dovecot#:~:text=Dovecot%20es%20un%20MDA%20que,no%20borrar%20el%20correo%20descargado>. [Último acceso: 29 12 2023].
- [7] «ManageEngine,» [En línea]. Available: <https://www.manageengine.com/latam/controles-de-seguridad-critica-cis.html>. [Último acceso: 29 12 2023].
- [8] «LinuxAdictos,» [En línea]. Available: <https://www.linuxadictos.com/openscap-herramientas-seguridad-linux.html>. [Último acceso: 30 12 2023].
- [9] «Ne Digital,» [En línea]. Available: <https://www.nedigital.com/es/blog/hardening-de-servidores#:~:text=de%20seguridad%20inform%C3%A1tica,-,%C2%BFQu%C3%A9%20es%20el%20hardening%20de%20servidores%3F,ataque%20de%20una%20infraestructura%20tecnol%C3%B3gica..> [Último acceso: 30 12 2023].
- [10] «studocu,» [En línea]. Available: [https://www.studocu.com/bo/document/universidad-autonoma-gabriel-](https://www.studocu.com/bo/document/universidad-autonoma-gabriel)

rene-moreno/sistemas-operativos-ii/informe-tecnico-hardening-fedora/36107127. [Último acceso: 30 12 2023].

- [11] «OVAL,» [En línea]. Available: https://oval.mitre.org/adoption/official_adopters.html. [Último acceso: 30 12 2023].
- [12] «CompuHoy.com,» [En línea]. Available: <https://www.compuhoy.com/cuales-la-diferencia-entre-el-servidor-fedora-y-la-estacion-de-trabajo/>. [Último acceso: 30 12 2023].
- [13] «slideshare,» [En línea]. Available: <https://es.slideshare.net/LuisFerAguas/9unidad-3-marcos-de-referencia-para-seguridad-de-la-informacin-31-iso-27000>. [Último acceso: 30 12 2023].
- [14] «BLOG,» [En línea]. Available: <https://preyproject.com/es/blog/marcos-de-ciberseguridad-la-guia-definitiva>. [Último acceso: 30 12 2023].

7 ANEXOS

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 21 de Febrero de 2024

De mi consideración:

Yo, GABRIELA KATHERINE CEVALLOS SALAZAR, en calidad de Directora del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE HARDENING EN FEDORA CON CIS asociado al IMPLEMENTACIÓN DE HARDENING EN SISTEMAS OPERATIVOS DE SERVIDOR elaborado por el estudiante STALIN RAMIRO GUANIN RODRIGUEZ de la carrera en Tecnología Superior en Redes y Telecomunicaciones, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 12%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

file:///C:/Users/ASUS/Downloads/turnitin_sr_stalin_guan%C3%ADn.pdf

Atentamente,

GABRIELA KATHERINE CEVALLOS SALAZAR

Director

Escuela de Formación de Tecnólogos

ANEXO II: Enlaces

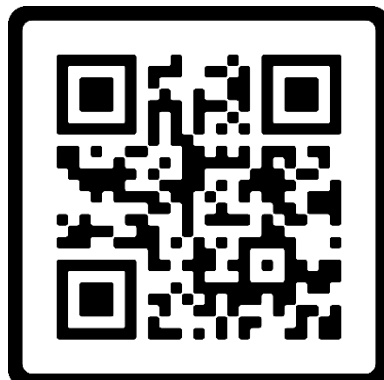
Anexo II.I Código QR de Perfil OSPP



Anexo III.II Código QR de Perfil PCI DSS



Anexo III.III Código QR de Perfil Estándar de Fedora



Anexo IIV.IV Código QR de Documentación CIS



Anexo IV.V Código QR de Reporte Inicial



Anexo IIV.IV Código QR de Segundo Reporte



Anexo IVII.VII Código QR Reporte Final



Anexo IVIII.VII Video Final

