ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE *HARDENING* MEDIANTE LA HERRAMIENTA DE DEVOPS ANSIBLE

IMPLEMENTACIÓN DE *HARDENING,* MEDIANTE ANSIBLE, EN UN SISTEMA OPERATIVO DE SERVIDOR LINUX DE BASE RED HAT

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

CHRISTIAN PATRICIO CONSTANTE VACA

christian.constante@epn.edu.ec

DIRECTOR: GABRIELA KATHERINE CEVALLOS SALAZAR

gabriela.cevalloss@epn.edu.ec

DMQ, febrero 2024

CERTIFICACIONES

Yo, CHRISTIAN PATRICIO CONSTANTE VACA 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.

CHRISTIAN PATRICIO CONSTANTE VACA

christian.constante@epn.edu.ec

christianpcv103@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por CHRISTIAN PATRICIO CONSTANTE VACA, 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.

CHRISTIAN PATRICIO CONSTANTE VACA

DEDICATORIA

A mis padres, Jorge Constante y Marlene Vaca,

Su amor incondicional y apoyo constante han sido mi mayor fortaleza. Gracias por confiar en mí en cada paso del camino hasta alcanzar esta meta, por ser mis guías, por enseñarme la importancia de valores invaluables como la perseverancia y la dedicación.

A mi hermano,

Tu guía y consejos han sido fundamentales en mi camino. Gracias por estar siempre a mi lado, alentándome a ser cada día mejor y brindarme apoyo cuando más lo he necesitado.

Este trabajo está dedicado a ellos, mis pilares, mi familia. Su influencia ha marcado cada momento en mi vida, mi gratitud hacia ustedes por el inmenso sacrificio que han hecho para que pueda llegar hasta aquí.

Christian

AGRADECIMIENTO

Agradezco a Dios por iluminar mi camino y brindarme la fuerza necesaria para enfrentar los desafíos. Creo que cada experiencia tiene un propósito y me siento agradecido por las oportunidades y cambios que se han presentado en mi vida para mi crecimiento.

Quiero expresar mi profundo agradecimiento a mis padres, Jorge Constante y Marlene Vaca. Su amor incondicional, valores sólidos y apoyo constante han sido el fundamento de mi educación y desarrollo. Agradezco a Dios por bendecirme con unos padres amorosos y dedicados, quienes con su esfuerzo incansable me han brindado la oportunidad de alcanzar mis metas académicas. Su ejemplo de unidad familiar y apoyo entre hermanos ha sido fundamental para mi éxito y bienestar.

A mi hermano, gracias por ser mi compañero en esta etapa de mi vida. Agradezco por la conexión que compartimos y por la fortaleza que aportas a mi vida.

A mi directora de tesis, Gabriela Katherine Cevallos Salazar, quiero expresar mi sincero agradecimiento por su paciencia, empatía y dedicación durante el desarrollo de este proyecto. Sus conocimientos y consejos han sido fundamentales, y su habilidad para enseñar ha sido una guía valiosa. Además, aprecio su excepcional labor como profesora y su calidez como persona durante mi desarrollo académico. Del mismo modo, agradezco a los Ingenieros Fernando Becerra y Carlos Yunga por su calidad como profesores y como personas.

Con sincero agradecimiento,

Christian

ÍNDICE DE CONTENIDOS

RESUMEN

El trabajo de titulación se centra en la implementación de *hardening* mediante Ansible en un sistema operativo de servidor Linux de base Red Hat. Para llevar a cabo esta implementación, se ejecutó un *playbook* en Alma Linux, nodo controlado, el cual es administrado por el nodo de control Ansible instalado en Fedora. Ansible utiliza el protocolo SSH para establecer una conexión segura entre ambos nodos. Este *playbook* contiene una serie de tareas diseñadas para fortalecer la seguridad del sistema operativo de servidor y reducir su superficie de ataque. Estas tareas hacen uso de los módulos de Ansible para gestionar los cambios en las configuraciones de manera eficiente.

El proyecto se divide en cinco etapas. En la primera etapa, se proporciona una visión general del proyecto, donde se describen los objetivos generales y específicos, el alcance del trabajo, finalmente el marco teórico. Este marco teórico proporciona la base para comprender tanto el funcionamiento de Ansible como la implementación de *hardening*. Se describen algunas herramientas de escaneo que permiten identificar y corregir vulnerabilidades en sistemas operativos, así como también los distintos marcos de referencia en la ciberseguridad.

La segunda etapa se enfoca en la metodología de la investigación, donde se explica el tipo de investigación utilizada y el desarrollo del proyecto en función de los objetivos establecidos. Se detallan los pasos seguidos para llevar a cabo la implementación de *hardening* utilizando Ansible.

En la tercera etapa, se presentan los resultados del proyecto. Esto incluye la identificación inicial de vulnerabilidades en el sistema operativo a través de la herramienta de escaneo SCAP *Workbench*, la implementación de soluciones mediante el *playbook* de Ansible y la realización de un análisis final para verificar la efectividad de las medidas implementadas.

Las dos últimas etapas del proyecto contienen las conclusiones basadas en los resultados obtenidos en relación con los objetivos planteados. Además, se ofrecen recomendaciones para futuros trabajos con relación a la implementación de *hardening* utilizando Ansible en sistemas Linux.

PALABRAS CLAVE: *Hardening*, Ansible, *playbook*, SSH, SCAP *Workbench*.

ABSTRACT

The degree work centers on the implementation of hardening using Ansible on a Red Hat based Linux server operating system. To realize this implementation, a playbook was run on Alma Linux, a controlled node, which is managed by the Ansible control node installed on Fedora. Ansible uses the SSH protocol to establish a secure connection between the two nodes. This playbook contains several tasks designed to strengthen the security of the server operating system and reduce its attack surface. These tasks make use of Ansible modules to manage configuration changes efficiently.

The project is divided into five stages. In the first stage, an overview of the project is provided, describing the general and specific objectives, the scope of the work, and finally the theoretical framework. This theoretical framework provides the basis for understanding both the operation of Ansible and the implementation of hardening. Some scanning tools that allow identifying and correcting vulnerabilities in operating systems are described, as well as the different cybersecurity frameworks.

The second stage focuses on the research methodology, where the type of research used and the development of the project according to the established objectives are explained. The steps followed to realize the implementation of hardening using Ansible are detailed.

In the third stage, the results of the project are presented. This includes the initial identification of vulnerabilities in the operating system through the SCAP Workbench scanning tool, the implementation of solutions using the Ansible playbook and the performance of a final analysis to verify the effectiveness of the implemented measures.

The last two stages of the project contain the conclusions based on the results obtained in relation to the stated objectives. In addition, recommendations for future work regarding the implementation of hardening using Ansible on Linux systems are provided.

KEYWORDS: *Hardening*, Ansible, *playbook*, SSH, SCAP *Workbench*.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El presente proyecto pretende automatizar un proceso de *hardening* en sistemas operativos de servidor, mediante el uso de la herramienta de DevOps Ansible. Ansible permite automatizar tareas de manera remota; con esto se asegura al sistema operativo de servidor, reduciendo significativamente la superficie de ataques; a la vez se garantiza eficiencia, rapidez y fiabilidad, con menos esfuerzo y riesgo de errores humanos.

Se escaneará el sistema operativo de servidor mediante una herramienta de escaneo de configuración y vulnerabilidades, donde se obtendrá un reporte inicial el cual será comparado con un reporte luego de haber aplicado, mediante Ansible, el *hardening* en el sistema operativo de servidor. Este análisis determinará si se ha mejorado la seguridad del sistema operativo de servidor.

1.1 Objetivo general

Implementar *hardening* mediante la herramienta de DevOps Ansible.

1.2 Objetivos específicos

- Identificar las vulnerabilidades en un sistema operativo de servidor sin políticas de seguridad.
- Instalar la herramienta de Ansible en el nodo de control
- Implementar el *playbook* para aplicar *hardening* al sistema operativo de servidor.
- Verificar los resultados obtenidos.

1.3 Alcance

En primera instancia se analizan las características y requerimientos de la herramienta Ansible, además se investigarán herramientas de escaneo, de configuración y vulnerabilidades. Partiendo de esto, se instalará un sistema operativo de servidor, sin ninguna política de seguridad; con la herramienta de escaneo se procede a obtener un primer informe de vulnerabilidades.

Luego se crea una máquina virtual para el nodo de control, donde se instala Ansible y se crea el *playbook* para aplicar *hardening* al sistema operativo de servidor. Con esto se despliega de manera remota las configuraciones necesarias para la implementación de *hardening*. Se obtiene un nuevo reporte de vulnerabilidades y se observará cuáles parámetros se han solventado.

Con esto se implementa *hardening* de manera automatizada, en un sistema operativo de 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

Ansible

Ansible es una herramienta de automatización de las Tecnologías de la Información (TI) que se emplea para automatizar la implementación, gestión y configuración de aplicaciones y sistemas. En forma general, se puede afirmar que Ansible opera mediante la elaboración de *playbooks* que detallan los pasos requeridos para llevar a cabo una tarea específica. Los *playbooks* están redactados en *Yet Another Markup Language* (YAML) y se emplean para establecer el estado deseado del sistema. YAML es un lenguaje de programación ampliamente utilizado que se caracteriza por su legibilidad y facilidad de comprensión. Posteriormente, Ansible se encarga de llevar el sistema al estado deseado por medio de la ejecución de los pasos descritos en el *playbook* [1].

Antes de instalar y configurar una aplicación, lo que se necesita es preparar la infraestructura por primera vez, dado que llevar a cabo la implementación manual de cientos o miles de servidores resulta inviable. Por lo tanto, las organizaciones que buscan expandir su infraestructura tecnológica de manera ágil y confiable optan por utilizar los *playbooks* de Ansible. Con esta solución, es posible diseñar una instancia y ponerla en funcionamiento de manera inmediata, o bien, utilizar múltiples servidores adicionales que compartan los mismos parámetros y detalles de la infraestructura. Una vez que el entorno haya sido preparado, se encuentra en condiciones de ser configurado de manera automatizada [1].

El nodo de control o nodo de gestión es el responsable de supervisar toda la ejecución del *playbook*, este es el nodo que se usa para llevar a cabo la instalación y es aquel que alberga el archivo de inventario, el cual contiene una lista de los *hosts* en los cuales se deben ejecutar los módulos correspondientes. El nodo de control establece una conexión SSH para posteriormente ejecutar el *playbook* con los módulos en las máquinas *host* y así instalar el producto. Es así cómo funciona el mecanismo operativo de Ansible [2].

Algunas de las características de Ansible son [3]:

- Es desarrollado en el lenguaje de programación Python y se encuentra disponible en todas las distribuciones de Linux.
- Es extensible y da la posibilidad de crear módulos personalizados para adaptarse a las posibles necesidades de cada proyecto.
- Utiliza el lenguaje YAML, lo que facilita su legibilidad y facilidad de escritura, además tiene una sintaxis uniforme.
- No es necesario la instalación de agentes para que funcione en servidores remotos, lo que disminuye el riesgo de errores y mejora el rendimiento.
- Tiene una curva de aprendizaje baja para los administradores, los desarrolladores y los gerentes de TI, ya que hace uso de un lenguaje simple y ampliamente utilizado.
- Es seguro y confiable, debido a que hace uso del protocolo Secure Shell (SSH) para comunicarse con los recursos y verifica la integridad del código antes de que este sea ejecutado.
- Se caracteriza por su flexibilidad y escalabilidad, ya que posibilita la incorporación de un número variable de servidores adicionales que comparten los mismos parámetros y detalles de la infraestructura.
- Es compatible con varios sistemas operativos y plataformas, como Windows, Kubernetes, AWS, Docker, Azure, etc.

Requerimientos de la herramienta Ansible

Para instalar Ansible es necesario un nodo de control y uno o más nodos controlados. El nodo de control es el sistema en el cual se instala Ansible, mientras que los nodos controlados son los sistemas que se administran mediante Ansible [4].

Para el nodo de control, se puede hacer uso de cualquier máquina UNIX-like con Python 2, en su versión 2.7 o Python 3, a partir de la versión 3.5 y versiones más recientes, incluyendo *Red Hat*, BSD, Ubuntu, Debian, macOS y Windows que cuente con una distribución de Windows *Subsystem for* Linux (WSL) [4].

Para los nodos controlados, no se necesita la instalación de Ansible, pero se requiere también de Python 2, en su versión 2.7 o Python 3, a partir de la versión 3.5 y versiones más recientes para la ejecución del código Python que va a estar siendo generado por Ansible. Además, es requerido que el nodo controlado cuente con una cuenta de usuario ya que solo así tiene la capacidad de establecer una conexión mediante SSH con el nodo de control [4].

Herramientas de escaneo de configuración y vulnerabilidades

Las herramientas de escaneo y configuración de vulnerabilidades son programas utilizados para identificar y corregir vulnerabilidades en sistemas operativos y aplicaciones. La importancia de estas herramientas es debido a que garantizan la seguridad de los sistemas informáticos y se encargan de protegerlos contra ataques malignos [4].

Las herramientas de escaneo y configuración de vulnerabilidades son programas empleados para detectar y remediar vulnerabilidades presentes en las aplicaciones y sistemas operativos. Estas herramientas son de vital importancia para asegurar la integridad de los sistemas informáticos y salvarlos de posibles ataques perpetrados con intenciones maliciosas [5].

En el mercado se encuentran disponibles diversas herramientas de escaneo y configuración de vulnerabilidades, las cuales presentan características y funcionalidades propias. Algunas de las herramientas más comunes son Nessus, Qualys, Rapid7, Microsoft *Baseline Security Analyzer* (MBSA), GFI LanGuard, OpenVAS, BurpSuite, AlienVault USM *Anywhere*, OpenSCAP, Nikto2, entre otras [5].

Estas herramientas operan mediante la ejecución de un análisis minucioso de los sistemas informáticos con el objetivo de detectar vulnerabilidades y debilidades. Una vez que se han identificado las vulnerabilidades, estas herramientas pueden ser utilizadas para remediarlas a través de la implementación de parches y actualizaciones de seguridad. Además, tienen la capacidad de ofrecer informes exhaustivos acerca de las vulnerabilidades detectadas, así como de proponer estrategias para reducir los riesgos asociados [5].

Es fundamental considerar que estas herramientas no constituyen una solución integral para la seguridad informática. Es recomendable emplearlos en conjunto con otras medidas de seguridad, tales como antivirus, cortafuegos y políticas de seguridad de la información, con el fin de asegurar una protección integral de los sistemas informáticos [5].

Protocolo SCAP

El Protocolo de Automatización de Contenido de Seguridad (SCAP) es un conjunto de estándares en el ámbito de la seguridad informática que se emplean con el propósito de cuantificar la seguridad, evaluar el nivel de vulnerabilidad y aplicar la política de

seguridad en los sistemas informáticos. El Protocolo SCAP por sus siglas en inglés, ofrece una metodología uniforme para la expresión y el intercambio de datos relacionados con la seguridad. Esto permite a las organizaciones automatizar la evaluación de vulnerabilidades, cuantificar el nivel de seguridad y la implementación de políticas de seguridad [6].

Hay varias herramientas de escaneo de vulnerabilidades fundamentadas en el protocolo SCAP a disposición de usuarios de sistemas operativos Linux. Algunas de las herramientas más comunes son:

- **SCAP *Workbench***: Es una de las herramientas que dispone OpenSCAP, es una utilidad gráfica a diferencia de OpenSCAP Base que es una herramienta de línea de comandos. De igual forma da la posibilidad a los usuarios hacer análisis de configuración y vulnerabilidades en un único sistema sea este local o remoto, llevar a cabo la corrección del sistema y generar informes fundamentados en las evaluaciones de los análisis [7].

  Para generar informes utilizando SCAP *Workbench*, se debe abrir la aplicación y seleccionar el perfil XCCDF que se utilizará para escanear el sistema local o remoto. Posteriormente, los resultados del análisis se almacenan en forma de archivos XCCDF, ARF o HTML. Sin embargo, también existe la posibilidad de elegir otros formatos, como PDF, XML o CSV [8].
- **Nessus**: Se trata de una herramienta de *software* propietario que se emplea con el fin de automatizar diversas tareas relacionadas con la seguridad, tales como la gestión de configuraciones, la evaluación de vulnerabilidades y la orquestación de servicios. Ofrece facilidad de uso y una integración perfecta con muchas herramientas y sistemas. La implementación y el mantenimiento de este sistema no requiere la instalación de agentes en los servidores remotos, lo cual lo convierte en una solución de fácil implementación y mantenimiento. Tiene una alta escalabilidad y es capaz de gestionar sin esfuerzo grandes entornos de TI [9].
- **OpenVAS**: Se trata de una herramienta de escaneo de vulnerabilidades de código abierto, se emplea con el propósito de detectar y solucionar vulnerabilidades presentes en sistemas operativos y aplicaciones. Esta herramienta hace uso del protocolo SCAP para evaluar la seguridad y determinar la vulnerabilidad de los sistemas informáticos. En general es una herramienta fácil de usar, es posible integrarla con otras herramientas y sistemas; no es

necesario instalar agentes en los servidores remotos. Por otro lado, es escalable y puede manejar fácilmente grandes entornos de TI [10].

- **Qualys:** Qualys es una plataforma que cuenta con un conjunto de herramientas diseñadas para garantizar la seguridad y el cumplimiento normativo en entornos informáticos, proporcionando una amplia gama de funcionalidades para el análisis y la protección de dichos sistemas. Una de las herramientas disponibles es Qualys *Threat Protection*, la cual facilita la realización de evaluaciones, mediciones y cumplimientos de políticas de seguridad, además de generar informes en diversos formatos. La solución de protección aprovecha el protocolo SCAP para correlacionar la información sobre amenazas con las vulnerabilidades y el inventario de los activos de TI, ayudando así a priorizar los esfuerzos de remediación de riesgos [11].

Marcos de referencia en la ciberseguridad

Un marco de ciberseguridad se define como un conjunto de normas, directrices y procedimientos recomendados que se utilizan para gestionar los riesgos asociados con el entorno digital. En términos generales, los controles de seguridad de una empresa suelen alinearse con sus objetivos de protección, tales como la prevención de accesos no autorizados al sistema mediante la implementación de medidas como la solicitud de un nombre de usuario y contraseña [12].

El marco de referencia está pensado para entregar a los administradores de seguridad una metodología confiable y sistemática para reducir el riesgo cibernético, independientemente de la complejidad del entorno. En muchas ocasiones, su implementación es obligatoria o, al menos, se incentiva fuertemente en las organizaciones que buscan cumplir con las regulaciones estatales, industriales y de ciberseguridad a nivel internacional. Un ejemplo ilustrativo es el requisito de que una empresa, con el fin de realizar a cabo transacciones con tarjeta de crédito, debe someterse a una auditoría que certifique su conformidad con el marco de los Estándares de Seguridad de Datos de la Industria de Tarjetas de Pago (PCI DSS) [12].

Existen varios marcos de referencia que se pueden considerar para mejorar la ciberseguridad de las organizaciones. Algunos de los más conocidos son:

NIST CSF (NIST *Cybersecurity Framework*)

El acrónimo en inglés de *National Institute of Standards and Technology* (NIST) corresponde al Instituto Nacional de Estándares y Tecnología. El NIST es una entidad perteneciente al Departamento de Comercio de los Estados Unidos que tiene como

objetivo impulsar la innovación y la competitividad a través de la promoción de la ciencia, la tecnología y los estándares de medición. Dentro del campo de la ciberseguridad, se ha llevado a cabo el desarrollo de diversos marcos de referencia por parte del Instituto Nacional de Estándares y Tecnología, como es el NIST, con el propósito de ofrecer directrices y prácticas recomendadas para mejorar la gestión de los riesgos relacionados con la ciberseguridad [12].

A veces también conocido como "Marco NIST de ciberseguridad", se ha diseñado con el propósito de salvar la infraestructura crítica, incluyendo instalaciones como plantas de energía y represas, contra posibles amenazas provenientes de ataques cibernéticos. Estos principios se pueden aplicar a cualquier organización que busque una mayor seguridad, ya que es uno de varios estándares que abarcan la ciberseguridad [12].

El marco NIST de ciberseguridad, al igual que la mayoría de los marcos existentes, se caracteriza por su complejidad y su amplio alcance. La implementación práctica del marco puede requerir una extensa dedicación de tiempo y la generación de una amplia documentación, que puede incluir cientos de páginas que contienen procedimientos, controles, entre otros. No obstante, en su esencia, el marco resulta bastante comprensible. El núcleo del marco consiste en una lista de funciones de ciberseguridad que se adhieren al patrón fundamental de la ciberdefensa: identificación, protección, detección, respuesta y recuperación [12].

ISO/IEC 27001

ISO 27001/27002, a la que también se le conoce con el nombre de ISO 27K, representa un estándar de referencia internacionalmente usado para la ciberseguridad. El marco determina que una entidad que adquiera la ISO 27001 tendrá a su disposición un Sistema de Gestión de Seguridad de la Información (SGSI). Teniendo eso en cuenta, para utilizar la ISO/IEC 27001 es necesario que la administración aborde sistemáticamente los riesgos de seguridad de la información de la organización, tomando en cuenta los riesgos y brechas de seguridad [12].

El marco exige que la empresa cree y ejecute sistemas de seguridad de información (InfoSec) convincentes y completos. El objetivo de estos controles es reducir los riesgos identificados. El marco sugiere que la organización implemente un proceso continuo de gestión de riesgos. La organización debe demostrar al auditor que está utilizando lo que ISO denomina ciclo *Plan, Do, Check y Adjust* (PDCA) para recibir la certificación de conformidad con ISO 27001 [12].

Center for Internet Security (CIS)

CIS fue establecido a finales de la década de 2000 por una coalición conformada por expertos y voluntarios. Su principal objetivo consiste en desarrollar un marco de protección para las empresas, con el propósito de mitigar las amenazas relacionadas con la ciberseguridad. El sistema está constituido por un conjunto de 20 controles que periódicamente se actualizan por expertos provenientes de diversos ámbitos, como el gobierno, la academia y la industria. Estas actualizaciones se llevan a cabo con el objetivo de mantener la modernidad y la capacidad de anticipación del sistema frente a las amenazas de ciberseguridad [12].

CIS resulta beneficioso para las organizaciones que buscan implementar cambios de manera gradual y progresiva. El proceso se divide en tres categorías: se inicia con lo básico, posteriormente se avanza hacia lo fundamental y finalmente se aborda la estructura organizativa. También puede ser una opción viable cuando se requiere un marco adicional para convivir con otros estándares específicos de la industria, como *Health Insurance Portability and Accountability Act* (HIPAA) y NIST [12].

Esta organización se basa en puntos de referencia o pautas establecidas por estándares ampliamente utilizados, como NIST e HIPAA, con el objetivo de proporcionar un marco de seguridad que ayude a las empresas a cumplir con dichos estándares. Además, también ofrecen configuraciones de seguridad básicas alternativas para aquellas empresas que no necesariamente requieren cumplir con los estándares, pero desean mejorar su nivel de seguridad [12].

La mayoría de los puntos de referencia de CIS incluyen múltiples perfiles de configuración, tal como se describe a continuación [13].

- El perfil de Nivel 1 se considera una recomendación elemental que puede ser implementada de manera ágil y está diseñada para minimizar el impacto sobre el rendimiento. La intención de las recomendaciones de las pruebas de perfil de nivel 1 es minimizar la superficie de ataque de las organizaciones, pero también mantener las máquinas utilizables y evitar interferir con las actividades empresariales.
- El perfil de Nivel 2 se establece como "defensa en profundidad" y está diseñada para entornos en los que la seguridad es de máxima importancia. Las recomendaciones vinculadas al perfil de Nivel 2 pueden generar consecuencias negativas para la organización si no se aplican de manera adecuada o sin la debida precaución.

- El perfil *Security Technical Implementation Guide* (STIG) reemplaza al nivel 3 anterior, y ofrece todas las recomendaciones específicas de STIG. La superposición de recomendaciones de otros perfiles, es decir, Nivel 1 y 2, es evidente en el perfil STIG.

Cada recomendación contenida en cada CIS *Benchmark* se encuentra asociada, al menos, con un perfil. Se recomienda aplicar primero la guía CIS *Benchmark* en un entorno de prueba para determinar el impacto potencial, independientemente del perfil de nivel que se planee utilizar en cualquier entorno [13].

***Control Objetives for Information and Related Technology* (COBIT)**

COBIT es un marco de referencia ampliamente reconocido, se centra en el gobierno y la administración de las tecnologías de la información empresariales y orientadoa toda la Entidad. Ha sido impulsado por *Information Systems Audit and Control Association* (ISACA) desde su versión inicial en 1996 hasta la fecha actual, está disponible la versión COBIT 2019 para su aplicación [14].

En la versión inicial del marco de referencia, COBIT fue concebido como un acrónimo que representa Objetivos de Control para la Información y Tecnología Relacionada y su audiencia primaria en ese momento consistía en los auditores de TI. La versión actual tiene en cuenta diversos sectores interesados, no únicamente el rol de TI de una empresa, también contemplan otras partes interesadas como la dirección ejecutiva, junta directiva y la auditoría, etc. [14].

La TI empresarial abarca el conjunto de tecnologías y procesos de información que una organización emplea con el fin de alcanzar las metas, sin importar su ubicación dentro de la estructura empresarial. En otras palabras, la TI empresarial no se encuentra restringida únicamente al Departamento de TI de una organización [14].

2 METODOLOGÍA

En el presente proyecto se empleó la investigación documental con el objetivo de tener una base sólida de conocimientos teóricos sobre el *hardening* y la automatización con Ansible. De este modo se logró comprender como implementar de forma eficaz las medidas de seguridad en el sistema operativo de servidor Alma Linux.

En primer lugar, se procedió a instalar el sistema operativo Alma Linux, en un entorno virtual a través del hipervisor VirtualBox. La elección de VirtualBox como plataforma de virtualización se debió a su compatibilidad y su facilidad de uso. Una vez iniciada la

máquina virtual se procedió a instalar SCAP *Workbench,* la herramienta de escaneo que permite identificar las vulnerabilidades del sistema realizando un análisis inicial. El perfil de ciberseguridad en el que se basó el análisis es CIS nivel 1 para servidor, este proporciona un conjunto de recomendaciones para mejorar la seguridad del sistema. Esta etapa fue fundamental, ya que se realizó una evaluación preliminar del sistema antes de aplicar *hardening* con Ansible.

Posteriormente, se procedió a instalar el sistema operativo Fedora mediante el hipervisor VirtualBox. Fedora actuó como nodo de control, ya que aquí se instaló Ansible.

A continuación, para la implementación del *hardening*, se estableció la comunicación entre el nodo de control Fedora y el nodo controlado Alma Linux por medio del servidor SSH. En el sistema operativo Fedora, se creó primero el archivo de inventario, el cual alberga la información para establecer la conexión con el nodo a administrar. Después se creó y ejecutó el archivo del *playbook* que cuenta con las tareas de *hardening*. Las tareas del *playbook* fueron diseñadas para solventar 5 vulnerabilidades de forma dinámica y 5 vulnerabilidades de forma estática. Con el propósito de obtener un reporte final, después de haber ejecutado el *playbook,* en Alma Linux, se llevó a cabo otro análisis con SCAP *Workbench.*

Finalmente, para verificar los resultados obtenidos, se compararon los reportes de análisis inicial y final. Asimismo, se analizaron las vulnerabilidades solventadas con el *playbook,* con el fin de evaluar las medidas de seguridad implementadas.

3 RESULTADOS

El presente trabajo consistió en la Implementación de *hardening*, mediante Ansible, en un sistema operativo de servidor Linux de base *Red Hat*. Para cumplir este propósito, primero se efectuó la instalación del sistema operativo de servidor Alma Linux en VirtualBox. Se realizó la instalación de SCAP *Workbench*, la herramienta para identificar las vulnerabilidades por medio un análisis inicial.

Respecto al nodo de control, se utilizó el sistema operativo Fedora 38. Posterior a la instalación, se verificó la versión de Python y se instaló Ansible. Se crearon los archivos del inventario y *playbook*. Posteriormente, con Ansible, se aplicó *hardening* en el sistema operativo de servidor Alma Linux. Finalmente, se realizó otro análisis para obtener un reporte final, el cual fue comparado con el reporte inicial para determinar que la seguridad del sistema operativo de servidor ha mejorado.

3.1 Identificación de las vulnerabilidades sin políticas de seguridad

Instalación del sistema operativo de servidor

Antes de iniciar la instalación del sistema operativo de servidor Alma Linux, fue importante conocer algunos detalles sobre esta distribución de Linux de código abierto. Alma Linux ha sido creada como una opción gratuita y compatible con *Red Hat Enterprise* Linux (RHEL), ya que se basa en su código fuente. Esto significa que las aplicaciones y paquetes diseñados para RHEL deberían funcionar perfectamente en Alma Linux, sin complicaciones.

Además, antes de proceder con la instalación, fue igualmente esencial verificar las especificaciones de la máquina anfitriona en relación con los requisitos mínimos del sistema operativo, Alma Linux, que se va a instalar mediante una máquina virtual en VirtualBox. Este análisis fue esencial para garantizar un rendimiento óptimo durante la instalación y el funcionamiento posterior del sistema.

En la Tabla 3.1 se puede ver las características de la máquina anfitriona respecto a los requerimientos mínimos de Alma Linux. Referente al procesador de la máquina anfitriona se puede decir que garantiza un rendimiento superior al que se requiere, también cuenta con una memoria RAM de 16 (GB) la cual es suficiente y además tiene una capacidad de almacenamiento que excede los 20 (GB) recomendados por Alma Linux. Con ello se puede afirmar que la máquina anfitriona superó significativamente a los requisitos mínimos del sistema operativo.

Tabla 3.1 Características máquina anfitriona vs. requisitos mínimos Alma Linux [15]

Parámetro	**Máquina anfitriona (DESKTOP-ABSOOAH)**	**Requisitos mínimos de Alma Linux**
Procesador	12th Gen Intel(R) Core (TM) i7-1255U a 1.70 (GHz)	Es recomendable un procesador de 64 (bits) (x86_64) compatible con un mínimo de 1.5 (GHz) de velocidad de reloj.
Memoria RAM	16.0 (GB) → 15.6 (GB) utilizable	Es recomendable al menos 1.5 (GB) de RAM. Para servidores, es preferible disponer de 2 (GB) o más de RAM.

Parámetro	Máquina anfitriona (DESKTOP-ABSOOAH)	Requisitos mínimos de Alma Linux
Espacio en Disco	200 (GB)	Para una instalación básica, es necesario disponer de 10 (GB) de espacio en disco, 20 (GB) recomendado.

Como primer paso se descargó la imagen ISO de la página oficial de Alma Linux https://almalinux.org/get-almalinux/, una vez descargada la imagen ISO se procedió con la creación de la nueva máquina virtual a la cual se le asignó el nombre de “AlmaLinux 9.2”, se seleccionó la imagen ISO antes descargada y también se definió la ruta en donde esta se alojó, en la sección 1 de la Figura 3.1 se pueden observar estos pasos. En la misma figura en la sección 2 se puede visualizar también la cantidad de memoria RAM y el número de procesadores que se destinó.

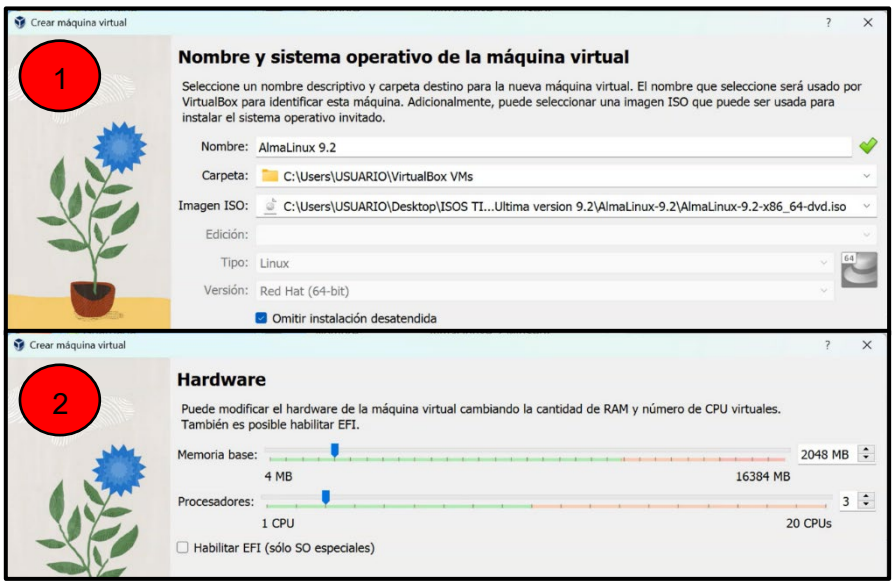

Figura 3.1 Asignación de nombre, RAM y procesadores para Alma Linux

Luego se definió el tamaño del disco duro virtual que en este caso fue de 40 (GB), esto se puede ver en la sección 1 de la Figura 3.2. Mientras que en la misma figura en la sección 2 se puede ver un resumen de todas las características de la máquina virtual a crear en donde se procedió a dar clic en “Terminar” para que esta sea implementada en VitualBox.

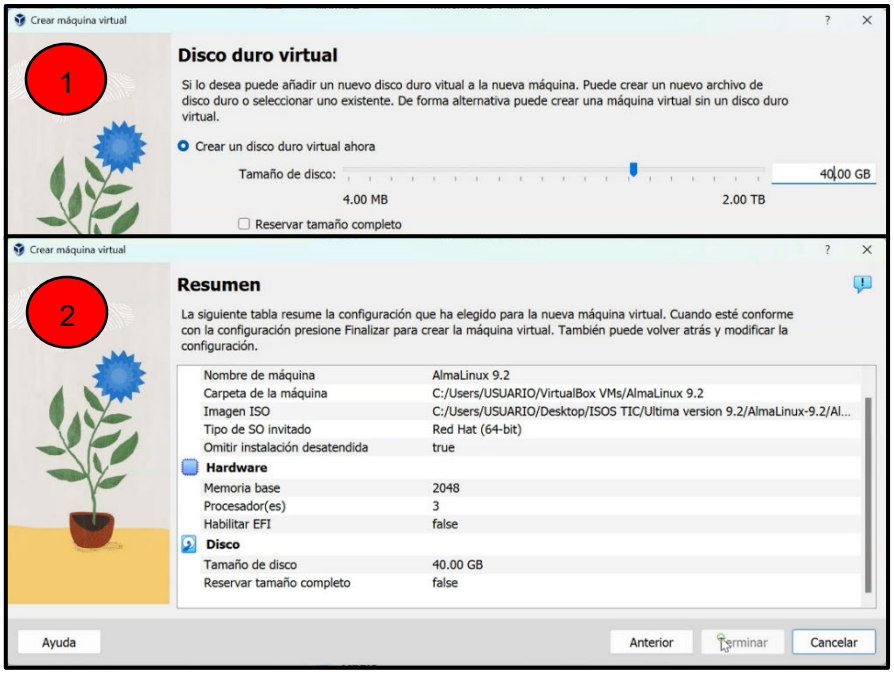

Figura 3.2 Resumen de los parámetros de configuración en Alma Linux

En la configuración de la máquina creada se realizó cambios en la pestaña" Avanzado" de los parámetros de "Portapapeles compartido" y "Arrastrar y soltar", colocando en ambas opciones bidireccional con el propósito de copiar texto de la máquina anfitriona y pegar en el sistema operativo Alma Linux de la máquina virtual, y viceversa. Así como también el de arrastrar carpetas o archivos desde un sistema a otro, estos cambios se pueden ver en la sección 1 de la Figura 3.3 y se realizaron con la finalidad de facilitar la transferencia de información entre ambos sistemas.

En la sección 2 de la Figura 3.3 también se puede visualizar el cambio que se realizó en la pestaña "Red" en el adaptador 1, seleccionando el adaptador de red tipo "NAT" con el propósito de tener acceso a Internet durante la instalación de Alma Linux para una posible descarga de paquetes y actualizaciones. Cabe recordar que con este tipo de adaptador la máquina virtual no es visible para otras máquinas en la red local.

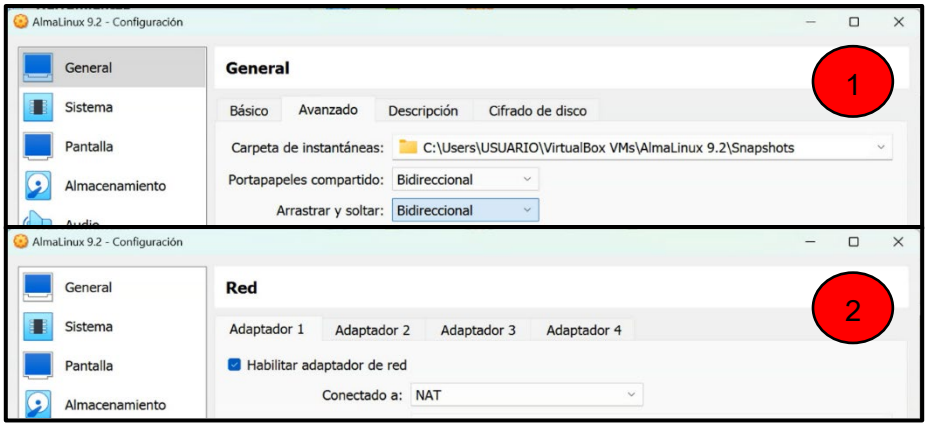

Figura 3.3 Configuraciones generales y de red en Alma Linux

Una vez que se inició la máquina virtual se seleccionó la opción de “Install AlmaLinux 9.2” para arrancar con su instalación, esto se puede ver en la sección 1 de la Figura 3.4, mientras que en la sección 2 de la misma figura se puede apreciar la selección del idioma para Alma Linux.

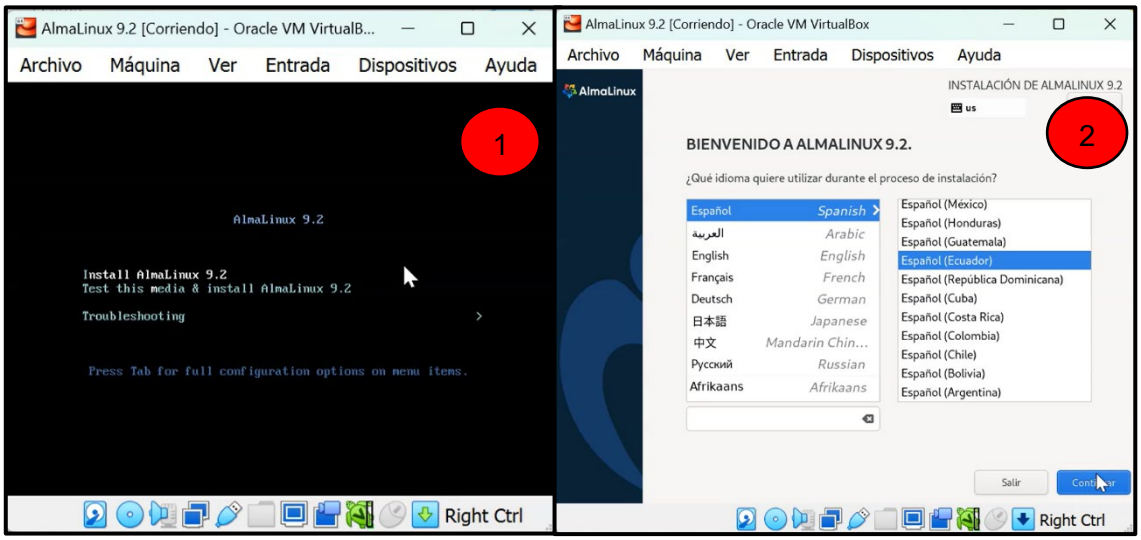

Figura 3.4 Arranque de la instalación y selección del idioma en Alma Linux

Al acceder al menú principal se realizaron las configuraciones del destino de la instalación, esto se puede ver en la sección 1 de la Figura 3.5. Ahí se seleccionó el disco duro virtual de 40 (GB) que antes ya se había creado, esto se puede visualizar en la sección 2 de la misma figura.

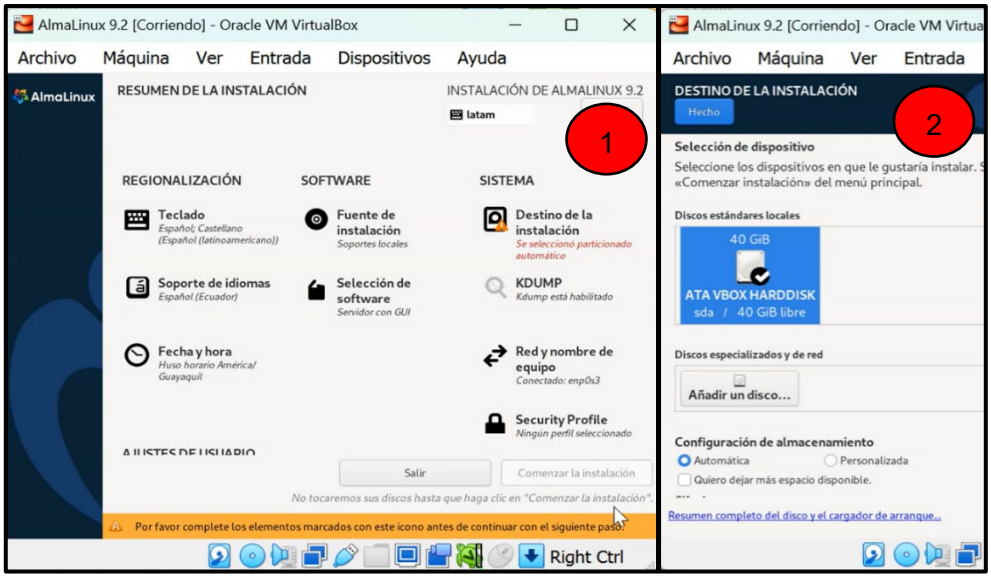

Figura 3.5 Menú principal de Alma Linux y selección del disco duro virtual

En la Figura 3.6 en la sección 1 se muestra la creación del usuario y contraseña, al cual también se le otorgaron los permisos de administrador. Realizado todo este proceso, se dio clic en “Comenzar la instalación” para que se lleven a cabo todas las configuraciones previamente establecidas y así finalmente reiniciar el sistema para completar el proceso de la instalación.

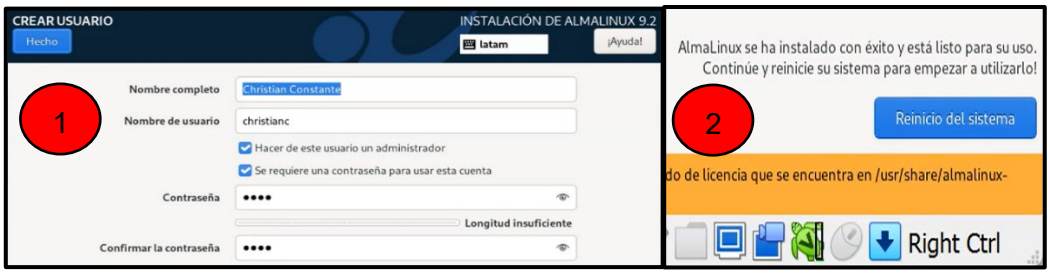

Figura 3.6 Creación del usuario y reinicio del sistema

Al iniciar la máquina virtual, se optó por revisar si el sistema operativo Alma Linux contaba con Python, ya que tener instalado Python es uno de los requisitos para que Ansible pueda administrar esta máquina por medio del nodo de control. Si hizo uso del comando “python --version” en la terminal, se logró ver que la versión instalada de Python es la 3.9.18.

Instalación de la herramienta de escaneo de configuración y vulnerabilidades SCAP *Workbench*

Se seleccionó la herramienta SCAP *Workbench* para realizar el análisis de configuración y la evaluación de vulnerabilidades. La elección de esta herramienta se debe principalmente a su interfaz gráfica fácil de entender y su capacidad para adaptarse a

una variedad de entornos y sistemas operativos. Además, SCAP *Work*bench ofrece una variedad de perfiles de referencia en ciberseguridad según el nivel de seguridad deseado, lo que permite la personalización de las evaluaciones de seguridad según estándares, normativas y mejores prácticas.

Para llevar a cabo la instalación de la herramienta de escaneo se accedió al sistema operativo de servidor como administrador. En la Figura 3.7 se puede observar el comando 1 el cual sirve para instalar el paquete “scap-workbench” por medio de “dnf” la cual es una herramienta muy útil en cuanto a la gestión de paquetes en sistemas operativos que toman como base el código fuente de *Red Hat*.

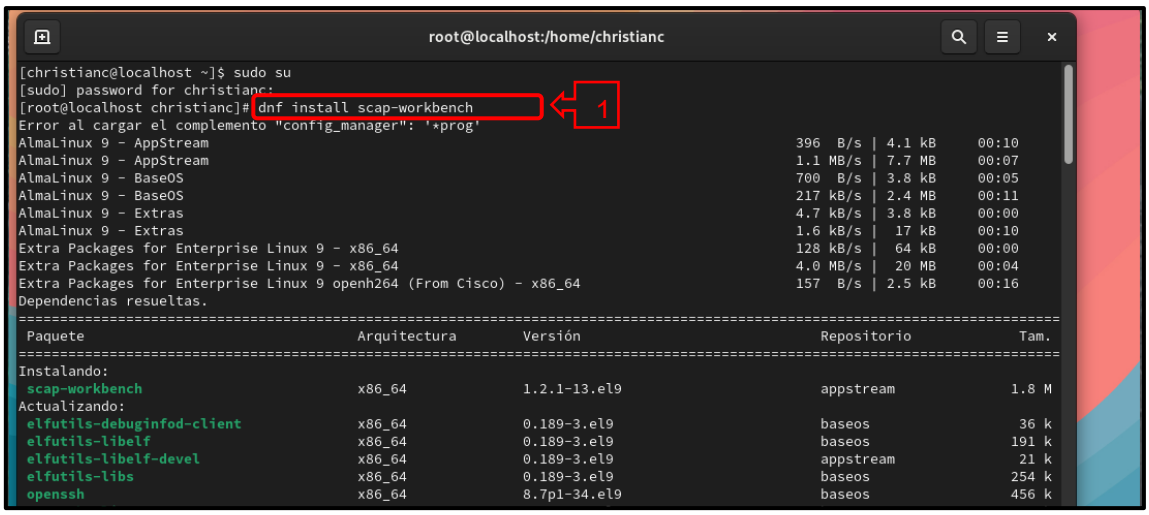

Figura 3.7 Comando para instalar el paquete “scap-workbench”

Para finalizar este proceso se ejecutó el comando 1 de la Figura 3.8, este comando sirve para instalar el paquete “scap-security-guide” que es la guía de seguridad SCAP para sistemas basados en *Red Hat*, el cual proporciona *checklists* o listas de verificación que contienen las configuraciones recomendadas de seguridad para evaluar y fortalecer el sistema operativo según el perfil que se esté aplicando.

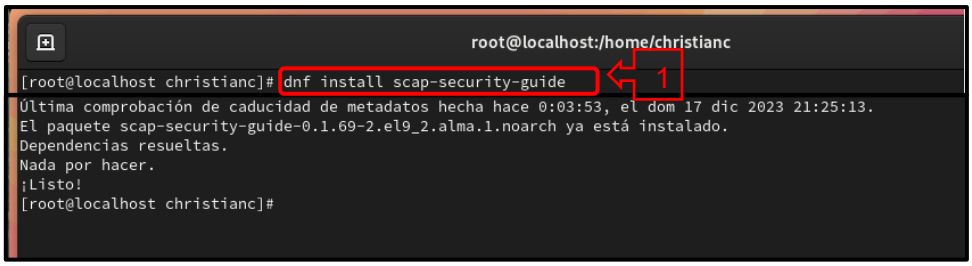

Figura 3.8 Comando para la instalación de la guía de seguridad SCAP

Obtención del reporte inicial por medio de SCAP *Workbench*

Para obtener el reporte inicial se ejecutó SCAP *Workbench* desde el menú de aplicaciones en el escritorio. Una vez ingresado a la herramienta aparece un cuadro de diálogo en el cual se seleccionó la guía de seguridad para el sistema operativo que en este caso es Alma Linux, finalmente se seleccionó la opción de cargar contenido para pasar a la siguiente ventana. En la Figura 3.9 se puede visualizar todo este proceso.

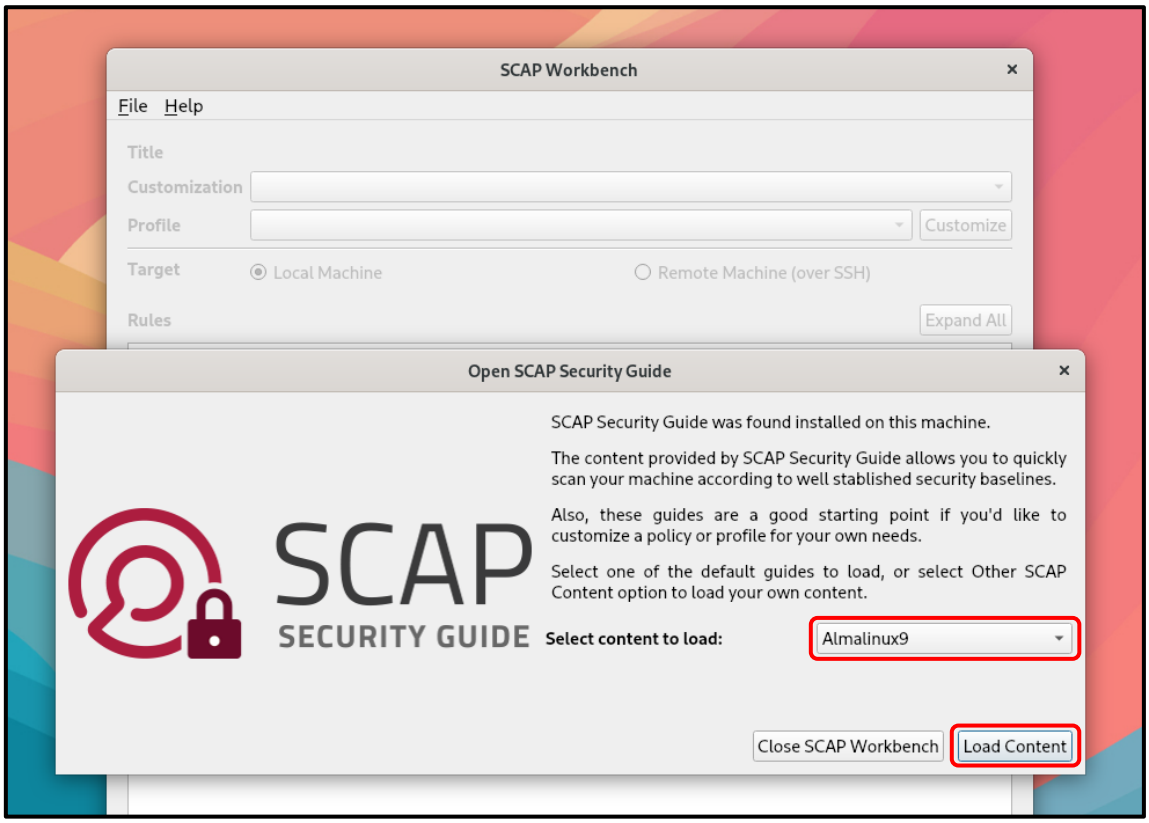

Figura 3.9 Selección del sistema operativo y carga de la guía de seguridad

Al acceder en esta ventana se realizaron las configuraciones para la selección del perfil de seguridad en el cual se basará el análisis, este proceso se puede ver en la sección 1 de la Figura 3.10. Ahí se seleccionó el perfil fundamentado en el Marco de referencia “CIS nivel 1” para servidor, este proporciona un conjunto de recomendaciones para mejorar la seguridad del sistema. Para empezar el análisis se dio clic en “Scan” y se confirmó la contraseña del usuario del sistema, esto se puede ver en la sección 2 de la misma figura.

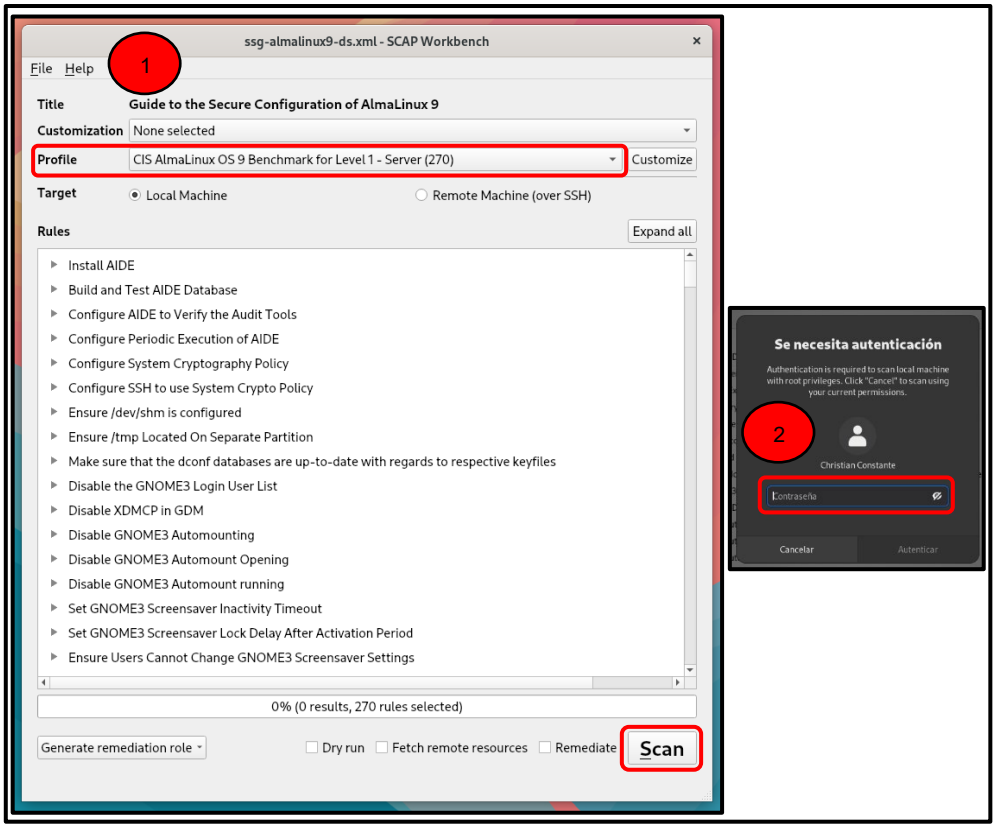

Figura 3.10 Arranque del análisis y confirmación de contraseña

Una vez terminado el análisis para ver los resultados se procedió a cerrar la ventana que apareció seleccionando “close”, esto se puede apreciar en la sección 1 de la Figura 3.11 y se dio clic en la opción “Show Report” para mostrar el reporte como se puede ver en la sección 2 de la misma figura.

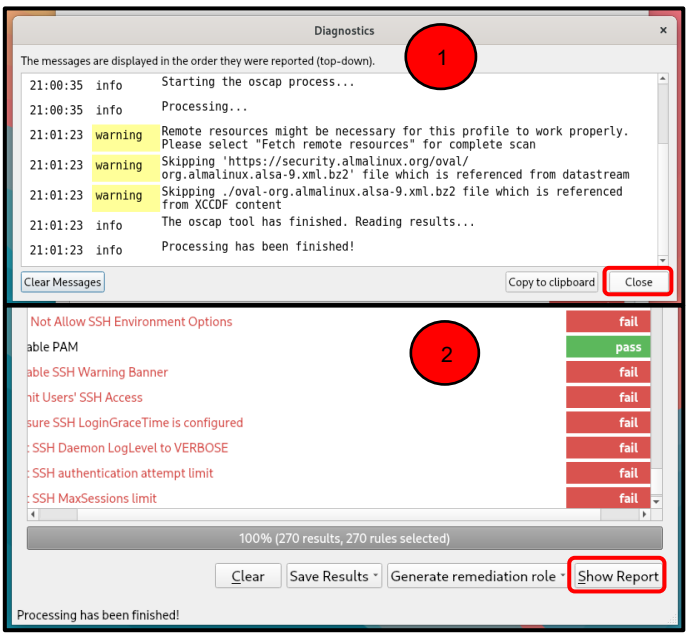

Figura 3.11 Selección de la opción “Show Report”

Realizados los pasos anteriormente descritos ya fue posible visualizar el reporte en el navegador. Observando los resultados que se pueden ver e la Figura 3.12, se evaluaron un total de 252 reglas, de las cuales 143 pasaron, es decir cumplieron con el perfil, 108 fallaron y 1 tuvo otro resultado. Las reglas que se clasifican como “other” generalmente es porque las reglas no se pueden clasificar debido a que la interpretación de la regla no coincide con las especificaciones de las configuraciones del sistema operativo.

Respecto a la severidad de las 108 reglas que fallaron, estas fueron clasificadas por la herramienta de acuerdo con los niveles que se muestran a continuación:

- 5 reglas se clasifican como "other" (otro).
- 6 reglas se clasifican como "low" (bajo).
- 93 reglas fueron clasificadas como "medium" (medio).
- 4 reglas fueron clasificadas como "high" (alto).

En cuanto a la puntuación general del cumplimiento, se puede ver que la puntuación general del sistema es del 70.88 % de reglas cumplidas del total posible que es el 100%.

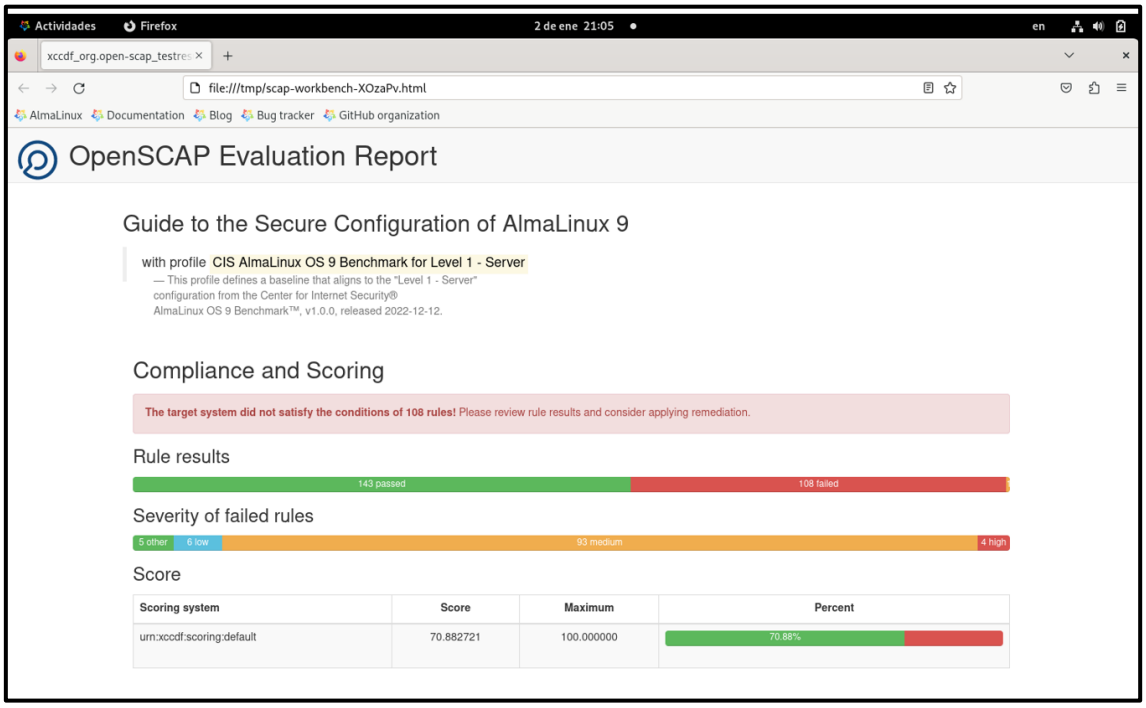

Figura 3.12 Resultados del reporte inicial

En la Figura 3.13 se detalla las reglas fallidas que fueron seleccionadas para su análisis, tomando como base la recomendación de la solución planteada por SCAP *Workbench, con* el fin de solucionarlas a través Ansible con un *playbook*. Como se puede visualizar en la figura se han seleccionado 2 reglas de nivel alto y 8 reglas de nivel medio.

Title	Severity	Result
Disable XDMCP in GDM	high	fail
Disable SSH Access via Empty Passwords	high	fail
Disable Kernel Parameter for Accepting Secure ICMP Redirects on all IPv4 Interfaces	medium	fail
Set Account Expiration Following Inactivity	medium	fail
Set SSH authentication attempt limit	medium	fail
Configure Firewalld to Trust Loopback Traffic	medium	fail
Set SSH MaxSessions limit	medium	fail
Set Password Maximum Age	medium	fail
Install AIDE	medium	fail
Uninstall avahi Server Package	medium	fail

Figura 3.13 Reglas seleccionadas para implementación en el *playbook*

3.2 Instalación de la herramienta Ansible en el nodo de control

Instalación del sistema operativo del nodo de control

El sistema operativo Fedora actúa como el nodo de control, ya que en este sistema se instala Ansible, el cual se encarga de la ejecución de las tareas de *hardening* a realizarse por medio del *playbook*. Desde este sistema operativo se administra el sistema operativo de servidor Alma Linux, el nodo controlado.

Entre los detalles importantes de Fedora se puede resaltar que es una distribución de Linux de código abierto y gratuita. El proyecto Fedora está patrocinado de forma oficial por RHEL, lo que lo convierte en un entorno de desarrollo para las tecnologías que se van integrando a RHEL. Debido a las similitudes que las dos distribuciones comparten, las aplicaciones y paquetes diseñados para RHEL son compatibles con Fedora.

Para iniciar con la instalación de este sistema operativo se descargó la imagen ISO de la página oficial de Fedora https://fedoraproject.org/es/workstation/download, descargada la imagen ISO se creó la nueva máquina virtual a la cual se la llamó “Fedora 38”. En la sección 1 de la Figura 3.14 se puede visualizar la ruta en donde se alojó la imagen ISO, mientras que en la sección 2 de la misma figura se puede observar la cantidad de memoria RAM y el número de procesadores.

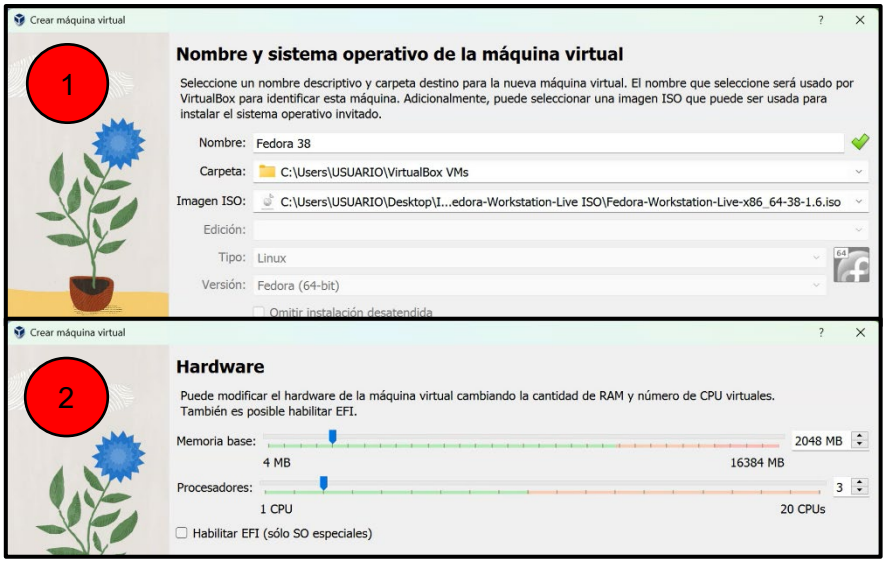

Figura 3.14 Asignación de nombre, RAM y procesadores para Fedora

Posteriormente se definió el tamaño del disco duro virtual, los 30 (GB) asignados se puede ver en la sección 1 de la Figura 3.15. Un resumen de todas las características de la máquina virtual Fedora se pueden ver en la sección 2 de la misma figura, finalmente se dio clic en “Terminar” para que esta sea integrada a VirtualBox.

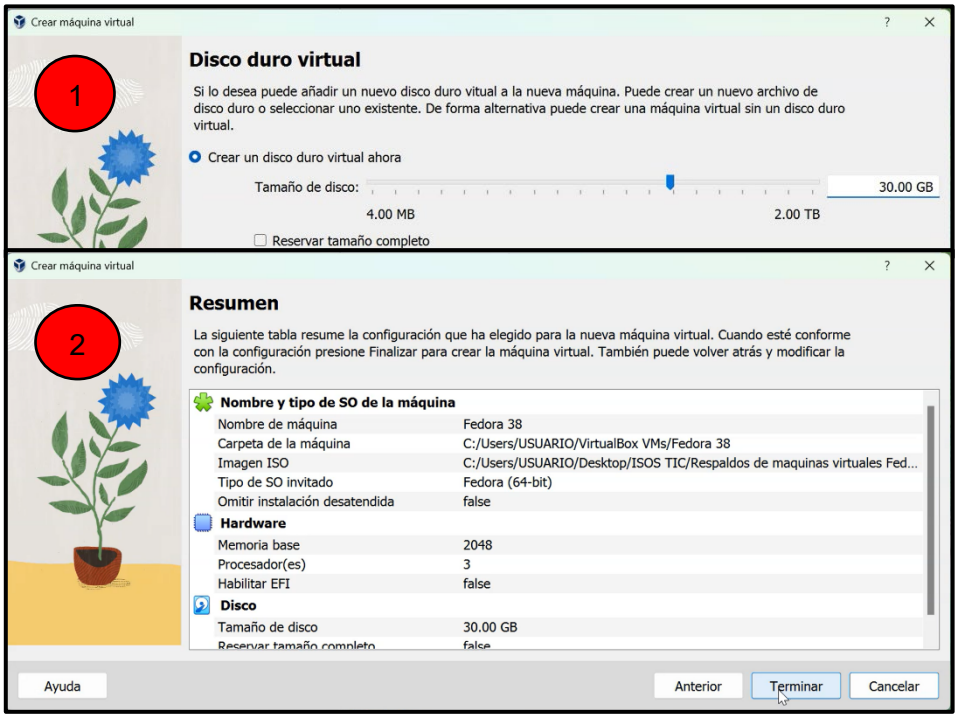

Figura 3.15 Resumen de los parámetros de configuración en Fedora

Respecto a la configuración de la máquina creada se realizó modificaciones en la pestaña de los parámetros: “Portapapeles compartido” y “Arrastrar y soltar”, seleccionando en ambas opciones “Bidireccional”. Esta modificación se realizó aquí al

igual que en la instalación del sistema operativo Alma Linux, para facilitar la transferencia de información entre ambos sistemas y la máquina anfitriona. Las modificaciones se pueden ver en la sección 1 de la Figura 3.16.

En la sección 2 de la Figura 3.16 se puede visualizar la modificación en la pestaña “Red” en el adaptador 1, se seleccionó el adaptador de red tipo “NAT” con la finalidad de tener acceso a Internet para una eventual descarga de paquetes y actualizaciones durante la instalación.

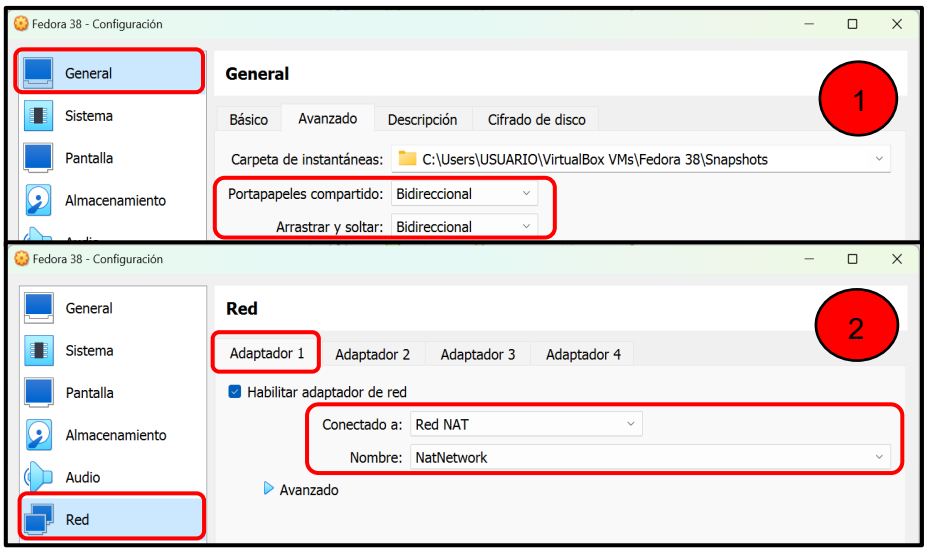

Figura 3.16 Configuraciones generales y de red en Fedora

Al iniciar la máquina virtual se seleccionó la opción de “Install Fedora” para arrancar la instalación, en la sección 1 de la Figura 3.17 se visualiza este paso, mientras que en la sección 2 de la misma figura se puede ver la selección del idioma español.

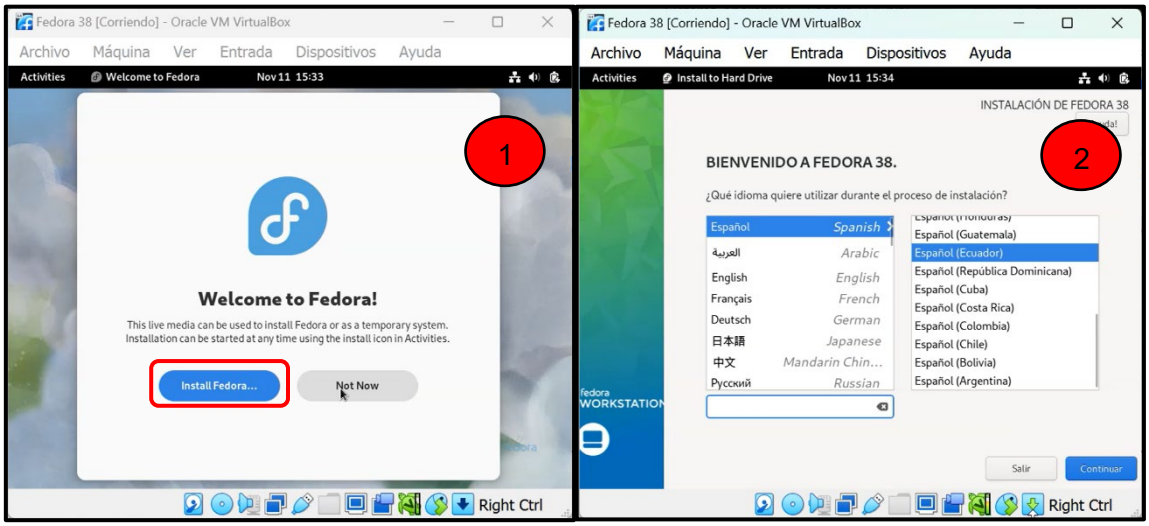

Figura 3.17 Inicio de la instalación y selección del idioma en Fedora

Luego se accedió al menú principal para configurar el destino de la instalación, esto se puede apreciar en la sección 1 de la Figura 3.18. En la sección 2 de la misma figura se puede ver la selección del disco duro virtual de 30 (GB).

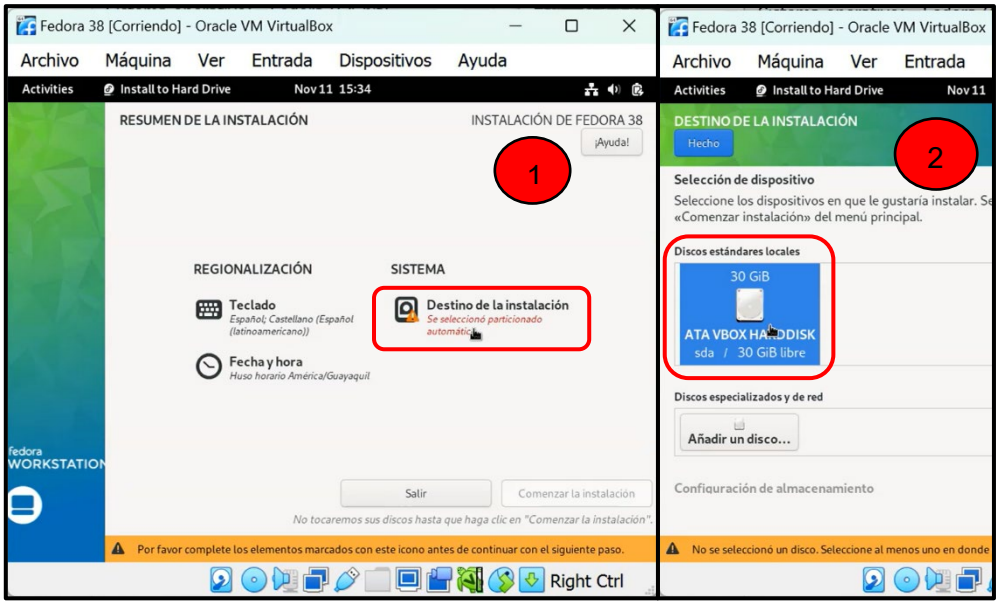

Figura 3.18 Menú principal de Fedora y selección del disco duro virtual

En la Figura 3.19 se muestra el proceso de carga de todas las configuraciones previamente establecidas al cual se accedió dando clic en “Comenzar la instalación”. Una vez terminado todo el proceso se reinició la máquina virtual para tener acceso a la pantalla de bienvenida del sistema y finalizar la instalación con la creación del usuario y contraseña.

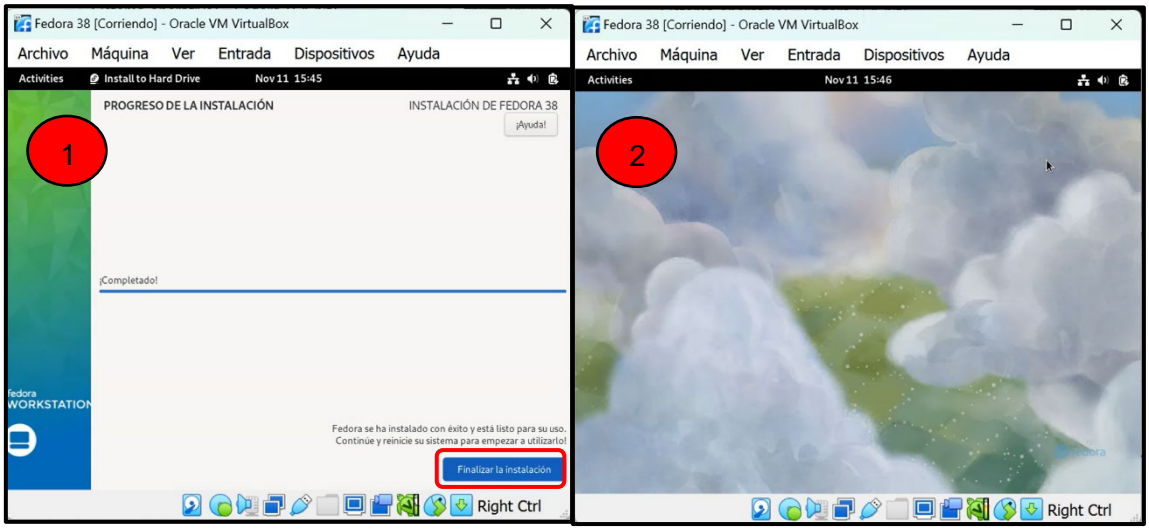

Figura 3.19 Proceso de carga de configuraciones y reinicio del sistema

Al iniciar la máquina virtual, se revisó si el sistema operativo Fedora contaba con Python, tener instalado Python es un requisito indispensable para la máquina en la que se instala

Ansible. Con el comando “python --version” que se ejecutó en la terminal se pudo ver la versión 3.11.6 instalada en el sistema operativo.

Instalación de la herramienta de DevOps Ansible

Ansible se instala solo en el nodo de control ya que desde este nodo se administran los nodos controlados. En el nodo de control se configura el inventario, se ejecuta el *playbook* y se gestiona toda la configuración de Ansible.

Al ser un *software* multiplataforma se puede instalar en todas las distribuciones de Linux, los requisitos para instalar Ansible en el nodo de control es tener instalado Python 2 en su versión 2.7 o Python 3 en sus versiones 3.5 en adelante. En los nodos controlados también se requiere de Python para que Ansible funcione correctamente debido a que los módulos que son utilizados para realizar tareas están escritos en Python, por esta razón Python es el intérprete en los nodos controlados al momento de ejecutar los módulos.

Para instalar la herramienta Ansible en Fedora se ejecutó el comando 1 con los privilegios de superusuario (sudo) que se observa en la Figura 3.20.

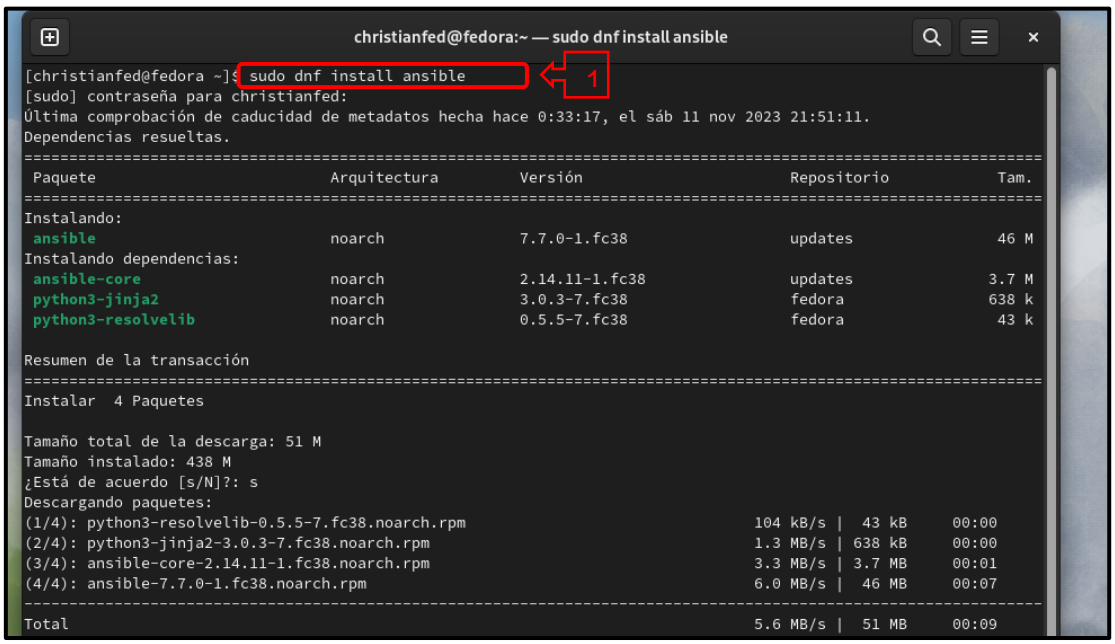

Figura 3.20 Comando para instalar el paquete “Ansible”

Para finalizar este proceso se ejecutó el comando 1 de la Figura 3.21, este comando sirve para verificar que Ansible se haya instalado de manera correcta. En esta figura se puede observar la versión de Ansible que se instaló en el sistema la cual es “ansible [core 2.14.11]”.

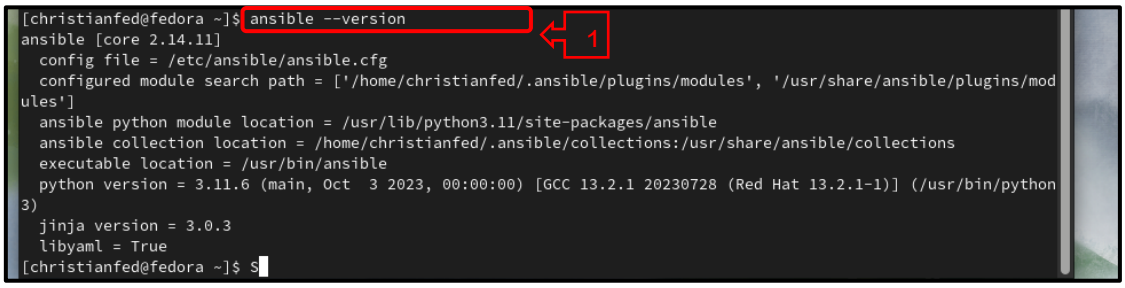

Figura 3.21 Comando para visualizar la versión de Ansible

3.3 Implementación del *playbook* para aplicar *hardening* al sistema operativo de servidor

Interconexión entre el sistema operativo de servidor y nodo de control

Se realizó cambios en el adaptador de red de las dos máquinas virtuales, tanto de la máquina Fedora con Ansible como de la máquina Alma Linux, durante el proceso de instalación se usó el adaptador de red tipo “NAT” el cual simula una red interna, las máquinas de la red obtienen una misma dirección IP reservada por VirtualBox. Se obtiene acceso a Internet, no obstante, la máquina virtual no es visible para las otras máquinas en la red del *host*.

La configuración de red que se usó para la interconexión de las máquinas virtuales es “Red NAT”, con esta opción es posible comunicar ambas máquinas ya que están en la misma red, además tener acceso a Internet. En este modo es posible activar la opción de DHCP para que además de ser una red interna, las direcciones de las máquinas se asignen dinámicamente.

Para activar esta configuración de red en las máquinas virtuales, se seleccionó la opción de “Archivo”, luego en “Herramientas y por último en “Administrador de red”, estas selecciones se pueden ver en la sección 1 de la Figura 3.22. Para crear la nueva red se seleccionó la opción de “Crear”, en la sección 2 de la misma figura se puede observar la nueva red 10.0.2.0/24 y el servidor DHCP habilitado.

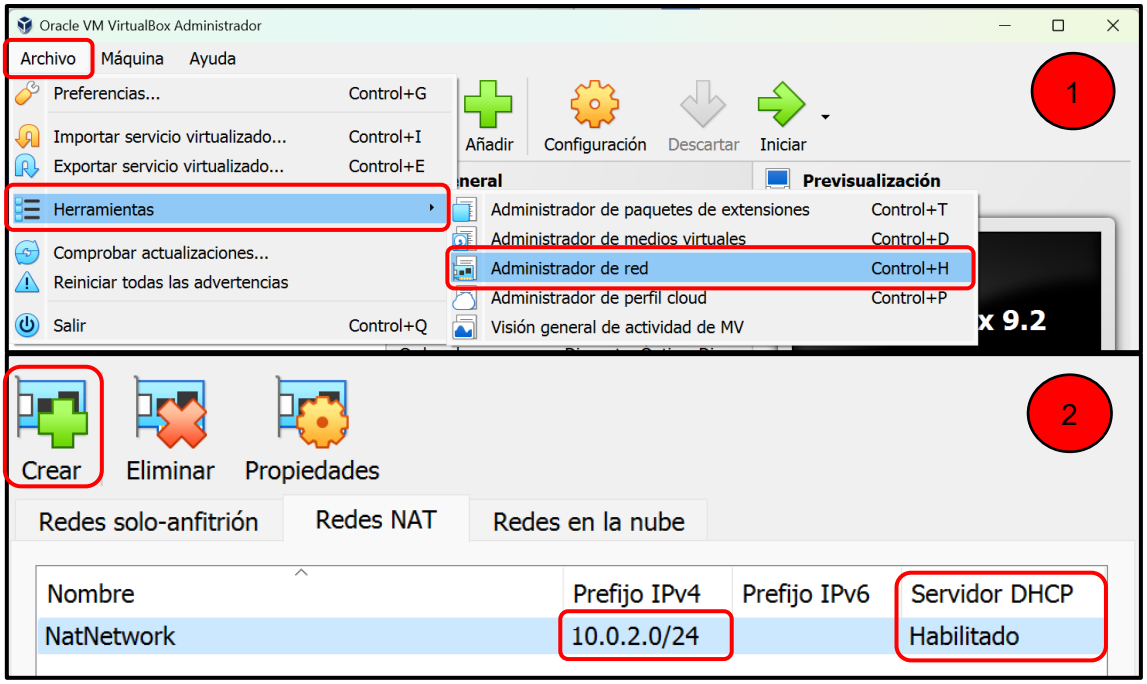

Figura 3.22 Configuraciones de red, habilitación de DHCP en redes NAT

Para aplicar esta configuración se entró en la configuración de las dos máquinas virtuales, específicamente en la pestaña “Red” en el adaptador 1. En la Figura 3.23 se puede ver activa la opción “Red NAT”.

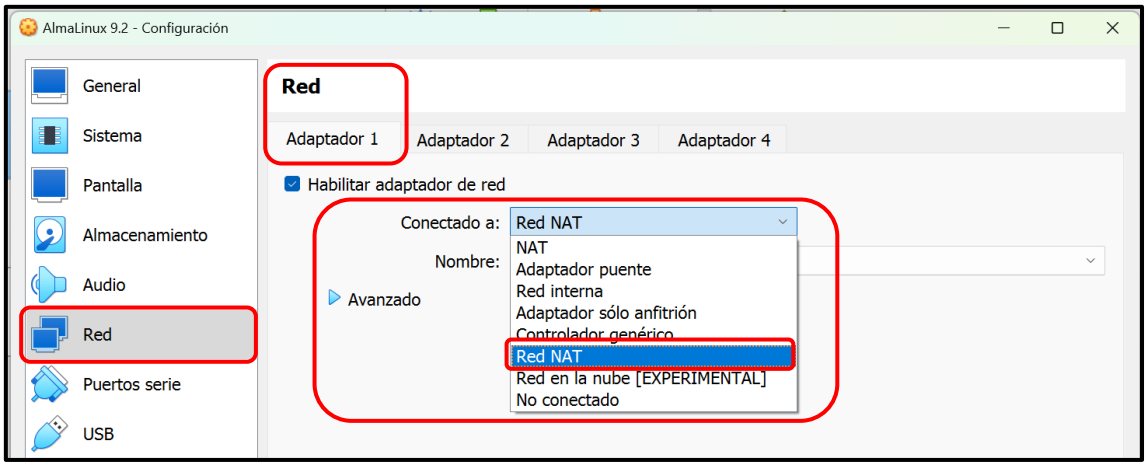

Figura 3.23 Configuración de adaptador 1, “Red NAT”

En la Figura 3.24 se puede observar las pruebas de comunicación entre las dos máquinas virtuales, así como el acceso a Internet. En la misma figura también se pueden visualizar las direcciones asignadas por medio del servidor DHCP las cuales son: 10.0.2.15 para el usuario “christianc” de la máquina con el sistema operativo de servidor Alma Linux y 10.0.2.4 para el usuario “christianfed” del sistema operativo que sirve como nodo de control Fedora.

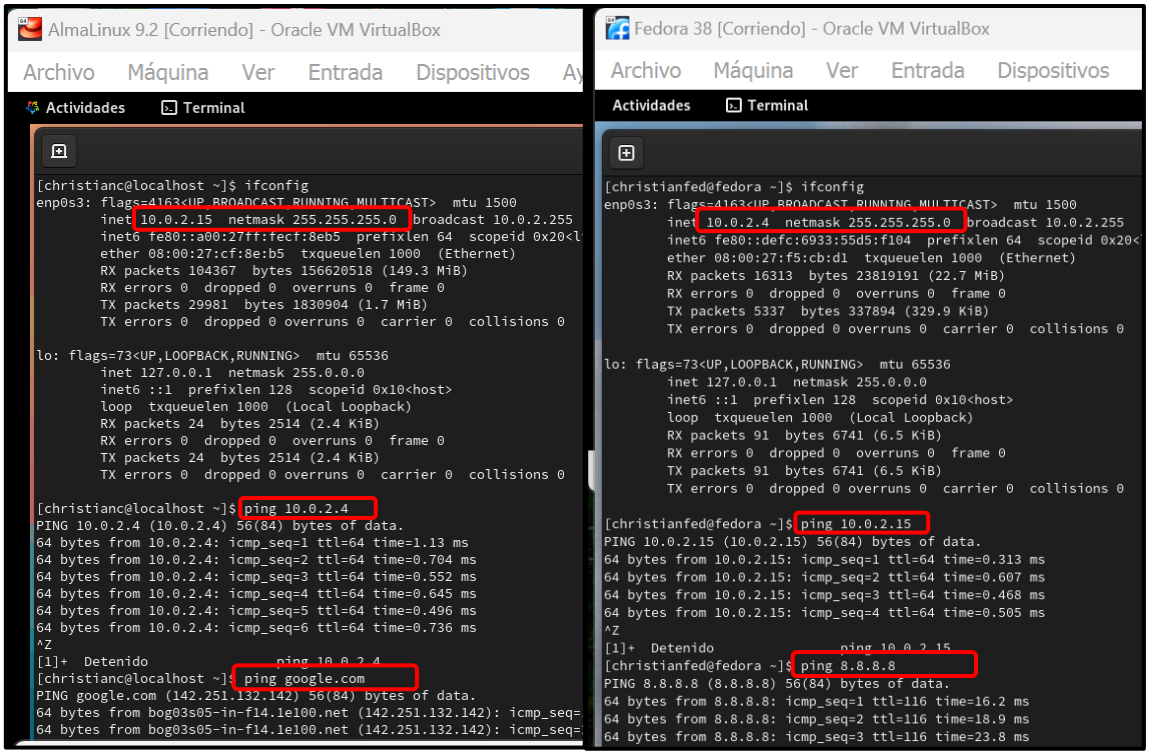

Figura 3.24 Pruebas de comunicación, máquinas virtuales

Comunicación SSH entre el nodo de control Fedora y el nodo controlado Alma Linux

Ansible emplea el protocolo SSH como método predeterminado para establecer conexiones seguras entre el nodo de control y el nodo controlado. SSH garantiza la transmisión de información cifrada, asegurando que los datos intercambiados entre el nodo de control y el nodo controlado estén protegidos contra cualquier amenaza de monitoreo no autorizado.

Para realizar la administración remota, es necesario tener acceso SSH al nodo controlado desde el nodo de control. Este procedimiento consiste en establecer claves SSH, donde la clave privada se guarda en el nodo de control y la clave pública se copia en el nodo controlado que se va a administrar. La integración adecuada de esta clave pública en el archivo “authorized_keys” del nodo controlado es esencial para proporcionar un flujo de información seguro y autorizado durante las operaciones de administración remota.

Instalación de SSH en el nodo controlado

Para instalar el servicio de SSH en el sistema operativo Alma Linux, primero se verificó que el sistema este actualizado ejecutando el comando 1 indicado en la Figura 3.25. En

la misma figura, se puede observar la instalación del paquete “openssh-server” realizada por medio del comando 2 y para iniciar el servicio SSH se ejecutó el comando 3.

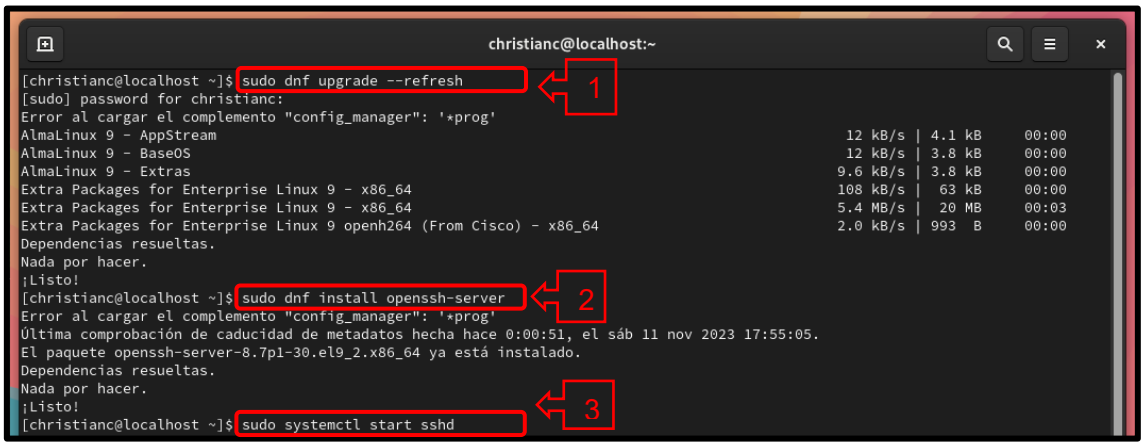

Figura 3.25 Proceso para iniciar el servidor SSH en Alma Linux

En la Figura 3.26, se puede visualizar la ejecución del comando 1, el cual sirve para garantizar que el servicio se inicie automáticamente en el arranque del sistema. Por último, para verificar el estado del servicio se ejecutó el comando 2 presente en la misma figura; cómo se puede ver el servicio SSH se encuentra activo.

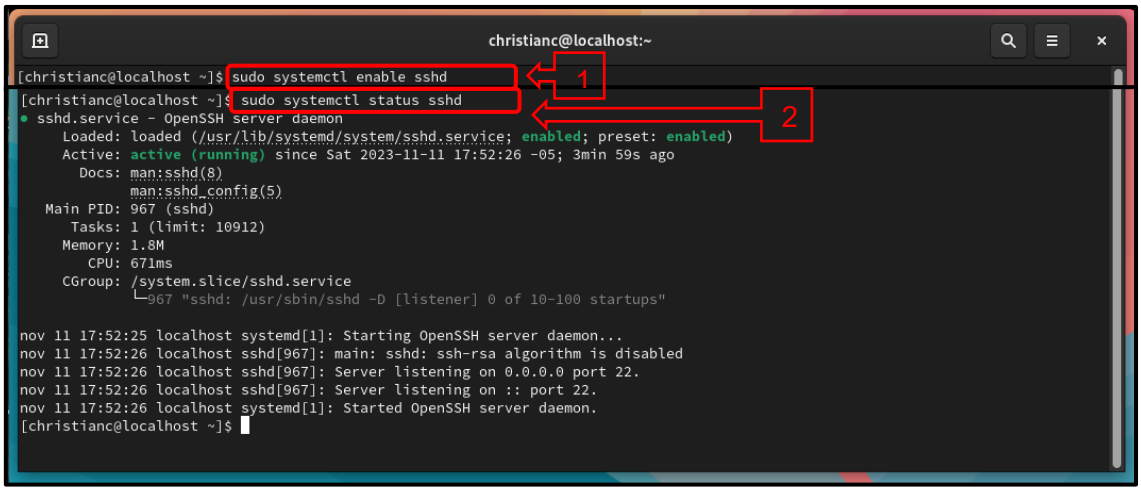

Figura 3.26 Estado del servidor SSH en Alma Linux

Instalación de SSH con autenticación de clave pública en el nodo de control

Respecto a la instalación del servicio SSH en el sistema operativo Fedora, en la Figura 3.27 se puede visualizar la actualización de todos los paquetes instalados en el sistema operativo a las versiones más recientes, la instalación del paquete “openssh-server” y el arranque del servicio SSH por medio de los comandos 1, 2, 3 respectivamente.

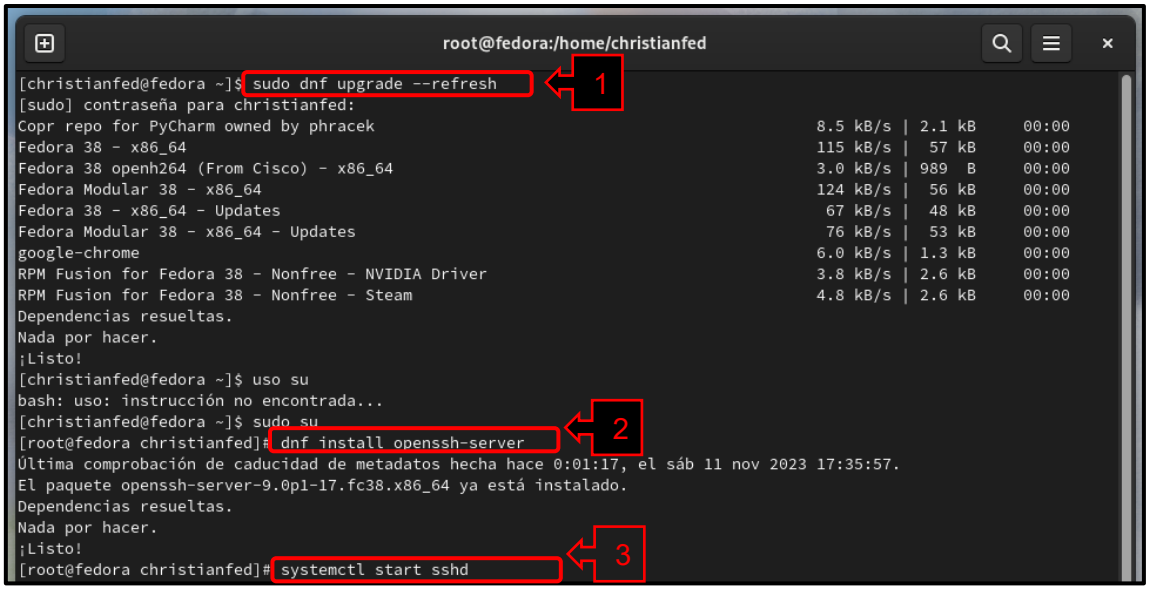

Figura 3.27 Proceso para iniciar el servidor SSH en Fedora

Para garantizar que el servicio arranque automáticamente, en el inicio del sistema se ejecutó el comando 1 de la Figura 3.28. En la misma figura se puede ver el comando 2 para comprobar el estado del servicio, como se puede ver el servicio SSH en Fedora está activo.

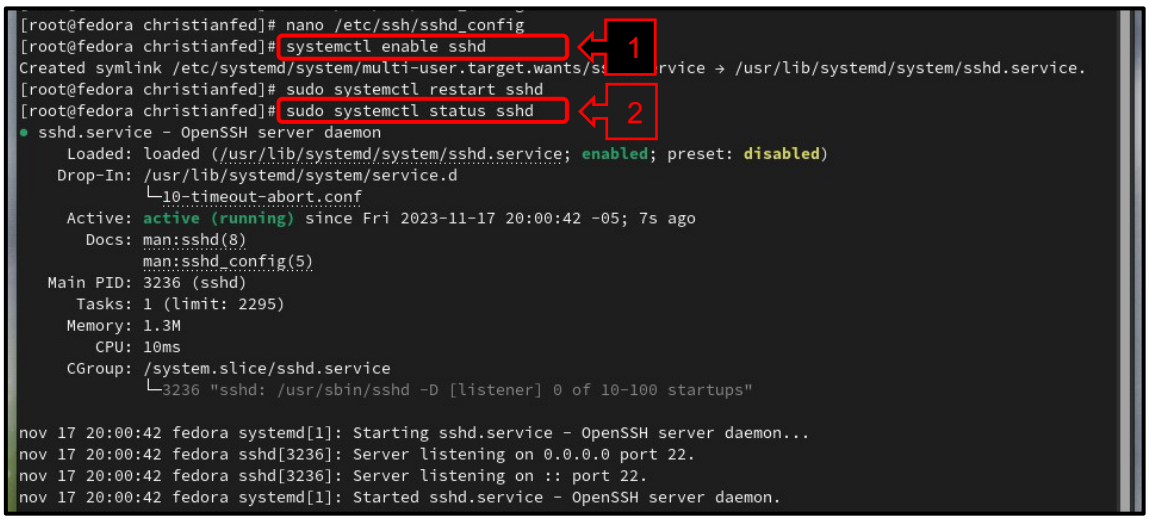

Figura 3.28 Estado del servidor SSH en Fedora

En la sección 1 de la Figura 3.29 se puede visualizar el proceso de conexión SSH de Alma Linux a Fedora mientras que en la sección 2 de la misma figura se puede observar el proceso de conexión SSH de Fedora a Alma Linux. Para realizar la conexión se utilizó el comando 1 y 2 presentes en las secciones 1 y 2 de la misma figura respectivamente, estos dos comandos siguen la misma estructura: ssh usuario@ip_deldestino, en dónde “usuario” se reemplaza por el nombre del usuario del sistema operativo al que se intenta ingresar e “ip_deldestino” por su dirección IP.

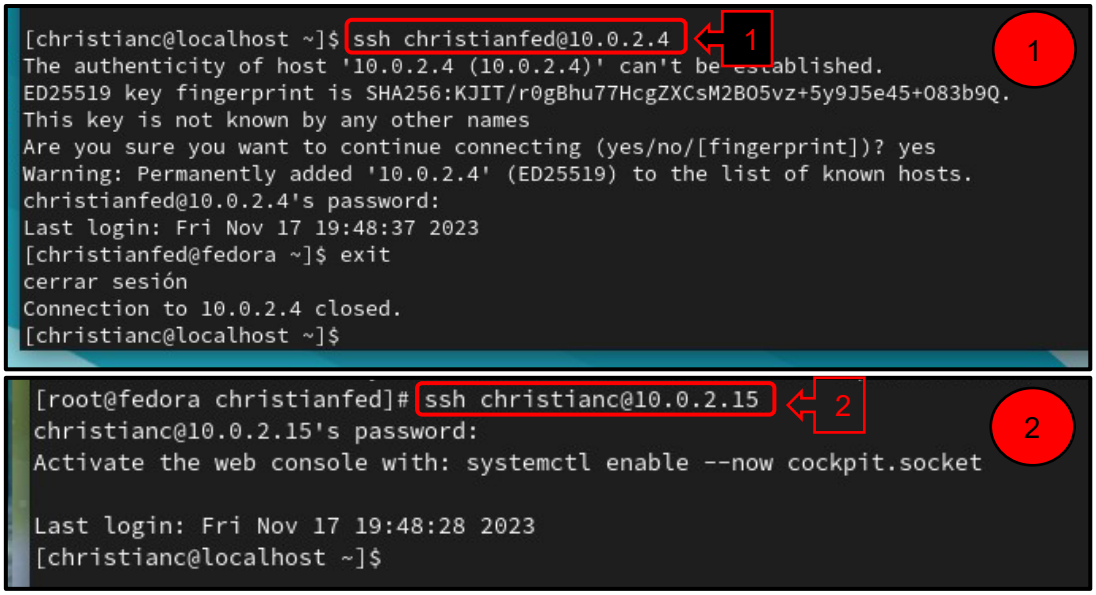

Figura 3.29 Prueba de conexión SSH bidireccional

Con los pasos anteriormente mostrados desde la instalación del servicio SSH en Alma Linux y Fedora hasta que se puede establecer la conexión en ambas direcciones, se puede comprobar que la instalación del servicio SSH ha sido exitosa; es necesario ingresar la contraseña en cada inicio de sesión de un sistema a otro. Este proceso de conexión SSH se denomina autenticación por contraseña.

A continuación, se detalla el proceso de conexión por clave pública desde Fedora con el cual no se requiere ingresar la contraseña del usuario del Alma Linux cada vez que se inicie la conexión. Ansible no exige el empleo de claves públicas para la conexión SSH, pero es una práctica recomendada debido a seguridad y automatización. Las claves públicas son mucho más difíciles de ser adivinadas o comprometidas en comparación con las contraseñas de usuario.

Para llevar a cabo este proceso, primero se editó el archivo en la ruta “etc/ssh/sshd_config” por medio del comando 1 que se observa en la sección 1 de la Figura 3.30. En la misma figura en la sección 2 se puede visualizar la línea “PubkeyAuthentication yes” que se modificó, la cual sirve para permitir la autenticación mediante claves públicas dado que está asignada con “yes”.

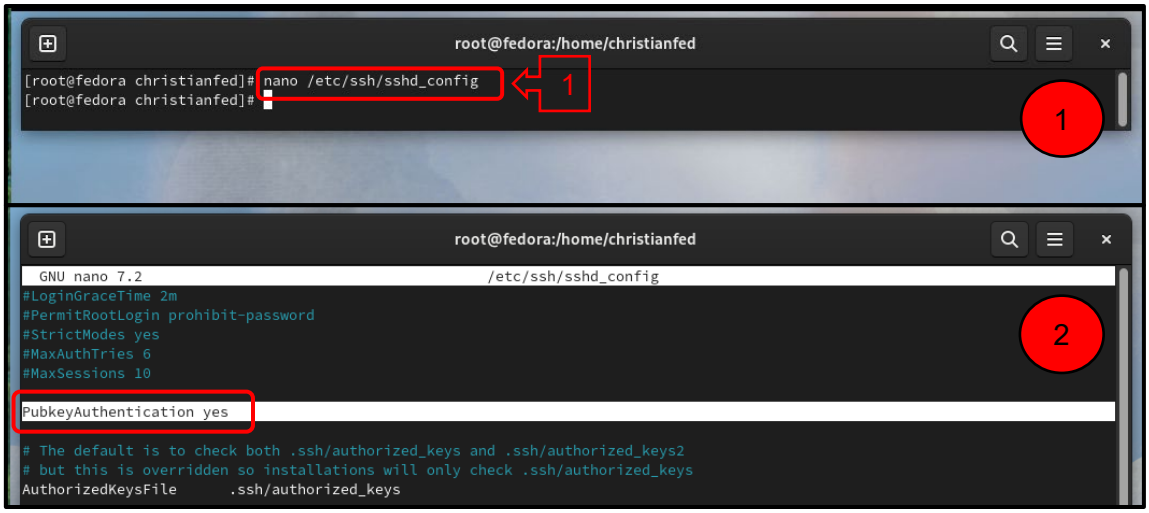

Figura 3.30 Edición en la configuración de SSH en Fedora

En la Figura 3.31 se puede ver el comando 1, el cual sirve para reiniciar el servicio SSH, con el propósito que los cambios realizados anteriormente tengan efecto. Con el comando 2 que se muestra en la misma figura se generó un par de claves SSH en Fedora, una clave privada y otra pública, se dio “Enter” para confirmar. Por este motivo se generan dos archivos en el directorio predeterminado de configuraciones de SSH. En la misma ejecución del comando apareció la opción para establecer una frase de contraseña, pero se omitió dando “Enter” y se confirmó de la misma manera.

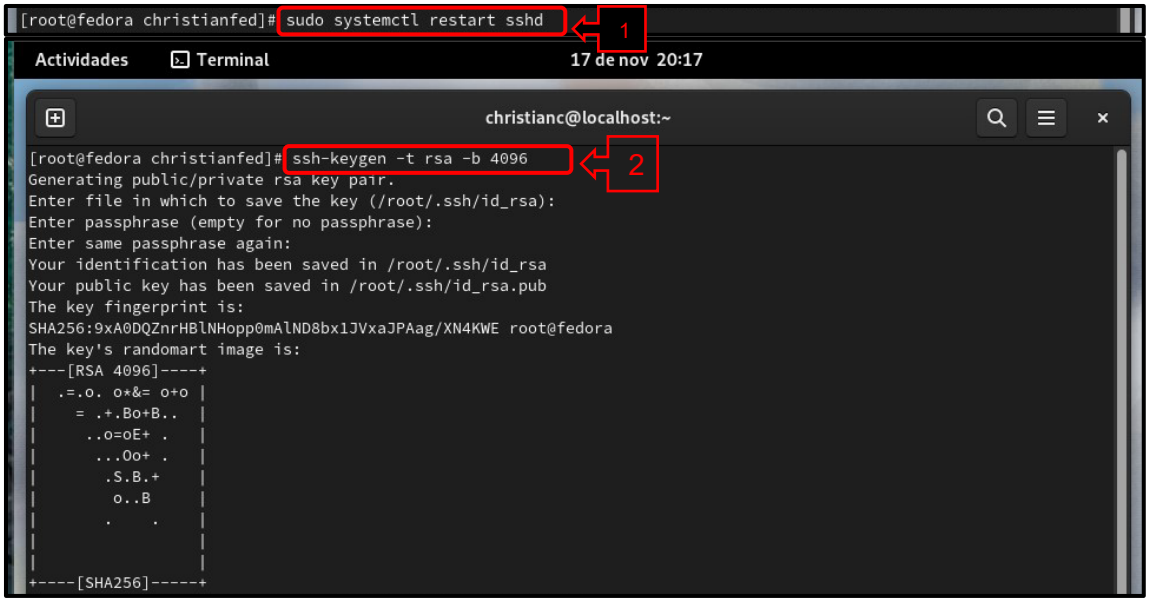

Figura 3.31 Comando para generar las claves pública y privada de Fedora

Con el comando 1 de la Figura 3.32 se copió la calve pública de Fedora a Alma Linux, para que sea posible el acceso sin contraseña. A continuación, el sistema solicitó

ingresar la contraseña del usuario de Alma Linux “christianc” para confirmar. Finalmente se puede ver en el recuadro amarillo de la misma figura que se ha añadido satisfactoriamente la clave pública.

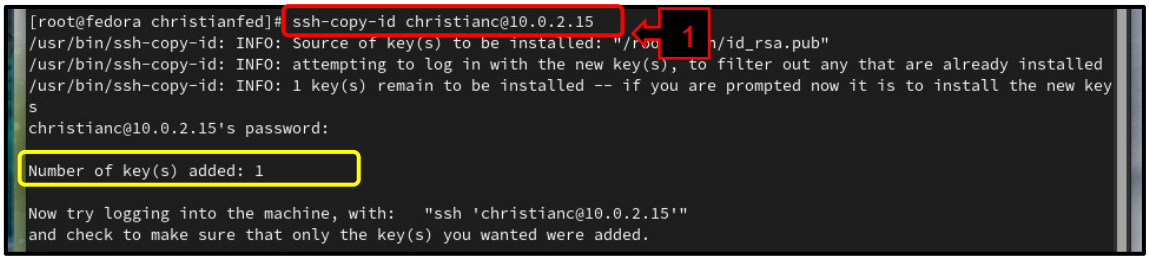

Figura 3.32 Comando para copiar la calve pública de Fedora a Alma Linux

Finalmente se comprobó mediante el comando 1, presente en la Figura 3.33, que se puede acceder por SSH desde Fedora al sistema Alma Linux sin la necesidad de introducir la contraseña. Se puede finalizar la conexión con el comando “exit”.

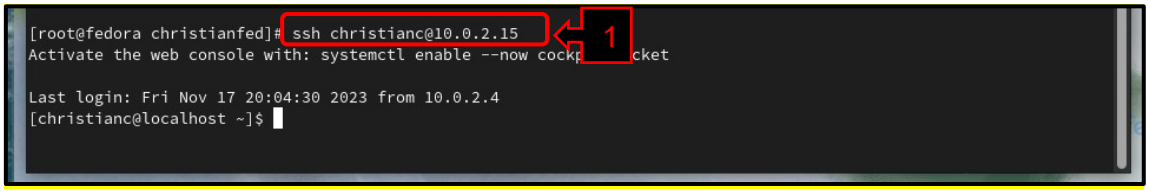

Figura 3.33 Comprobación de conexión por clave pública desde Fedora

Creación del inventario

El inventario es un archivo estructurado en formato INI o YAML, en Ansible el inventario contiene la información sobre cómo establecer la conexión con el nodo controlado. El inventario de este proyecto está en formato INI, por ende, todo lo que está entre corchetes corresponde a un grupo, las variables que se escriben debajo de estos se usan para especificar características y así personalizar la configuración según las necesidades requeridas por un grupo o por cada *host* dentro de ese grupo.

Existen dos tipos de inventarios, los inventarios estáticos y dinámicos. La diferencia principal entre ambos es cómo se gestiona y actualiza la información acerca de los nodos controlados. Un inventario dinámico está basado en *scripts* mientras que un inventario estático está definido en un fichero, por lo que si su estructura cambia el administrador debe realizar actualizaciones manuales. Sin embargo, para entornos como el presente proyecto, en donde el nodo a ser controlado se conoce de antemano, el inventario estático es el más adecuado.

Por defecto, Ansible usa la ruta predeterminada “/etc/ansible/hosts” para el inventario como se puede observar en la sección 1 de la Figura 3.34. Pero para organizar los archivos de inventario y *playbook* en la sección 2 de la misma figura se puede visualizar la estructura del directorio “Ansible” y los subdirectorios que se crearon, la cual es también una opción válida.

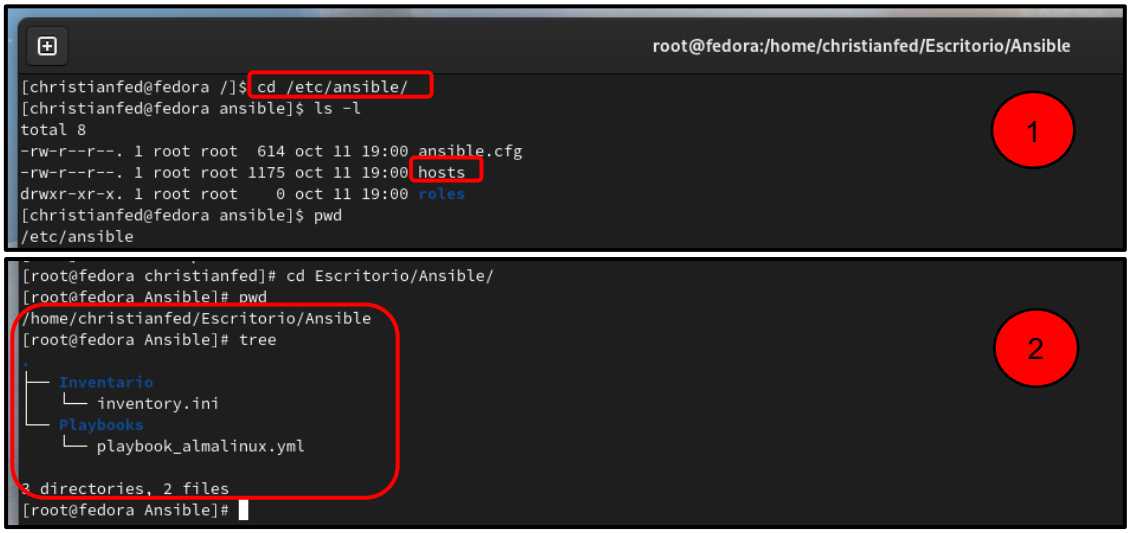

Figura 3.34 Ruta y estructura del directorio “Ansible”

En la Figura 3.35 se observa el inventario estático “inventory.ini”. En el archivo se puede observar el grupo “[alma_linux_servers]” el cual contiene a los *hosts* o clientes en los que se va a ejecutar el *plakbook;* en este caso un solo cliente especificado como “cliente 1” el cual pertenece al usuario de Alma Linux “christianc”. Por medio de las variables “ansible_host, ansible_user y ansible_become_pass”, que se observan junto con sus comentarios, Ansible establece la conexión y ejecuta las tareas del *playbook* como administrador del sistema.

Figura 3.35 Estructura del inventario

Con el objetivo de comprobar que la estructura del inventario es correcta y también verificar la conectividad sin contraseña con el nodo controlado, se realizó la prueba con el módulo ping. En Ansible se utiliza el módulo ping para realizar la verificación de

conectividad con los nodos especificados en el inventario. En este caso garantiza que el nodo de control tenga los permisos adecuados y evalúa la capacidad de establecer la conexión SSH con el nodo del sistema operativo de servidor Alma Linux.

Para realizar esta comprobación se entró a la ruta “cd Escritorio/Ansible/Inventario/”, en donde se ejecutó el comando 1 que se visualiza en la Figura 3.36. Con el parámetro “-i” se especifica la ubicación del archivo, pero si se está ejecutando desde el directorio que contiene el inventario simplemente se escribe el nombre “inventory.ini”. Con el parámetro “-m” se especifica el módulo, en este caso el módulo ping. Este comando ejecuta el módulo ping en todos los nodos que pertenecen al grupo “[alma_linux_servers]” del inventario.

Como se puede visualizar en el recuadro amarillo de la Figura 3.36 el resultado de la ejecución del módulo fue exitosa ya que la respuesta es “pong”, esta es una pauta de que el nodo controlado respondió correctamente. Indica que Ansible puede establecer la conexión con el “cliente 1”.

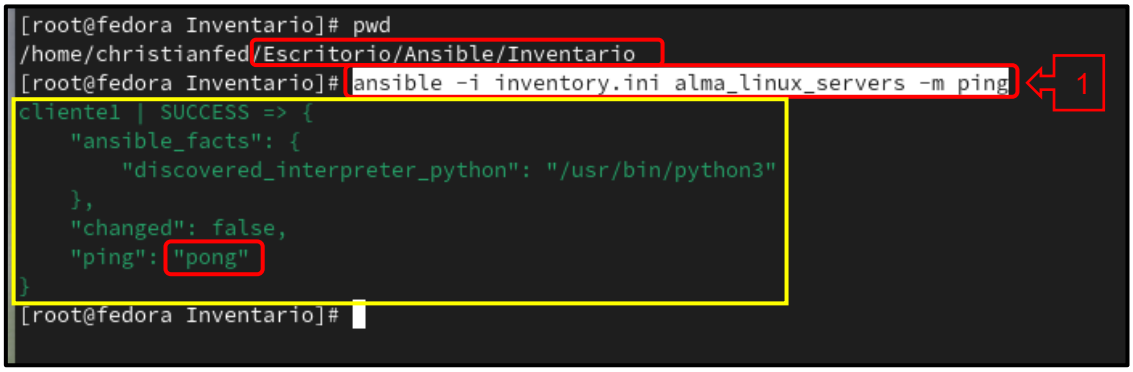

Figura 3.36 Comando para comprobar la estructura del inventario

Creación del *playbook*

Un *playbook* es un archivo redactado en YAML que permite describir las tareas que se quieren realizar sobre un *host* que se tiene en el inventario, el *playbook* se puede correr las veces que se desee. Para definir las tareas es necesario utilizar los módulos de Ansible, con el comando 1 “ansible-doc nombre_del_modulo” que se muestra en la Figura 3.37 es posible visualizar información detallada sobre cualquier módulo concreto. Otra forma de acceder a la información acerca de los módulos es por medio de la página de documentación oficial de Ansible docs.ansible.com. Aquí se encuentran ejemplos de uso, detalles de sus parámetros o atributos que sirven para adaptar las acciones que se quieren realizar con el módulo.

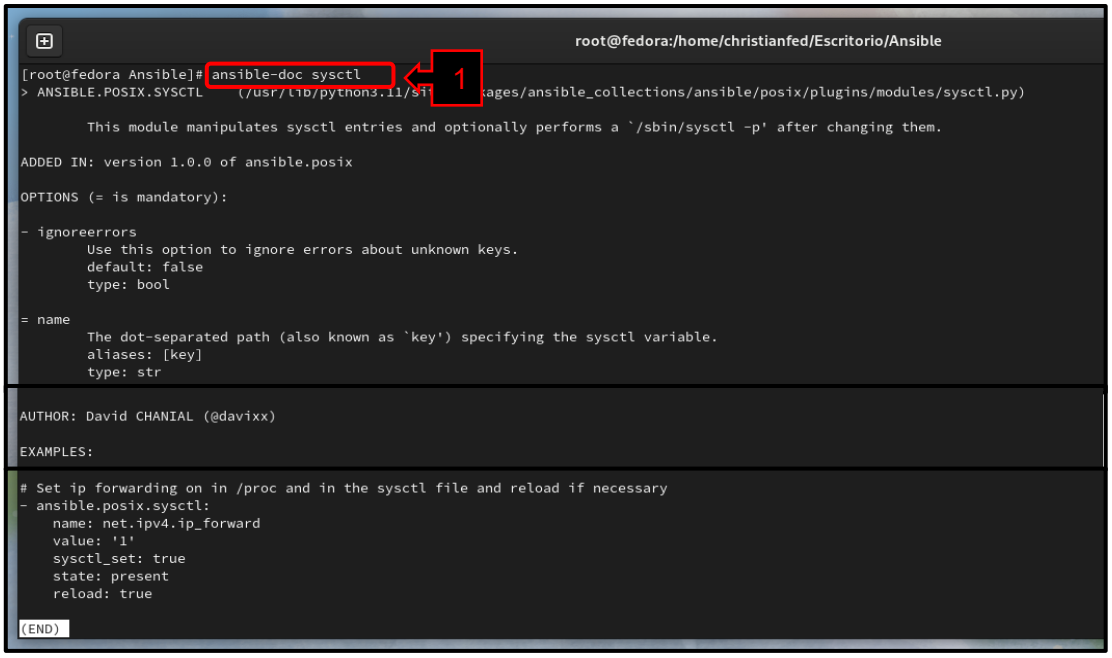

Figura 3.37 Comando para visualizar información de los módulos

En la Figura 3.38 se puede ver la estructura completa del *playbook* creado, con todos sus bloques. Se ocupó los bloques: *vars prompt, tasks* y *handlers,* para diseñar las tareas correspondientes a vulnerabilidades solventadas de forma dinámica y los bloques: *tasks* y *handlers,* para las tareas correspondientes a vulnerabilidades solventadas de manera estática.

Figura 3.38 Bloques del *playbook*

Para comprender la estructura del *playbook* creado, es necesario dar una descripción de sus secciones o bloques como se conocen en Ansible a las partes del *playbook*, estos se muestran a continuación:

- ***Play*:** El bloque *play* que se muestra en la Figura 3.38 corresponde a la sección más grande dentro del *playbook*, este agrupa a los bloques más pequeños; *vars prompt*, *tasks* y *handlers*.

  Respecto a este bloque *play*, primero se colocó el encabezado que empieza por un triple guion "---", este se usa para indicar el inicio de un documento YAML. En un archivo es posible tener varios documentos YAML en donde cada "---" puede representar un *playbook*. Después se estableció la configuración principal para la ejecución del *playbook*, aquí se definió las características como el grupo "[alma_linux_servers]" por medio del parámetro "hosts", la descripción del *playbook* con "name" y con "become: true" declarado de forma global se indicó que todas las tareas se ejecuten con privilegios de superusuario. Todos los parámetros mencionados se pueden visualizar en la Figura 3.39.

Figura 3.39 Bloque *play* del *playbook*

- ***Vars prompt***: Este bloque que se visualiza en la Figura 3.40, representa la forma de definir variables interactivas, por medio de estas se solicitó al usuario el ingreso de datos durante la ejecución del *playbook*. Es una opción para definir variables para interactuar con el usuario en lugar de hacerlo de forma estática con el bloque *vars*.

  En relación con este bloque, en el *playbook* se creó dos tareas simples para dar una indicación general que se presenta al usuario antes y después de ejecutar las tareas dinámicas colocadas igualmente en este bloque, pero que se explicarán más adelante junto con cada vulnerabilidad a la que están relacionadas. En la Figura 3.40 se pueden ver las dos tareas a las que se les asignó el nombre de "indicacion_general" e "indicacion_aplicar cambios" por medio del parámetro "name". Por medio de "prompt: |" se permite la entrada de un bloque de texto respecto a las indicaciones que se mostrarán al usuario y con "private: no" se establece que cualquier dato que ingrese el usuario no es privado por lo que en la ejecución del *playbook* se visualizará.

Figura 3.40 Bloque *vars prompt* del *playbook*

- ***Tasks***: En el bloque *tasks* observado en la Figura 3.41, se encuentra la lista de tareas que serán ejecutadas por medio del nodo de control. Cada una de estas tareas realiza una acción específica por medio de un módulo, empiezan con un guion y el parámetro “name” con el propósito de indicar su inicio y proporcionar una pequeña descripción de lo que hacen. Cada módulo tiene diferentes parámetros reconocidos por el mismo, el parámetro “name” y su correspondiente descripción se pueden visualizar también en la salida al ejecutar el *playbook*. Esto con el objetivo de que el usuario pueda leer y entender fácilmente la ejecución.

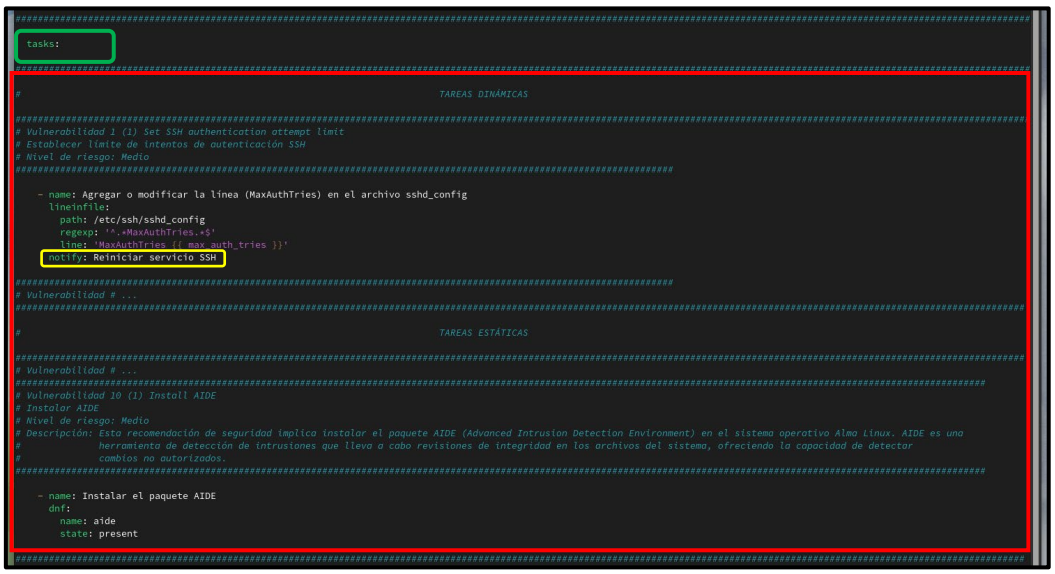

Figura 3.41 Bloque *tasks* del *playbook*

- ***Handlers***: También llamados manejadores, este bloque que se puede ver en la Figura 3.42, corresponde a tareas que se ejecutan solo cuando se han detectado

cambios durante la ejecución del *playbook*. Es decir, los *handlers* son acciones que se ejecutan después de las tareas definidas en el bloque *tasks*. Solo se activan como respuesta a cambios provocados por estas, por ejemplo, en el *playbook* se programó una tarea en el bloque *tasks* que realiza cambios en los archivos de configuración del servicio SSH, entonces se activa el *handler* relacionado para reiniciar el servicio y que esos cambios se efectúen. Un *handler* se diferencia de una tarea del bloque *tasks,* la cual se puede crear para reiniciar un servicio, porque la tarea se ejecutará sin importar si hubo o no cambios.

En la Figura 3.42 se pueden observar los comentarios refrentes a las tareas de las vulnerabilidades 1, 3 y 8, debajo de estos el *handler* "- name: Reiniciar servicio SSH". Las tres tareas de las vulnerabilidades cuentan con el parámetro "notify: Reiniciar servicio SSH" definido de la misma forma que se puede visualizar para la vulnerabilidad 1 en el recuadro amarillo de la Figura 3.41. El propósito de esta comparación es notar que, como se trata de reiniciar un mismo servicio que solicitan 3 tareas diferentes por medio de "notify", no se debe hacer uso de 3 *handlers* diferentes para cada una sino uno solo, esto simplemente se hizo colocando el mismo parámetro "notify: Reiniciar servicio SSH" a todas las tareas.

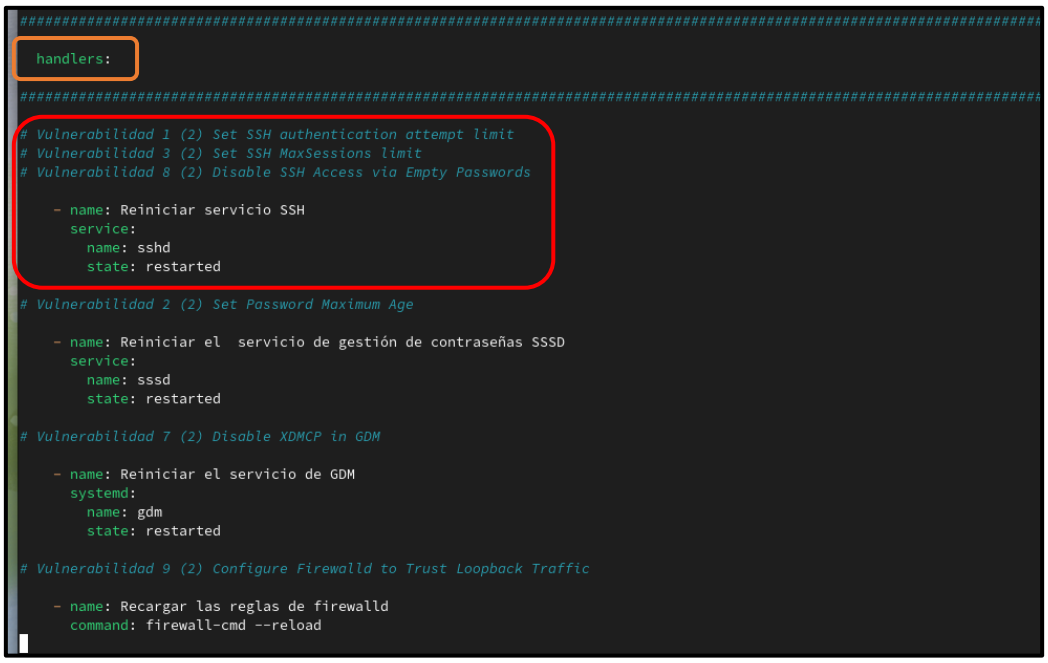

Figura 3.42 Bloque *handlers* del *playbook*

Para solventar las vulnerabilidades, la herramienta de escaneo generó información para cada una, acerca del nivel de riesgo "severity", una descripción de la solución para abordar el problema "description" y una justificación "Rationale", explicando la razón de la recomendación de seguridad. En la Figura 3.43 se pueden visualizar estos elementos.

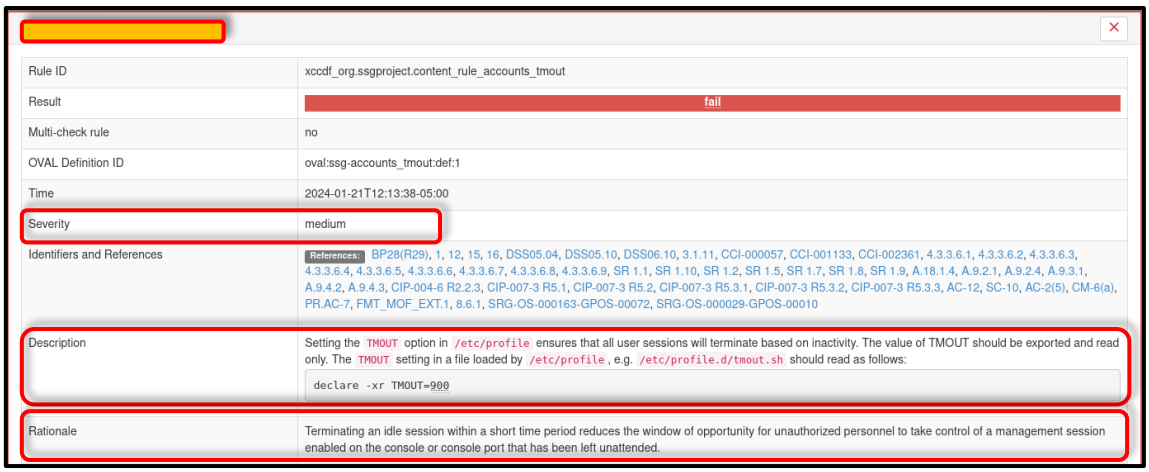

Figura 3.43 Elementos de la información generada de las vulnerabilidades

Se han solventado 10 vulnerabilidades, las 5 primeras de forma dinámica y las restantes de forma estática. Además, se ha incorporado información detallada sobre cada vulnerabilidad al *playbook* mediante el uso de comentarios. Especialmente en las vulnerabilidades solventadas de forma estática, ya que no se muestra ninguna información del problema al usuario durante la ejecución del playbook, debido a que solo se ejecutan los cambios programados. Estos comentarios han sido utilizados también para agrupar tareas relacionadas con una misma vulnerabilidad y para separar los distintos bloques del código, contribuyendo así a mejorar su comprensión y legibilidad.

A partir de las vulnerabilidades que encontró SCAP *Workbench* después del análisis en Alma Linux, siguiendo el perfil CIS nivel 1, se crearon las tareas estáticas y dinámicas en el *playbook*. A continuación, se detalla más acerca de cada vulnerabilidad y recomendación generada por la herramienta, con sus correspondientes tareas en el *playbook* para solventarlas.

Vulnerabilidad # 1: *Set* SSH *authentication attempt limit*

Para solventar la vulnerabilidad número 1, acerca de establecer el límite de intentos de autenticación SSH por parte de un cliente, en la Figura 3.44 se puede observar que el nivel de riesgo es medio. La solución proporcionada es editar “sshd_config”, el cual es un archivo de configuración utilizado por el servicio SSH, y establecer el parámetro “MaxAuthTries” que representa el número máximo de intentos permitidos en 4.

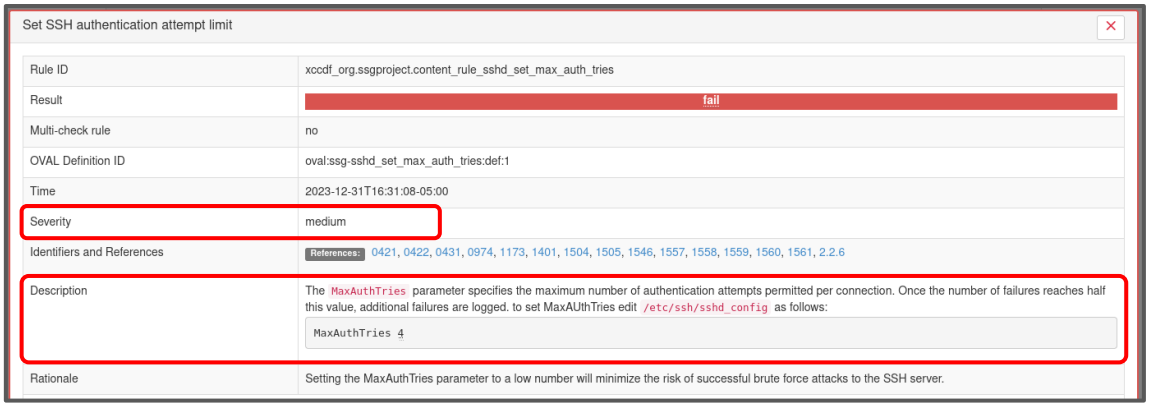

Figura 3.44 Información generada de la vulnerabilidad 1

Respecto a la implementación de la solución, en el *playbook* se creó una tarea en el bloque “vars_prompt”, por medio del parámetro “name” se especificó el nombre de la variable “MaxAuthTries” que almacenará el dato ingresado por el usuario. También mediante “prompt: |” se proporcionó información sobre la vulnerabilidad. Esta información se mostrará al usuario durante la ejecución del *playbook* y además le solicitará la entrada del dato. Finalmente, con “private no” se determinó que el dato ingresado no estará oculto. Todos los parámetros mencionados se pueden visualizar en la Figura 3.45.

Figura 3.45 Tarea en el bloque “vars_prompt”, vulnerabilidad 1

En la Figura 3.46 se muestra la tarea que se diseñó en el bloque “tasks” para editar la línea “MaxAuthTries” en el archivo “sshd_config”. Se hizo uso del módulo “lineinfile” para llevar a cabo la modificación. Con “path” se indica la ruta del archivo, mientras que “regexp” se utiliza para realizar la búsqueda de la expresión “MaxAuthTries” y por medio de “line” se especifica el contenido de la nueva línea que será agregada, con el dato proporcionado por el usuario. Finalmente, con el parámetro “notify” seguido del nombre “Reiniciar servicio SSH” se activa el *handler* que se relaciona a esta tarea.

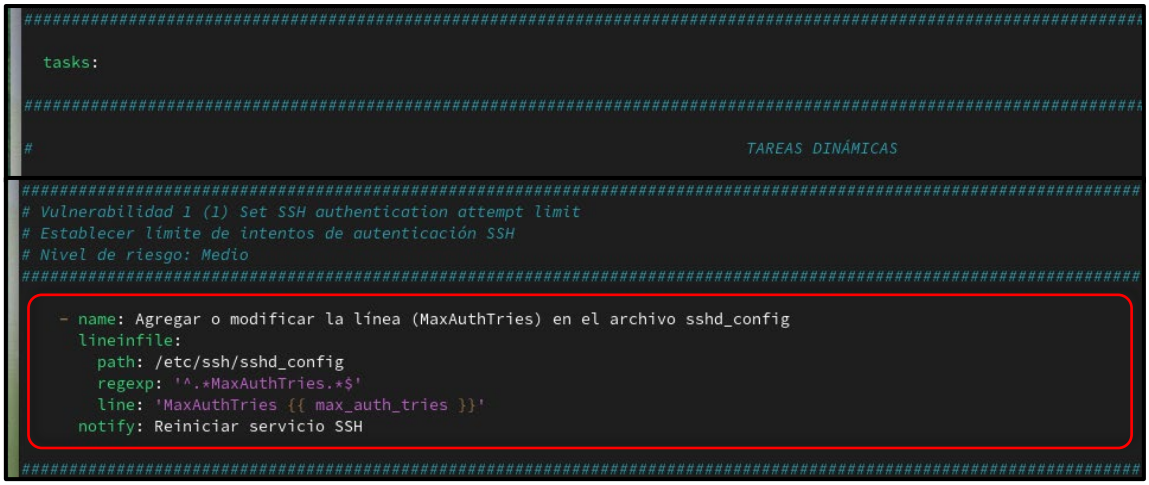

Figura 3.46 Tarea en el bloque "tasks", vulnerabilidad 1

Para completar el proceso, se creó un *handler,* el cual se puede visualizar en la Figura 3.47. Su objetivo es reiniciar el servicio SSH y aplicar los cambios una vez que detecte que el archivo de configuración "sshd_config" ha sido modificado por la tarea "Agregar o modificar la línea (MaxAuthTries) en el archivo sshd_config".

Por medio del parámetro "name", se especificó el nombre del *handler*, como se puede observar lleva la misma etiqueta del parámetro "notify" de la Figura 3.46. También se hizo uso del módulo "service" para gestionar el reinicio del servicio SSH. Como paso final, con "name" se especificó el nombre del servicio y con "state: restarted" se indicó el reinicio.

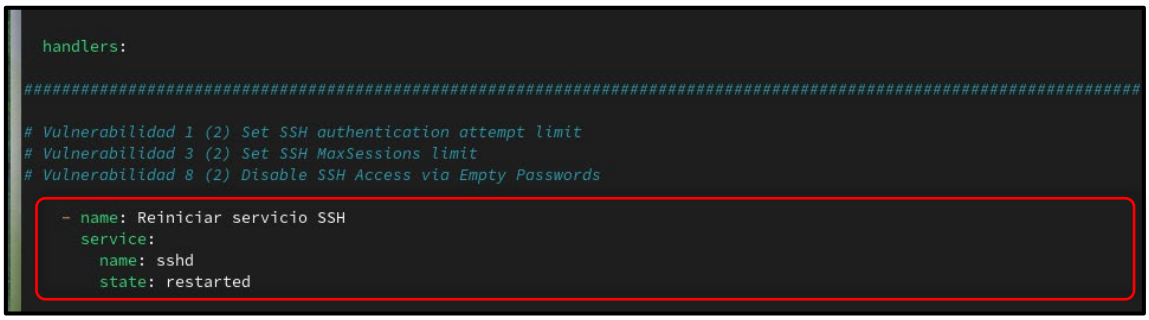

Figura 3.47 Tarea en el bloque "handlers", vulnerabilidad 1

Vulnerabilidad # 2: *Set Password Maximum Age*

Respecto a la vulnerabilidad número 2, acerca de establecer un límite máximo de tiempo de duración para las contraseñas de los usuarios del sistema operativo Alma Linux. En la Figura 3.48 se puede observar que el nivel de riesgo es medio. La solución dada es entrar en el directorio "/etc" y editar el archivo "login.defs", colocando 365 que representa el número de días en la línea "PASS_MAX_DAYS". "login.defs" es un

archivo de configuración para gestionar las cuentas de usuario en los sistemas basados en RHEL.

Figura 3.48 Información generada de la vulnerabilidad 2

Respecto a la aplicación de la solución, en el *playbook* se incluyó una tarea en el bloque “vars_prompt” con el propósito de solicitar el usuario la introducción del número de días. Se hizo uso del parámetro “name” para establecer el nombre de la variable “pass_max_days” que almacenará el dato. Además, con “prompt: |” se escribió el mensaje de información acerca de la vulnerabilidad. En la última instancia, con “private no” se especificó que el dato introducido sea visible en la ejecución del *playbook*. Todos los parámetros mencionados se pueden visualizar en la Figura 3.49.

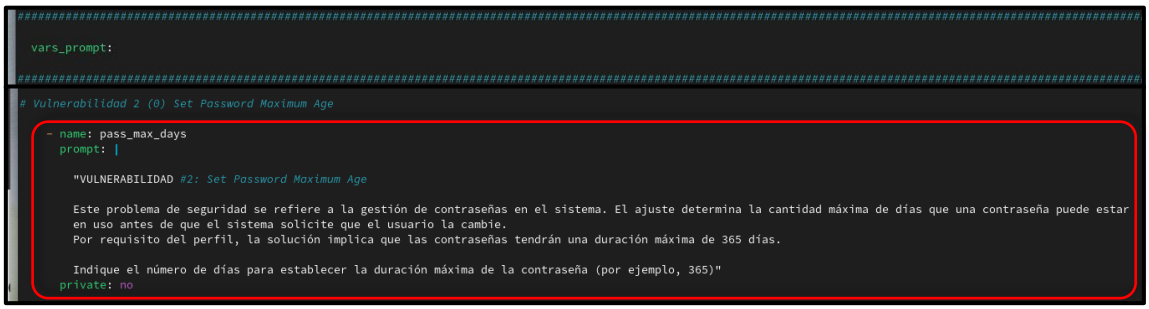

Figura 3.49 Tarea en el bloque “vars_prompt”, vulnerabilidad 2

Se presenta en la Figura 3.50 otra tarea creada en el bloque “tasks”, con el propósito de editar la línea “PASS_MAX_DAYS” en el archivo “login.defs”. Para modificar la línea, se empleó el módulo "lineinfile". La ruta del archivo se indicó mediante "path", y "regexp" se utilizó para buscar la expresión "PASS_MAX_DAYS". La nueva línea, que contiene la expresión y el dato proporcionado por el usuario, se especificó con "line". Finalmente,

para activar el *handler* relacionado con esta tarea se utilizó el parámetro "notify", seguido del nombre " Reiniciar el servicio de gestión de contraseñas SSSD".

```
tasks:

#                                                                       TAREAS DINÁMICAS

# Vulnerabilidad 2 (1) Set Password Maximum Age
# Establecer un límite máximo de edad para las contraseñas de los usuarios del sistema
# Nivel de riesgo: Medio

  - name: Agregar o modificar la línea (PASS_MAX_DAYS) en el archivo login.defs
    lineinfile:
      path: /etc/login.defs
      regexp: '^.*PASS_MAX_DAYS.*$'
      line: 'PASS_MAX_DAYS {{ pass_max_days }}'
    notify: Reiniciar el  servicio de gestión de contraseñas SSSD
```

Figura 3.50 Tarea en el bloque “tasks”, vulnerabilidad 2

Para concluir el proceso, se creó un *handler* cuyo objetivo es reiniciar el servicio de gestión de contraseñas SSSD y aplicar los cambios programados por la tarea “Agregar o modificar la línea (PASS_MAX_DAYS) en el archivo login.defs”. Este *handler* se puede ver en la Figura 3.51. El nombre del *handler* se especificó mediante el parámetro "name", el cual coincide con la etiqueta del parámetro "notify" de la Figura 3.50. Se utilizó el módulo "service" para programar el reinicio del servicio SSSD. Finalmente, mediante "name" se indicó el nombre del servicio y "state: restarted" sirve para señalar el reinicio.

```
handlers:

# Vulnerabilidad 2 (2) Set Password Maximum Age

  - name: Reiniciar el  servicio de gestión de contraseñas SSSD
    service:
      name: sssd
      state: restarted
```

Figura 3.51 Tarea en el bloque “handlers”, vulnerabilidad 2

Vulnerabilidad # 3: *Set* SSH *MaxSessions limit*

Se abordó la vulnerabilidad número 3, sobre establecer un límite máximo de sesiones simultáneas permitidas para un usuario, al conectarse al sistema operativo Alma Linux a través del servicio SSH. En la Figura 3.52 se puede visualizar que el nivel de riesgo es medio. La solución sugerida es editar el archivo “sshd_config” en el directorio “/etc/ssh” y establecer el parámetro “MaxSessions” que representa el número máximo de sesiones en 10.

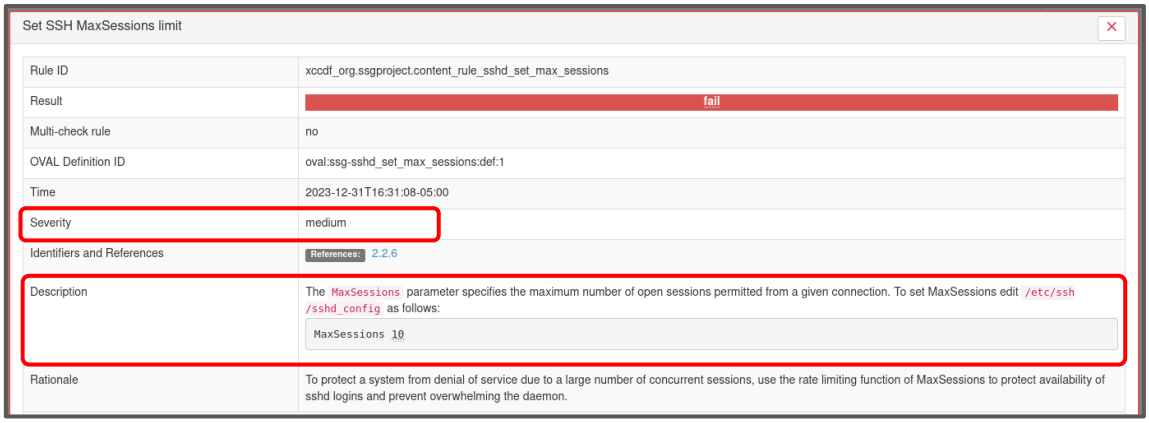

Figura 3.52 Información generada de la vulnerabilidad 3

Con relación a la implementación de la solución, en el *playbook* se incorporó al bloque “vars_prompt” una tarea. En la que se definió la variable “max_sessions” mediante el parámetro “name” para guardar el dato ingresado por el usuario. También por medio de “prompt: |” se colocó la información de la vulnerabilidad y la petición para que el usuario ingrese el valor. Por último, con “private no” se estableció que el valor ingresado no esté oculto. La Figura 3.53 muestra todos los parámetros que se han mencionado acerca de la tarea.

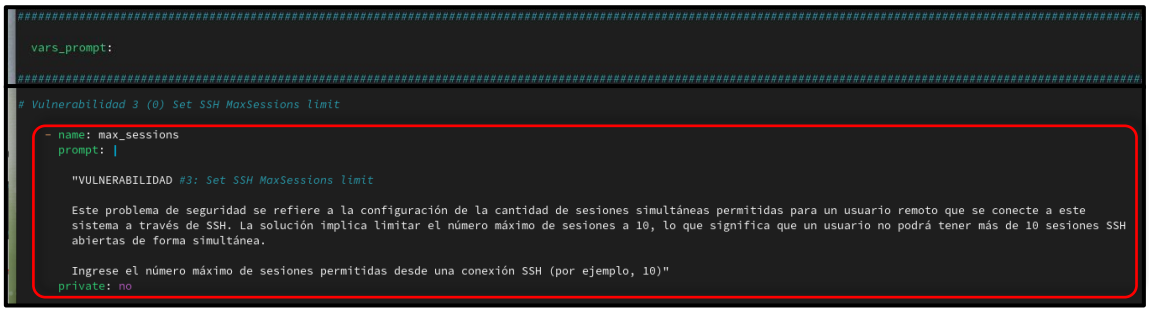

Figura 3.53 Tarea en el bloque “vars_prompt”, vulnerabilidad 3

En el bloque "tasks" se creó otra tarea para editar la línea " MaxSessions" en el archivo “sshd_config”. Se utilizó el módulo "lineinfile" para llevar a cabo esta modificación. La ruta del archivo se especificó con "path" y se empleó el atributo "regexp" para identificar la expresión "MaxSessions". La nueva línea creada con el atributo "line", será resultado de la expresión "MaxSessions" más el dato ingresado por el usuario. Finalmente, se activó el *handler* relacionado con esta tarea mediante el parámetro "notify" seguido del nombre "Reiniciar servicio SSH". La Figura 3.54 muestra la tarea creada en el bloque "tasks" para realizar estas modificaciones en la configuración del servicio SSH.

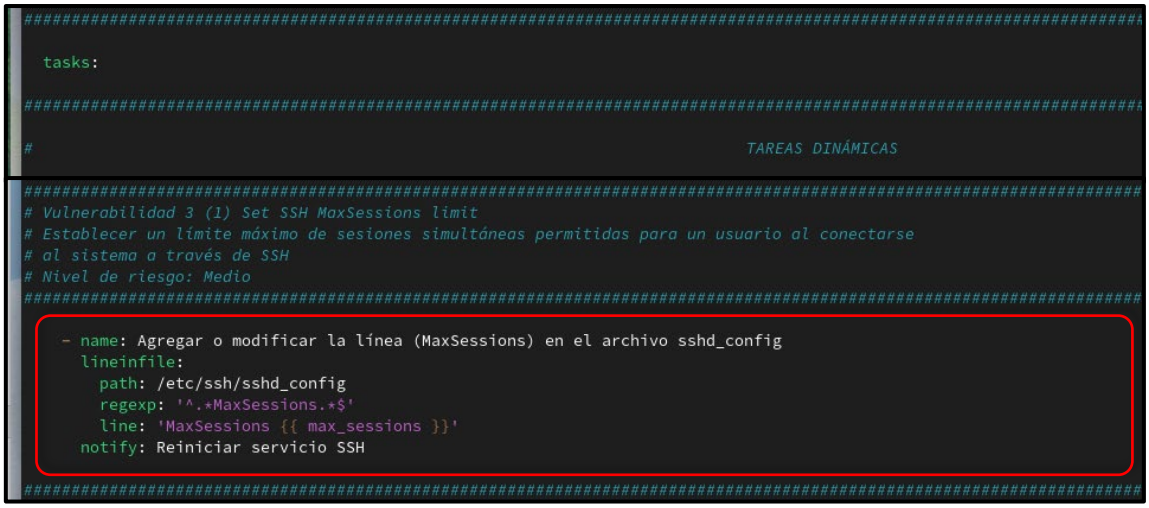

Figura 3.54 Tarea en el bloque “tasks”, vulnerabilidad 3

El *handler*, de la Figura 3.55, se creó con el objetivo de reiniciar el servicio SSH y aplicar las modificaciones, una vez que detecte los cambios en el archivo de configuración "sshd_config", causados por la tarea "Agregar o modificar la línea (MaxSessions) en el archivo sshd_config". Respecto a su estructura, a través del parámetro "name" se atribuyó un nombre al *handler*, el cual coincide con la etiqueta utilizada en el atributo "notify" de la Figura 3.54. Asimismo, se utilizó el módulo "service" para llevar a cabo el reinicio del servicio SSH, mediante los atributos "name: sshd" y "state: restarted".

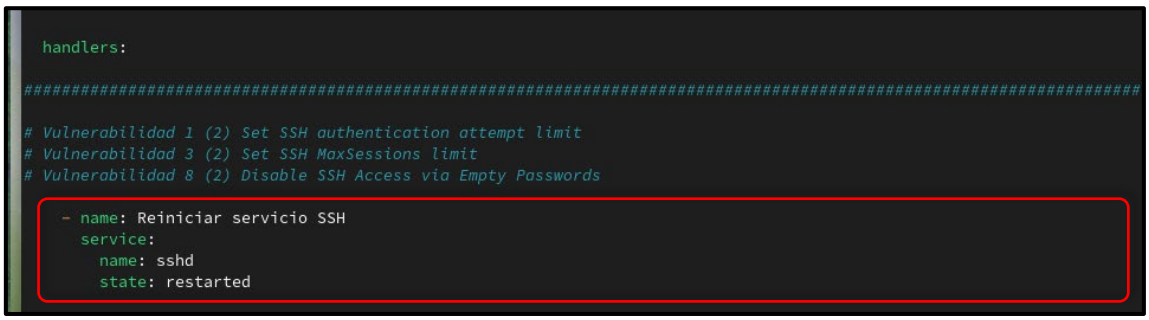

Figura 3.55 Tarea en el bloque “handlers”, vulnerabilidad 3

Vulnerabilidad # 4: *Set Account Expiration Following Inactivity*

Para resolver la vulnerabilidad número 4, relacionada con establecer el número de días después de que una contraseña expire hasta que la cuenta sea inhabilitada por inactividad del usuario. En la Figura 3.56 se puede visualizar que el nivel de riesgo es medio. La solución implica entrar en la ruta “/etc/default” al archivo “useradd” y modificar el parámetro “INACTIVE” poniendo 30, que significa el número de días. Respecto al archivo “useradd”, este se utiliza para gestionar la configuración de cuentas de usuario.

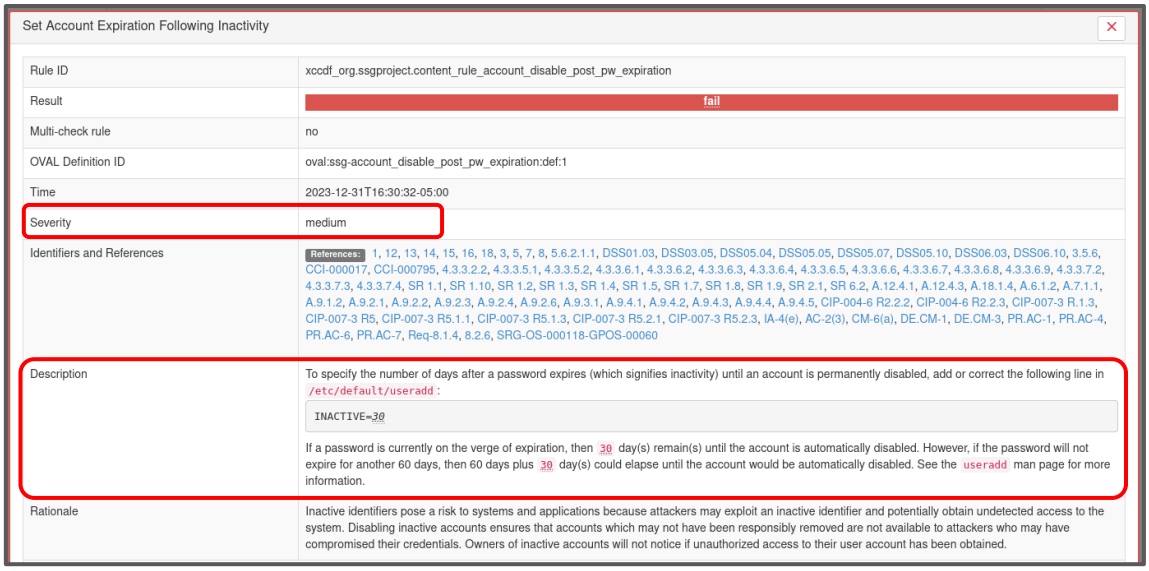

Figura 3.56 Información generada de la vulnerabilidad 4

En cuanto al proceso de aplicar la solución, en el *playbook* se creó una tarea en el bloque “vars_prompt”. Con el objetivo de guardar el dato ingresado por el usuario se hizo uso de la variable” inactive_days” establecida mediante el parámetro “name”. Por otra parte, para para presentar al usuario información respecto a la vulnerabilidad y solicitar la entrada se usó “prompt: |” y finalmente con “private no” se especificó que el dato ingresado sea visible. En la Figura 3.57 se pueden observar todos los parámetros nombrados.

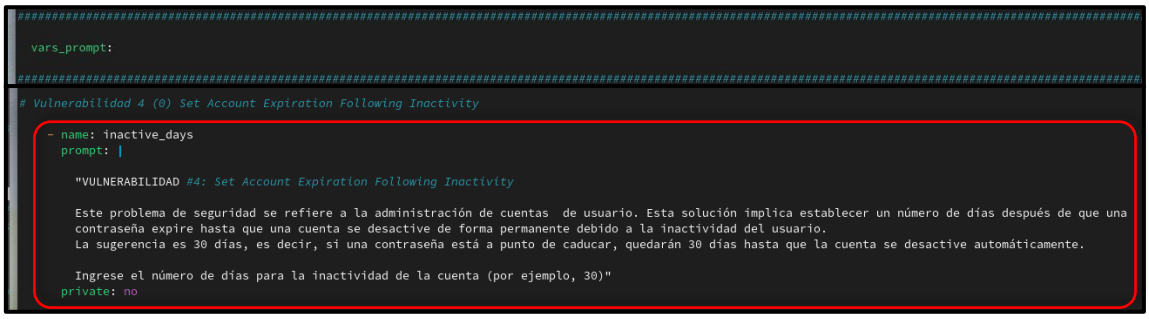

Figura 3.57 Tarea en el bloque “vars_prompt”, vulnerabilidad 4

La tarea diseñada en el bloque "tasks" para editar la línea "INACTIVE" en el archivo" useradd" se muestra en la Figura 3.58. Se utilizó el módulo "lineinfile" para modificar la línea. El directorio del archivo se especificó mediante "path", mientras que "regexp" se empleó para identificar la expresión "INACTIVE". La nueva línea que se agregó con la expresión "INACTIVE", más el dato proporcionado por el usuario a través de la variable “inactive_days”, se especificó mediante "line".

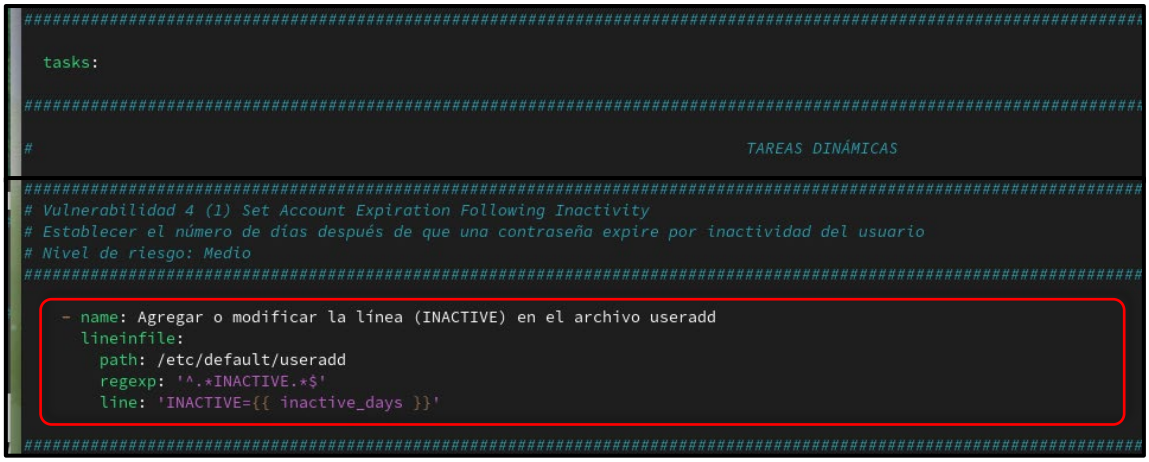

Figura 3.58 Tarea en el bloque “tasks”, vulnerabilidad 4

Vulnerabilidad # 5: *Uninstall* avahi *Server Package*

En relación con la vulnerabilidad número 5, acerca de desinstalar el paquete del servidor Avahi del sistema operativo Alma Linux. En la Figura 3.59 se puede observar que el nivel de riesgo es medio. La recomendación proporcionada es instalar los paquetes relacionados al servidor Avahi: avahi-autoipd y avahi.

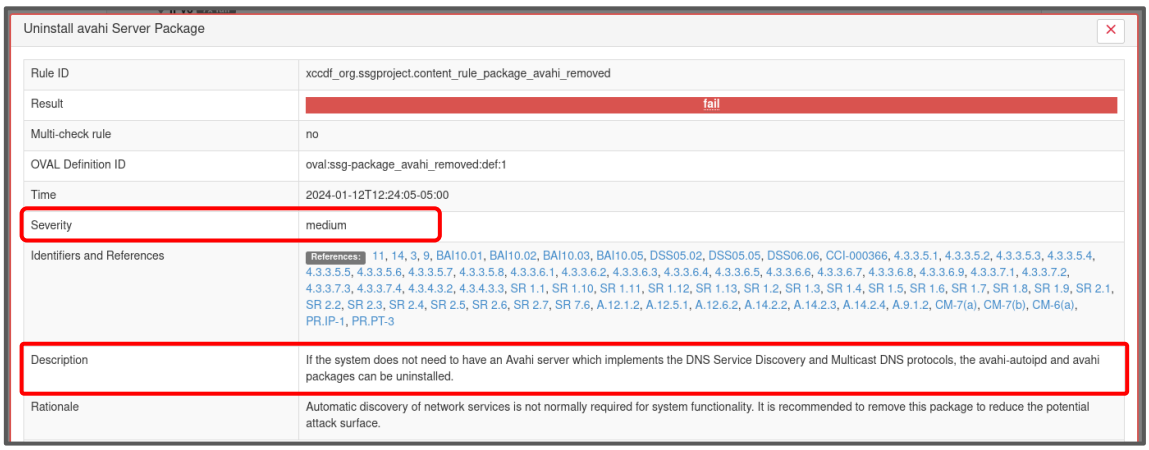

Figura 3.59 Información generada de la vulnerabilidad 5

Respecto a la implementación de la solución, en el *playbook* se incluyó en el bloque “vars_prompt” una tarea. En la que se definió la variable “desinstalar_avahi” por medio del parámetro “name” para almacenar el dato introducido por el usuario. También por medio de “prompt: |” se proporcionó la información de la vulnerabilidad y la pregunta acerca de la desinstalación. Por último, con “private no” se estableció que el dato ingresado sea visible. La Figura 3.60 muestra todos los parámetros que se han mencionado.

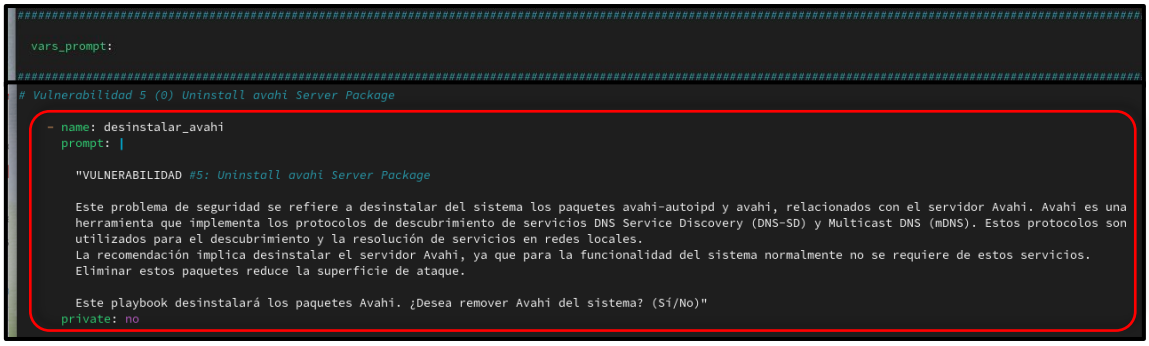

Figura 3.60 Tarea en el bloque “vars_prompt”, vulnerabilidad 5

En el bloque “tasks” se crearon tres tareas, en sus descripciones indicadas por medio del parámetro “- name:” se puede identificar sus propósitos. Estas se pueden visualizar en la Figura 3.61. Por medio del recuadro azul se muestra la primera tarea “- name: Confirmación del usuario para desinstalar los paquetes Avahi”, en donde se hizo uso del módulo “pause:” para detener la ejecución del *playbook* y solicitar al usuario que confirme la respuesta. Caso contrario si ingresa “no” se omite la tarea, por ende, la desinstalación de la herramienta. El parámetro “prompt:” sirve para presentar el mensaje de confirmación al usuario y con “register:” se guarda la respuesta del usuario en la variable “desinstalar_avahi”.

El recuadro verde que se observa, de igual manera en la Figura 3.61, muestra la segunda tarea, cuyo propósito es: “- name: Detener la ejecución del *playbook* si la respuesta ingresada no es válida”. Se utilizó el módulo “fail:” para detener la ejecución del *playbook*, con el parámetro “msg:” se muestra el mensaje acerca de la razón de la detención. Finalmente, con la condición “when:” se comprueba que la respuesta ingresada haya sido “si” o “no”; con “user imput” se accede a la entrada del usuario y con “| lower” que es un filtro, se convierte cualquier texto ingresado en minúsculas.

En el último recuadro morado, también de la Figura 3.61, se observa la tercera tarea “- name: Desinstalar paquetes Avahi si se ha confirmado”. El módulo “package:” sirve para gestionar los paquetes, con “state: absent” se indica que se eliminen los paquetes del sistema y mediante el blucle “loop” se coloca la lista de paquetes que se quiere desinstalar. Finalmente, con “when” se establece que solo se ejecutará esta tarea si la respuesta ingresada en la anterior tarea era “si”. La tarea se ejecuta dos veces por el bucle “loop”, una vez para desinstalar “avahi-autoipd” y la otra para “avahi”.

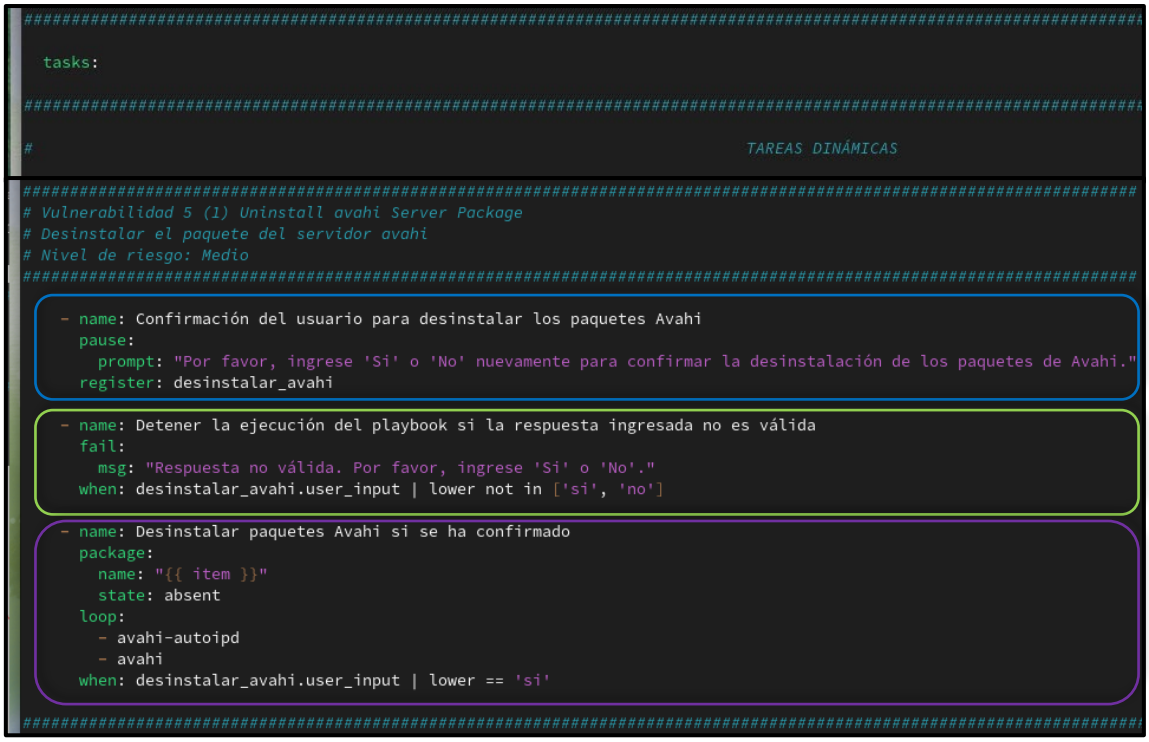

Figura 3.61 Tareas en el bloque “tasks”, vulnerabilidad 5

Vulnerabilidad # 6: *Disable Kernel Parameter for Accepting Secure ICMP Redirects on all IPv4 Interfaces*

Para solucionar la vulnerabilidad número 6, acerca de deshabilitar el parámetro del kernel para aceptar redirecciones ICMP seguras en todas las interfaces IPv4. Se puede observar en la Figura 3.62, que el nivel de riesgo es medio. La solución primero es ejecutar el comando 1, que sirve para deshabilitar ese parámetro y para que la configuración sea persistente al reiniciar el sistema, la segunda indicación es editar el archivo “99-sysctl.conf” en el directorio “/etc/sysctl.d”, colocando la línea de texto del recuadro azul presente en la misma figura.

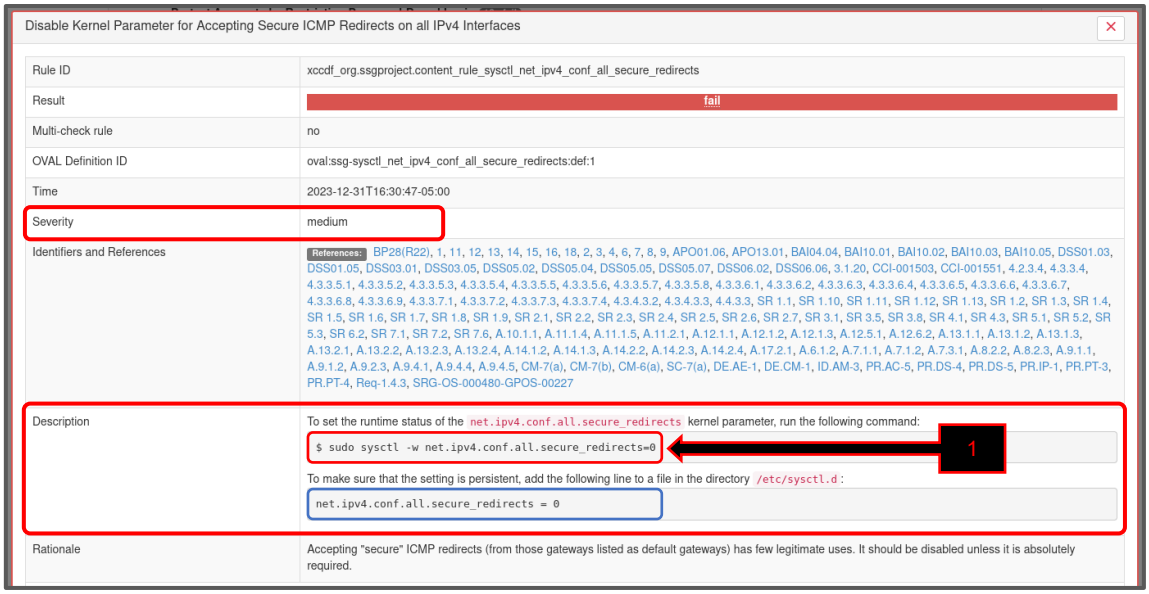

Figura 3.62 Información generada de la vulnerabilidad 6

Dentro del bloque "tasks", se han agregado tres tareas, cuyos propósitos pueden ser identificados en sus descripciones, señalados mediante el parámetro "- name:". Además, se incluyó información detallada sobre la vulnerabilidad mediante comentarios. Estas tareas se visualizan en la Figura 3.63. La primera tarea "Desactivar aceptación de redirecciones ICMP seguras en tiempo de ejecución" se presenta dentro del recuadro azul. En esta tarea, se empleó el módulo " sysctl:" para modificar el parámetro del kernel “net.ipv4.conf.all.secure_redirects” que es el que controla la aceptación de redirecciones ICMP, para especificar ese parámetro se ha hecho uso de “name”. Finalmente, mediante “value 0” se le está asignando un valor de cero al parámetro del kernel para deshabilitarlo.

Se aprecia en el recuadro verde, también presente en la Figura 3.63, la segunda tarea titulada "Asegurar que la configuración sea persistente cada vez que el sistema se inicie". En esta tarea, se hizo uso del módulo "lineinfile:" para colocar la línea en el archivo “99-sysctl.conf”. El parámetro "dest:" se utilizó para indicar el directorio del archivo. Asimismo, se empleó el parámetro “line:” para especificar el contenido de la línea que se agregará. El “become: true” que se aprecia en esta y en las otras tareas sirve para ejecutarlas con privilegios de superusuario, pero en el encabezado del bloque *play* que se puede ver en la Figura 3.39 ya se estableció de forma global, por lo que ya es opcional definirlo en cada tarea.

En el recuadro morado final de Figura 3.63, se presenta la tercera tarea: "Recargar configuración de sysctl para aplicar los cambios". Para esta tarea, se utilizó el módulo "command:" para ejecutar el comando “sysctl -p”, que carga los valores de configuración

de los parámetros del sistema en el kernel sin necesidad de reiniciar el sistema. Estos valores de configuración han sido modificados antes por medio de las dos tareas anteriores.

```
tasks:

#                                                    TAREAS ESTÁTICAS

# Vulnerabilidad 6 (1) Disable Kernel Parameter for Accepting Secure ICMP Redirects on all IPv4 Interfaces
# Deshabilitar el parámetro del núcleo para aceptar redirecciones ICMP seguras en todas las interfaces IPv4
# Nivel de riesgo: Medio
# Descripción: Esta recomendación de seguridad implica deshabilitar la aceptación de redirecciones ICMP seguras en todos los interfaces IPv4 del kernel. ICMP es un protocolo
#              que se utiliza para el control y el manejo de mensajes en la capa de red del modelo OSI. Las redirecciones ICMP seguras son mensajes usados por los routers
#              para notificar a los hosts acerca de rutas más eficientes o preferidas. No obstante, hay situaciones en las que son objeto de uso indebido de manera maliciosa.

  - name: Desactivar aceptación de redirecciones ICMP seguras en tiempo de ejecución
    sysctl:
      name: "net.ipv4.conf.all.secure_redirects"
      value: "0"
    become: true

  - name: Asegurar que la configuración sea persistente cada vez que el sistema se inicie
    lineinfile:
      dest: /etc/sysctl.d/99-sysctl.conf
      line: "net.ipv4.conf.all.secure_redirects = 0"
    become: true

  - name: Recargar configuración de sysctl para aplicar los cambios.
    command: sysctl -p
    become: true
```

Figura 3.63 Tareas en el bloque “tasks”, vulnerabilidad 6

Vulnerabilidad # 7: *Disable* XDMCP *in* GDM

Para abordar la vulnerabilidad número 7, respecto a deshabilitar XDMCP en GDM, se puede visualizar en la Figura 3.64, que el nivel de riesgo es alto. La solución proporcionada es editar el archivo “custom.conf”, en el directorio “/etc/gdm” colocando bajo la sección “[xdmcp]” la línea de texto “Enable=false”, para deshabilitar el protocolo XDMCP.

Disable XDMCP in GDM

Rule ID	xccdf_org.ssgproject.content_rule_gnome_gdm_disable_xdmcp
Result	fail
Multi-check rule	no
OVAL Definition ID	oval:ssg-gnome_gdm_disable_xdmcp:def:1
Time	2023-12-31T16:30:32-05:00
Severity	high
Identifiers and References	

Description	XDMCP is an unencrypted protocol, and therefore, presents a security risk, see e.g. XDMCP Gnome docs. To disable XDMCP support in Gnome, set `Enable` to `false` under the `[xdmcp]` configuration section in `/etc/gdm/custom.conf`. For example:
	`[xdmcp]`
	`Enable=false`

Rationale	XDMCP provides unencrypted remote access through the Gnome Display Manager (GDM) which does not provide for the confidentiality and integrity of user passwords or the remote session. If a privileged user were to login using XDMCP, the privileged user password could be compromised due to typed XEvents and keystrokes will traversing over the network in clear text.

Figura 3.64 Información generada de la vulnerabilidad 7

En el bloque "tasks" se han incluido dos tareas. El propósito asociado a cada tarea puede ser identificado en su respectiva descripción, señalada por el parámetro "-name:". Asimismo, se presenta información detallada sobre la vulnerabilidad a través de comentarios. Estas tareas se muestran en la Figura 3.65. Dentro del recuadro azul se encuentra la primera tarea, titulada "Verificar y eliminar la sección [xdmcp] si existe". En esta tarea, se utilizó el módulo "lineinfile:" para revisar si está presente en el archivo "custom.conf" la sección "[xdmcp]". El parámetro "path:" sirve para indicar la ruta del archivo, al este lo complementa "regexp:" que contribuye buscando la sección "[xdmcp]" y con "state absent" se elimina.

En la misma Figura 3.65 se distingue un recuadro verde que corresponde a la segunda tarea, "Modificar la configuración en el archivo custom.conf". En esta tarea, se empleó el módulo "blockinfile:" que sirve para agregar, modificar o eliminar bloques de texto en un archivo. El parámetro "path:" se empleó para especificar la ruta del archivo y mediante "block: |" se incorporó el bloque de texto que se aprecia en el recuadro amarillo. Finalmente, se utilizó "notify:" para activar el *handler* que reinicia el servicio de GDM.

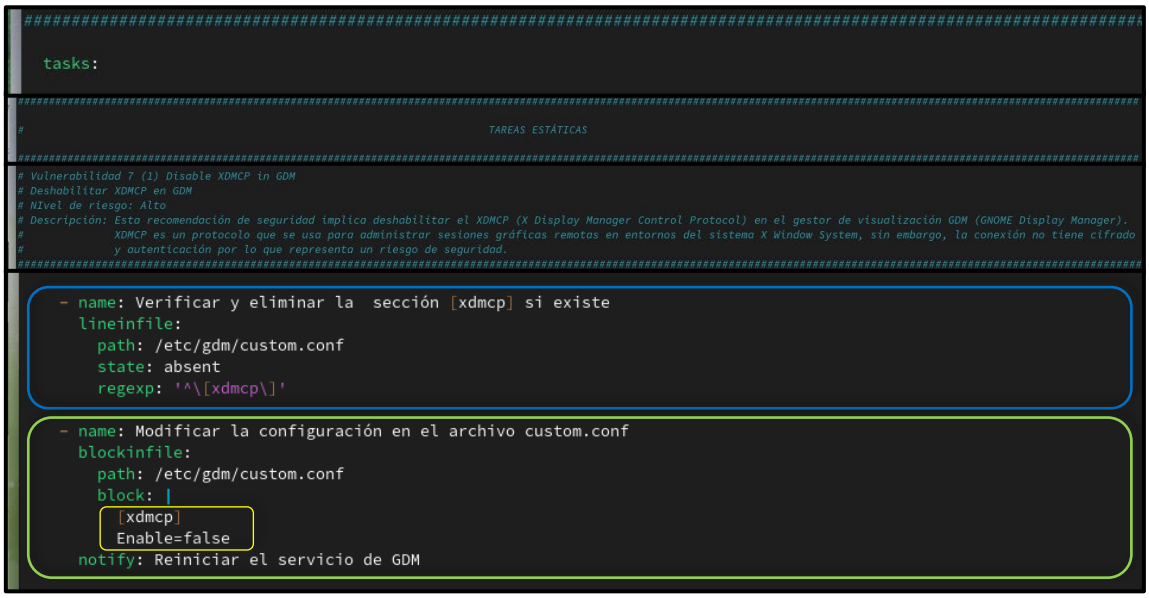

Figura 3.65 Tareas en el bloque "tasks", vulnerabilidad 7

Finalmente se incluyó un *handler* para reiniciar el servicio GDM, con "name" se especificó el nombre de la tarea. También se usó el módulo "systemd" para definir el reinicio a través de los parámetros: "name: gdm" y "state: restarted". El *handler* y sus parámetros se observan en la Figura 3.66.

```
handlers:

# Vulnerabilidad 7 (2) Disable XDMCP in GDM

  - name: Reiniciar el servicio de GDM
    systemd:
      name: gdm
      state: restarted
```

Figura 3.66 Tarea en el bloque “handlers”, vulnerabilidad 7

Vulnerabilidad # 8: *Disable* SSH *Access via Empty Passwords*

Para solventar la vulnerabilidad número 8, respecto a deshabilitar el acceso al servicio SSH para cuentas con contraseñas vacías, en la Figura 3.67 se puede observar que el nivel de riesgo es alto. La solución implica modificar la línea "PermitEmptyPasswords” especificando “no” para deshabilitar el acceso. La herramienta proporcionó la ruta “/etc/ssh/sshd_config.d” donde se encuentra el archivo de configuración “00-complianceascode-handlreha.conf”. Sin embargo, no se encontró ningún archivo con ese nombre, por lo que primero se procedió a crear el archivo “.conf”. Posteriormente se llevaron a cabo las recomendaciones sugeridas por la herramienta.

Disable SSH Access via Empty Passwords

Rule ID	xccdf_org.ssgproject.content_rule_sshd_disable_empty_passwords
Result	fail
Multi-check rule	no
OVAL Definition ID	oval:ssg-sshd_disable_empty_passwords:def:1
Time	2023-12-31T16:31:08-05:00
Severity	high
Identifiers and References	References: NT007(R17), 11, 12, 13, 14, 15, 16, 18, 3, 5, 9, 5.5.6, APO01.06, BAI10.01, BAI10.02, BAI10.03, BAI10.05, DSS05.02, DSS05.04, DSS05.05, DSS05.07, DSS06.02, DSS06.03, DSS06.06, 3.1.1, 3.1.5, CCI-000366, CCI-000766, 164.308(a)(4)(i), 164.308(b)(1), 164.308(b)(3), 164.310(b), 164.312(e)(1), 164.312(e)(2)(ii), 4.3.3.2.2, 4.3.3.5.1, 4.3.3.5.2, 4.3.3.5.3, 4.3.3.5.4, 4.3.3.5.5, 4.3.3.5.6, 4.3.3.5.7, 4.3.3.5.8, 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, 4.3.3.7.1, 4.3.3.7.2, 4.3.3.7.3, 4.3.3.7.4, 4.3.4.3.2, 4.3.4.3.3, SR 1.1, SR 1.10, SR 1.11, SR 1.12, SR 1.13, SR 1.2, SR 1.3, SR 1.4, SR 1.5, SR 1.6, SR 1.7, SR 1.8, SR 1.9, SR 2.1, SR 2.2, SR 2.3, SR 2.4, SR 2.5, SR 2.6, SR 2.7, SR 5.2, SR 7.6, A.10.1.1, A.11.1.4, A.11.1.5, A.11.2.1, A.12.1.2, A.12.5.1, A.12.6.2, A.13.1.1, A.13.1.3, A.13.2.1, A.13.2.3, A.13.2.4, A.14.1.2, A.14.1.3, A.14.2.2, A.14.2.3, A.14.2.4, A.6.1.2, A.7.1.1, A.7.1.2, A.7.3.1, A.8.2.2, A.8.2.3, A.9.1.1, A.9.1.2, A.9.2.1, A.9.2.3, A.9.4.1, A.9.4.4, A.9.4.5, AC-17(a), CM-7(a), CM-7(b), CM-6(a), PR.AC-4, PR.AC-6, PR.DS-5, PR.IP-1, PR.PT-3, FIA_UAU.1, Req-2.2.4, 2.2.6, SRG-OS-000106-GPOS-00053, SRG-OS-000480-GPOS-00229, SRG-OS-000480-GPOS-00227
Description	Disallow SSH login with empty passwords. The default SSH configuration disables logins with empty passwords. The appropriate configuration is used if no value is set for `PermitEmptyPasswords`. To explicitly disallow SSH login from accounts with empty passwords, add or correct the following line in `/etc/ssh/sshd_config.d/00-complianceascode-hardening.conf`: `PermitEmptyPasswords no` Any accounts with empty passwords should be disabled immediately, and PAM configuration should prevent users from being able to assign themselves empty passwords.
Rationale	Configuring this setting for the SSH daemon provides additional assurance that remote login via SSH will require a password, even in the event of misconfiguration elsewhere.

Figura 3.67 Información generada de la vulnerabilidad 8

En el bloque “tasks” se incluyó dos tareas. El objetivo de estas tareas puede ser identificado en su respectiva descripción, mediante el parámetro "- name:". Además, se incluye información detallada sobre la vulnerabilidad mediante comentarios. La Figura 3.68 muestra la estructura de las tareas, la tarea inicial, denominada "Crear archivo de configuración 00-complianceascode-hardening.conf", está resaltada en el recuadro azul. En esta tarea, se empleó el módulo "file:" para crear el archivo “00-complianceascode-hardening.conf”. El parámetro "path:" se usó para indicar el directorio

en donde se creará, por medio de “state: touch” se crea el archivo y con “mode: '0644'” se otorgan los permisos. Los permisos concedidos al archivo fueron de lectura y escritura al propietario, mientras que al grupo y a otros usuarios solo permisos de lectura.

En el recuadro verde de la Figura 3.68 se encuentra la segunda tarea denominada "Agregar la línea (PermitEmptyPasswords no) al archivo 00-complianceascode-hardening.conf". Para llevar a cabo esta tarea, se utilizó el módulo "lineinfile:" con el propósito de agregar la línea al archivo “00-complianceascode-hardening.conf”. Mediante el parámetro "path:" se especificó el directorio y con “line:” el contenido de la línea que se va a agregar. Por último, se utilizó “notify:” para habilitar el *handler* que reinicia el servicio de SSH.

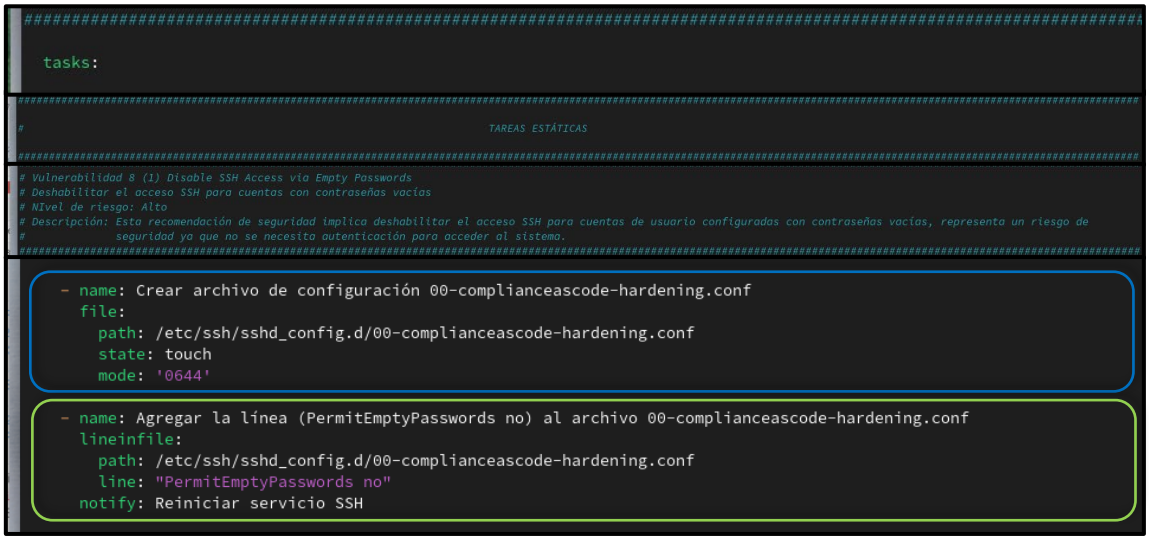

Figura 3.68 Tareas en el bloque “tasks”, vulnerabilidad 8

La Figura 3.69, muestra el *handler* que se diseñó para para reiniciar el servicio SSH, una vez que este detecte los cambios en el archivo de configuración "sshd_config". El *handler* utiliza el módulo "service" para reiniciar el servicio SSH. Con el parámetro "name", se identifica el servicio y mediante "state: restarted" se ejecuta el reinicio.

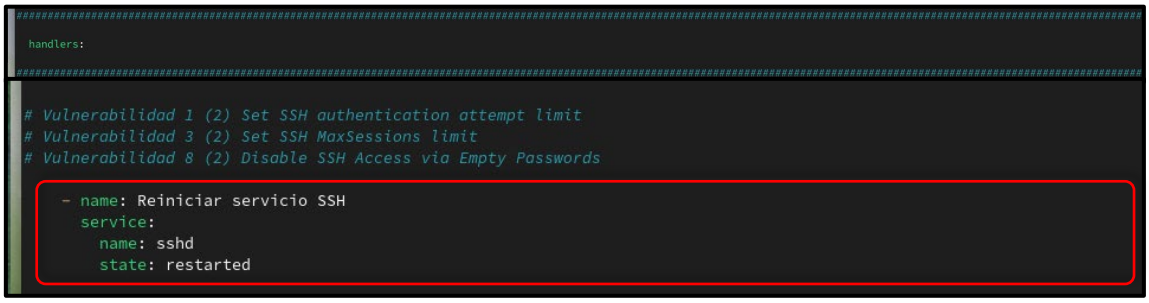

Figura 3.69 Tarea en el bloque “handlers”, vulnerabilidad 8

Vulnerabilidad # 9: *Configure* Firewalld *to Trust Loopback Traffic*

En relación con la vulnerabilidad número 9, acerca de configurar el firewalld para confiar en el tráfico de bucle local (*loopback*). Se muestra en la Figura 3.70, que el nivel de riesgo es medio. La solución implica ejecutar los comandos 1 y 2 presentes en la misma figura, el primero para permitir el tráfico de bucle local, el segundo para recargar las reglas del *firewall* y aplicar los cambios.

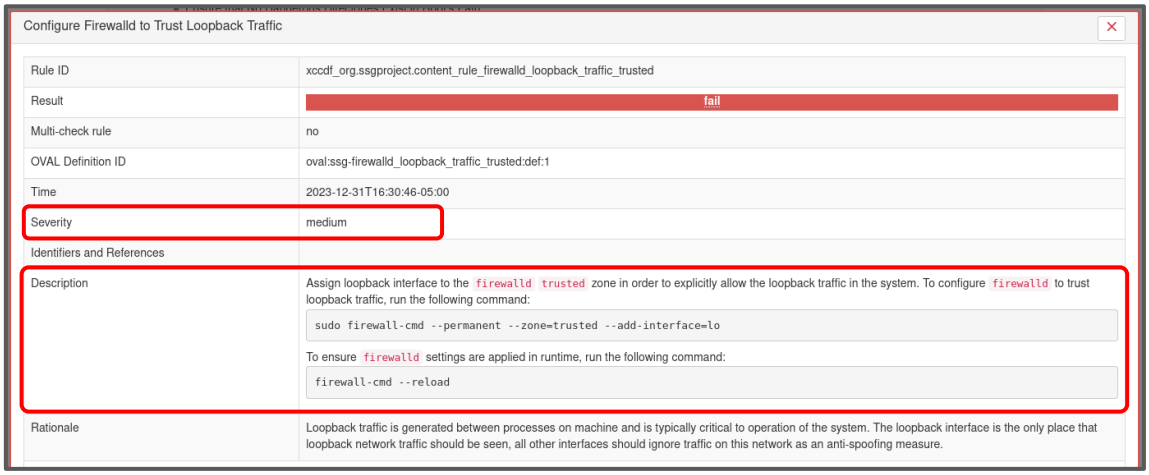

Figura 3.70 Información generada de la vulnerabilidad 9

En la Figura 3.71 se observa en el bloque “tasks” la tarea para solucionar el problema, haciendo uso del parámetro “- name:” se dio una descripción. Además, se incluyó información detallada sobre la vulnerabilidad por medio de comentarios. Mediante el módulo “command:” se ejecuta el comando para asignar a la interfaz de *loopback* a la zona confiable de Firewalld de forma permanente. Finalmente, se utilizó “notify:” para activar el *handler* que recarga las reglas del *firewall*.

```
tasks:

#                                                    TAREAS ESTÁTICAS

# Vulnerabilidad 9 (1) Configure Firewalld to Trust Loopback Traffic
# Configurar Firewalld para confiar en el tráfico de bucle local (loopback)
# Nivel de riesgo: Medio
# Descripción: Esta medida de seguridad implica modificar la configuración del firewall para autorizar el tráfico en la interfaz de bucle local (loopback). Esto quiere decir
#              que, después de realizar este cambio en la configuración, el tráfico de bucle local podrá pasar a través del firewall sin restricciones, lo cual resulta crucial
#              para facilitar la comunicación interna en la máquina.

  - name: Asignar interfaz de loopback a la zona confiable de firewalld
    command: firewall-cmd --permanent --zone=trusted --add-interface=lo
    notify: Recargar las reglas de firewalld
```

Figura 3.71 Tarea en el bloque “tasks”, vulnerabilidad 9

El *handler* para recargar las reglas del *firewall* y aplicar los cambios realizados por la tarea “Asignar interfaz de loopback a la zona confiable de firewalld” se puede visualizar en la Figura 3.72. Se especificó el nombre del *handler* mediante el parámetro "name",

el nombre es igual al de la etiqueta del parámetro "notify" de la Figura 3.71 con el propósito de activarlo. Como paso final se utilizó el módulo "command", que sirve para ejecutar el comando “firewall-cmd –reload”.

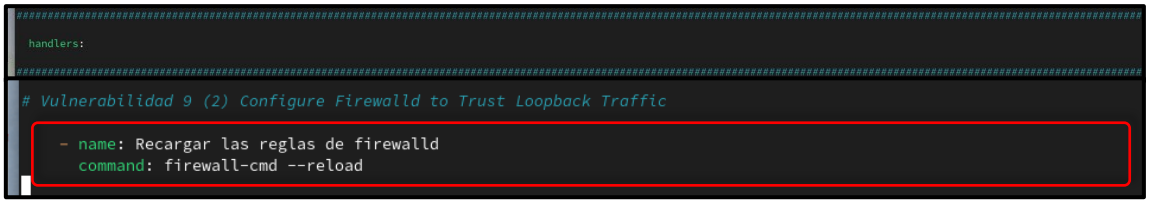

Figura 3.72 Tarea en el bloque “handlers”, vulnerabilidad 9

Vulnerabilidad # 10: *Install* AIDE

Para resolver la vulnerabilidad número 10 acerca de instalar AIDE, la recomendación implica la instalación de esta herramienta; se puede visualizar el comando 1 en la Figura 3.73 que sirve para la instalación. El nivel de riesgo de este problema de seguridad es medio.

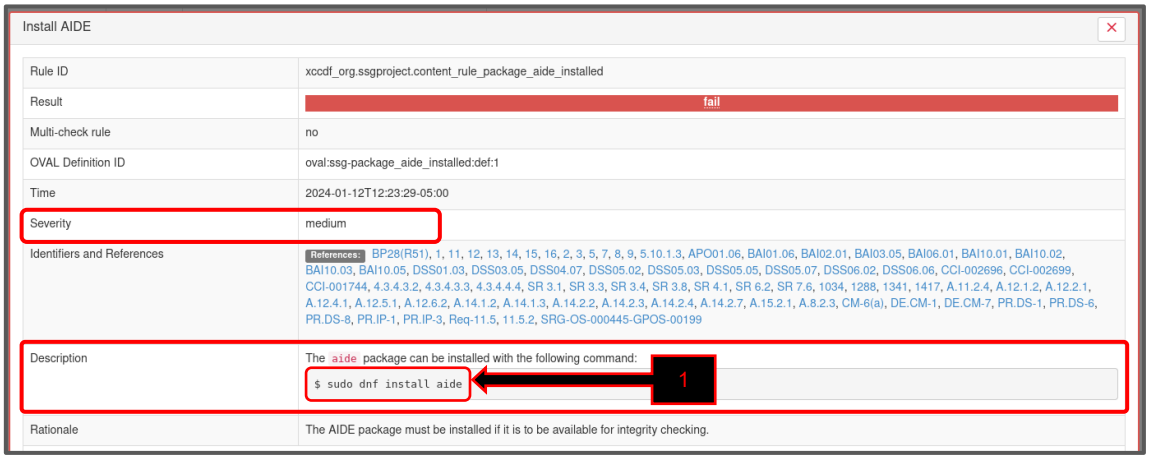

Figura 3.73 Información generada de la vulnerabilidad 10

En el bloque “tasks” que se visualiza en la Figura 3.74, se presenta la tarea para solucionar el problema, por medio del parámetro “- name:” se dio una descripción. Además, se añadió información detallada sobre la vulnerabilidad mediante comentarios. El módulo “dnf:” sirve para gestionar paquetes en los sistemas operativos que toman como base el código fuente de RHEL. Con el parámetro “name:” se proporcionó el nombre del paquete y finalmente con “state: present” se indica a Ansible que debe asegurar que el paquete este instalado en el sistema operativo Alma Linux.

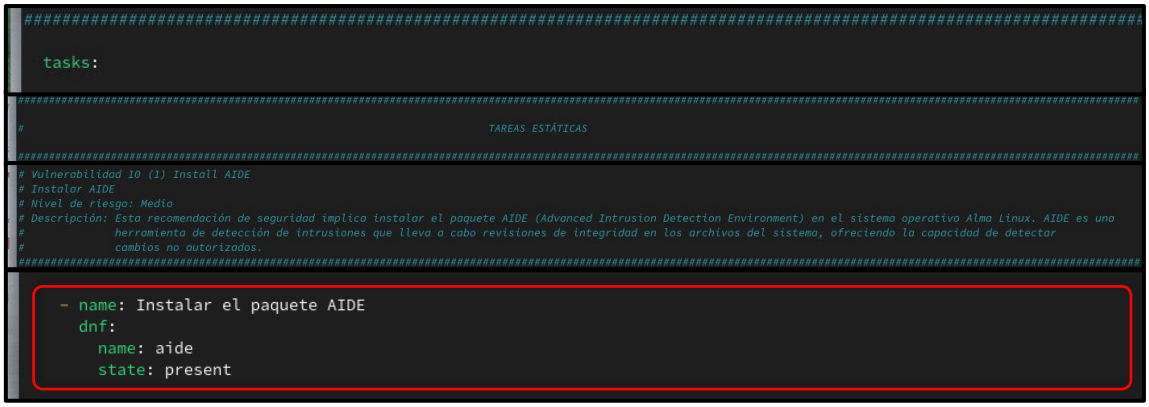

Figura 3.74 Tarea en el bloque “tasks”, vulnerabilidad 10

Ejecución del *playbook*

Para la ejecución del *playbook* se ingresó a la ruta “Escritorio/Ansible/” que contiene a los subdirectorios donde se encuentran los archivos de inventario y *playbook*. En el directorio “/Ansible” se ejecutó el comando 1 que se muestra en la sección 1 de la Figura 3.75, en donde con la opción “-i” se especifica la ruta del archivo del inventario “Inventario/inventory.ini”; seguido de esto se especifica la ruta del archivo del *playbook* “Playbooks/playbook_almalinux.yml”. La estructura de este comando se visualiza en la sección 2 de la misma figura.

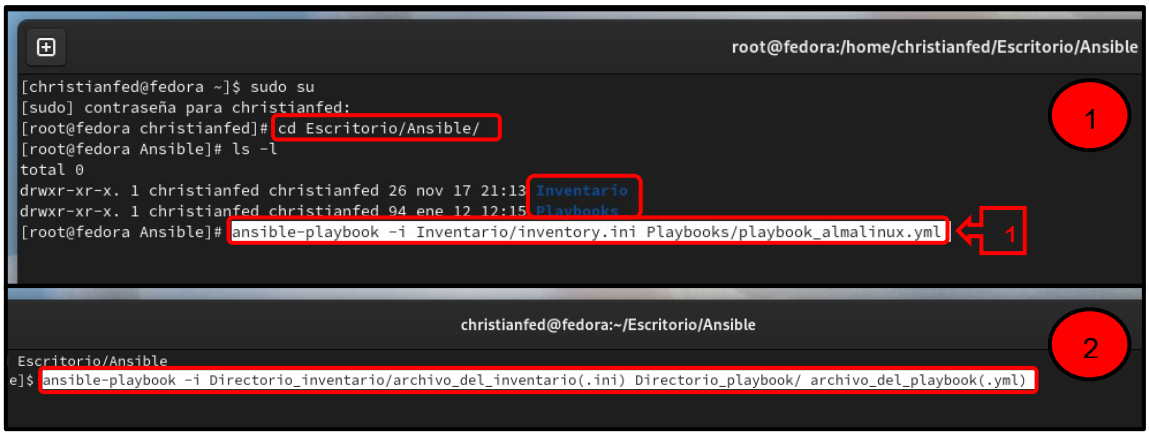

Figura 3.75 Comando para la ejecución del *playbook*

Lo primero que se presenta es información acerca del propósito del *playbook* y una indicación sobre las vulnerabilidades que serán solventadas de forma dinámica. En la Figura 3.76 se muestran los detalles de la ejecución.

```
[root@fedora Ansible]# ansible-playbook -i Inventario/inventory.ini Playbooks/playbook_almalinux.yml

"Después de realizar un escaneo previo con SCAP Workbench en el sistema operativo Alma Linux 9.2, siguiendo el marco de referencia
CIS Level 1 - Server se identificaron vulnerabilidades. Para abordar estos problemas de seguridad, se aplicarán los cambios sugeridos por este.
Leer detenidamente las recomendaciones de los parámetros sugeridos para cada problema por esta herramienta para solucionarlos.

A continuación, se presentan algunas de estas vulnerabilidades que SCAP Workbench detectó y que serán solucionadas dinámicamente mediante este playbook."

Presione ENTER para continuar.
:
```

Figura 3.76 Información acerca de la ejecución del *playbook*

A continuación, se van presentando una a una las vulnerabilidades: el nombre, la descripción del problema, la solución, la solicitud de ingreso del dato al usuario y al final un ejemplo de cómo ingresar ese valor. Los detalles descritos se aprecian en la Figura 3.77; en esta parte de la ejecución se ingresaron los datos sugeridos por el perfil según CIS, con ello las 5 primeras vulnerabilidades serán solventadas.

```
"VULNERABILIDAD #1: Set SSH authentication attempt limit

"Este problema de seguridad se refiere a restringir el número máximo de intentos de autenticación permitidos antes de que el servidor SSH desconecte al cliente.
La solución sugerida implica permitir hasta 4 intentos de autenticación antes de desconectar el cliente. Si el número de intentos de autenticación fallidos alcanza
la mitad de este valor, es decir 2, cualquier intento adicional se registrará como fallido.

Ingrese el límite de intentos de autenticación SSH a ser permitidos (por ejemplo, 4)"
: 4

"VULNERABILIDAD #2: Set Password Maximum Age

Este problema de seguridad se refiere a la gestión de contraseñas en el sistema. El ajuste determina la cantidad máxima de días que una contraseña puede estar
en uso antes de que el sistema solicite que el usuario la cambie.
Por requisito del perfil, la solución implica que las contraseñas tendrán una duración máxima de 365 días.

Indique el número de días para establecer la duración máxima de la contraseña (por ejemplo, 365)"
: 365

"VULNERABILIDAD #3: Set SSH MaxSessions limit

Este problema de seguridad se refiere a la configuración de la cantidad de sesiones simultáneas permitidas para un usuario remoto que se conecte a este
sistema a través de SSH. La solución implica limitar el número máximo de sesiones a 10, lo que significa que un usuario no podrá tener más de 10 sesiones SSH
abiertas de forma simultánea.

Ingrese el número máximo de sesiones permitidas desde una conexión SSH (por ejemplo, 10)"
: 10

"VULNERABILIDAD #4: Set Account Expiration Following Inactivity

Este problema de seguridad se refiere a la administración de cuentas  de usuario. Esta solución implica establecer un número de días después de que una
contraseña expire hasta que una cuenta se desactive de forma permanente debido a la inactividad del usuario.
La sugerencia es 30 días, es decir, si una contraseña está a punto de caducar, quedarán 30 días hasta que la cuenta se desactive automáticamente.

Ingrese el número de días para la inactividad de la cuenta (por ejemplo, 30)"
: 30

"VULNERABILIDAD #5: Uninstall avahi Server Package

Este problema de seguridad se refiere a desinstalar del sistema los paquetes avahi-autoipd y avahi, relacionados con el servidor Avahi. Avahi es una
herramienta que implementa los protocolos de descubrimiento de servicios DNS Service Discovery (DNS-SD) y Multicast DNS (mDNS). Estos protocolos son
utilizados para el descubrimiento y la resolución de servicios en redes locales.
La recomendación implica desinstalar el servidor Avahi, ya que para la funcionalidad del sistema normalmente no se requiere de estos servicios.
Eliminar estos paquetes reduce la superficie de ataque.

Este playbook desinstalará los paquetes Avahi. ¿Desea remover Avahi del sistema? (Si/No)"
: si

"Se ha completado el ingreso de todos los datos necesarios para realizar las configuraciones de las 5 vulnerabilidades que serán solucionadas de forma dinámica.
Las 5 vulnerabilidades restantes serán resueltas mediante tareas estáticas, por lo que no se requiere la introducción de ningún dato adicional por parte del usuario."

Por favor, presione ENTER para continuar con el proceso de ejecución y aplicar los cambios de endurecimiento al sistema Alma Linux.
:
```

Figura 3.77 Ingreso de datos, vulnerabilidades solventadas dinámicamente

Como se estableció en el *playbook,* las 5 primeras vulnerabilidades requieren de la interacción del usuario para ser solventadas, los 5 restantes no. Para continuar con el proceso de ejecución se dio “Enter”. En la Figura 3.78 se puede visualizar las salidas asociadas a cada tarea. La primera tarea “Gathering Facts” se ejecuta de forma predeterminada y sirve para recopilar información del sistema en el que se despliega el *playbook*, así garantiza que el *playbook* sea compatible con las características de Alma

Linux, antes de realizar cualquier otra tarea. El “ok: [cliente1]” indica que la tarea se ejecutó correctamente.

En las demás salidas asociadas a cada tarea que se muestran de igual manera en la Figura 3.78, se puede ver de forma general que las configuraciones programadas en cada una fueron realizadas para el cliente 1 y se realizaron los cambios “changed:”

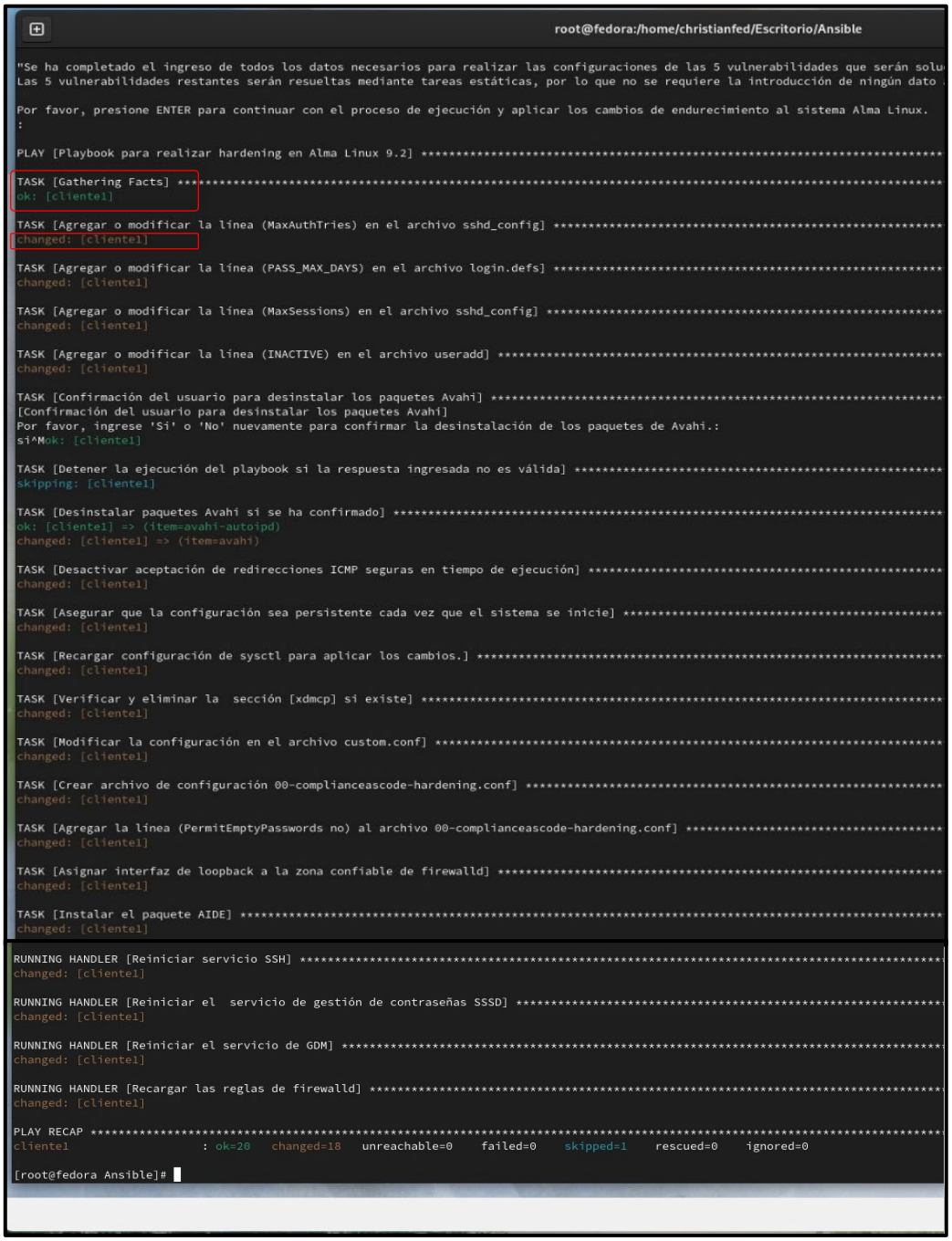

Figura 3.78 Salidas asociadas a las tareas del *playbook*

En la sección 1 de la Figura 3.79 se muestra un resumen de la ejecución para el cliente 1, proporcionado por Ansible. Se puede visualizar que: 20 tareas se ejecutaron

correctamente “ok=20”, 18 tareas hicieron algún cambio en el sistema “changed=18”, no hubo nodos inalcanzables “unreachable=0”, ninguna tarea falló “failed=0”, se omitió 1 tarea durante la ejecución “skipped=1” y no existieron tareas ignoradas “ignored=0”.

Respecto a la tarea que se omitió durante la ejecución, la cual se puede observar en la sección 2 de la Figura 3.79, esta tarea se relaciona con la vulnerabilidad número 5. El fin de esta tarea es desinstalar los paquetes vinculados con el servidor Avahi y se omitió debido a que solo se debía ejecutar para detener el avance del *playbook*, en el caso de que el usuario ingrese una respuesta no válida “si” o “no”. Pero como se ingresó “si” se omitió la tarea.

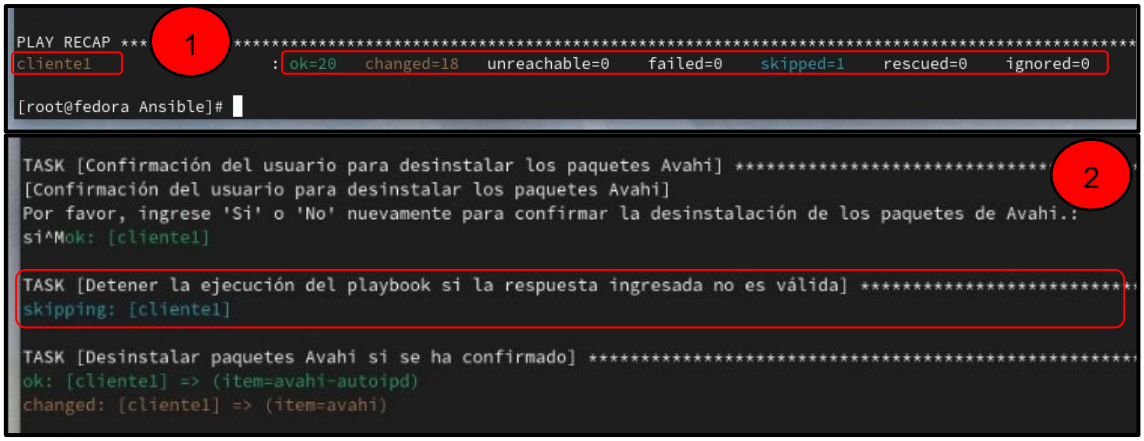

Figura 3.79 Resumen de la ejecución y tarea omitida

Obtención del reporte final por medio de SCAP *Workbench*

Para obtener el reporte final se ejecutó nuevamente SCAP *Workbench* desde el menú de aplicaciones en el escritorio de Alma Linux. De la misma manera que se realizó el análisis para obtener el reporte inicial, se siguieron los mismos pasos: al aparecer el cuadro de diálogo se seleccionó el sistema operativo, después se escogió el perfil que toma como base el Marco de referencia CIS nivel 1 para servidor y se inició el análisis.

En la Figura 3.80 se puede apreciar el reporte final. Observando los resultados se determina que: se analizaron un total de 252 reglas, de las cuales 153 pasaron, lo que significa que se ajustaron al perfil, 98 fallaron y en 1 se obtuvo otro resultado.

Con relación a la gravedad de las 98 reglas que no cumplieron con el perfil, la herramienta las categorizó según los niveles que se detallan a continuación:

- 5 reglas se categorizan como "other" (otro).
- 6 reglas se categorizan como "low" (bajo).
- 85 reglas fueron categorizadas como "medium" (medio).

- 2 reglas fueron categorizadas como "high" (alto).

Respecto a la puntuación general del cumplimiento, se observa que la puntuación general del sistema ahora es de 74.76 %.

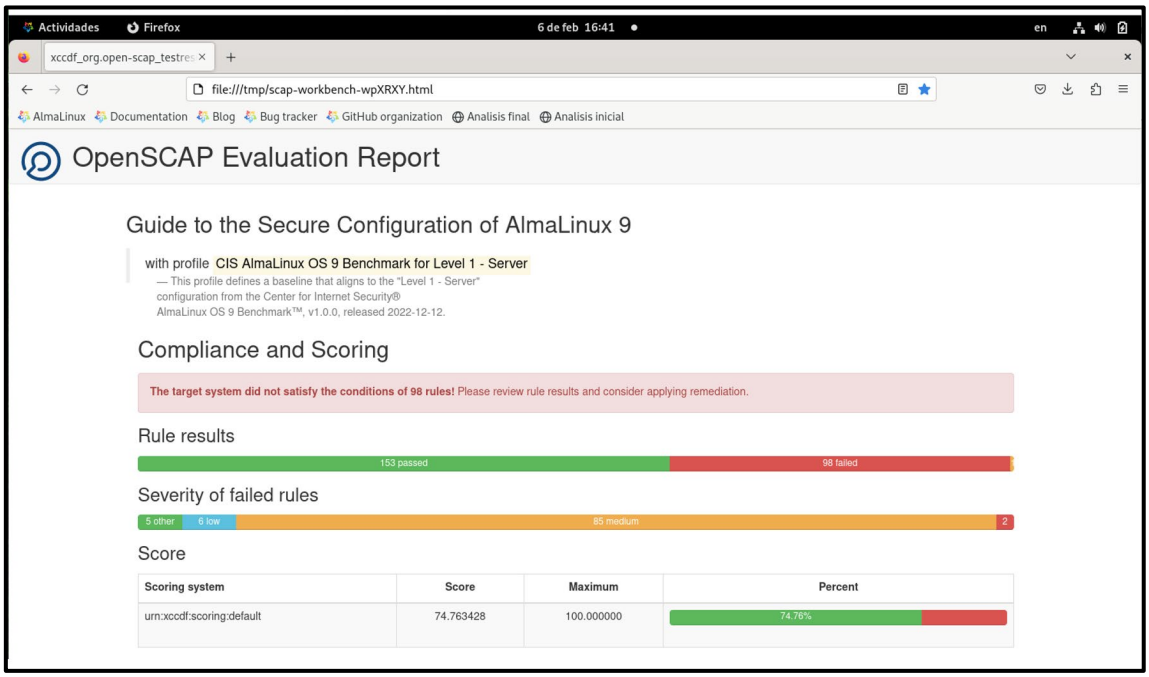

Figura 3.80 Resultados del reporte final

En la Figura 3.81 se detallan las 10 vulnerabilidades que han sido resueltas por medio de las recomendaciones generadas por SCAP *Workbench*, ya que luego fueron implementadas en el sistema operativo de servidor a través del *playbook*. Estas vulnerabilidades habían sido previamente identificadas como fallidas, tal como se observó en el cuadro de la Figura 3.13.

Title	Severity	Result
Disable XDMCP in GDM	high	pass
Disable SSH Access via Empty Passwords	high	pass
Disable Kernel Parameter for Accepting Secure ICMP Redirects on all IPv4 Interfaces	medium	pass
Set Account Expiration Following Inactivity	medium	pass
Set SSH authentication attempt limit	medium	pass
Configure Firewalld to Trust Loopback Traffic	medium	pass
Set SSH MaxSessions limit	medium	pass
Set Password Maximum Age	medium	pass
Install AIDE	medium	pass
Uninstall avahi Server Package	medium	pass

Figura 3.81 Reglas solventadas con el *playbook*

3.4 Verificación de los resultados obtenidos

Para la verificación de los resultados obtenidos, se puede ver en las secciones 1 y 2 Figura 3.82 el reporte del análisis inicial y final respectivamente. Observando los resultados de los reportes se puede decir que, de las 252 reglas que se evaluaron, el reporte final muestra que 153 reglas cumplen con el perfil mientras que 98 fallaron. A comparación de las 143 reglas que cumplían y 108 fallaban en el reporte inicial.

En la puntuación general del sistema que se evalúa respecto al porcentaje de reglas que se cumplen, este subió al 74.76 % en el reporte final, respecto al 70.88% del reporte inicial. Este incremento refleja un impacto positivo de las acciones implementadas mediante el *playbook* para el fortalecimiento de la seguridad del sistema.

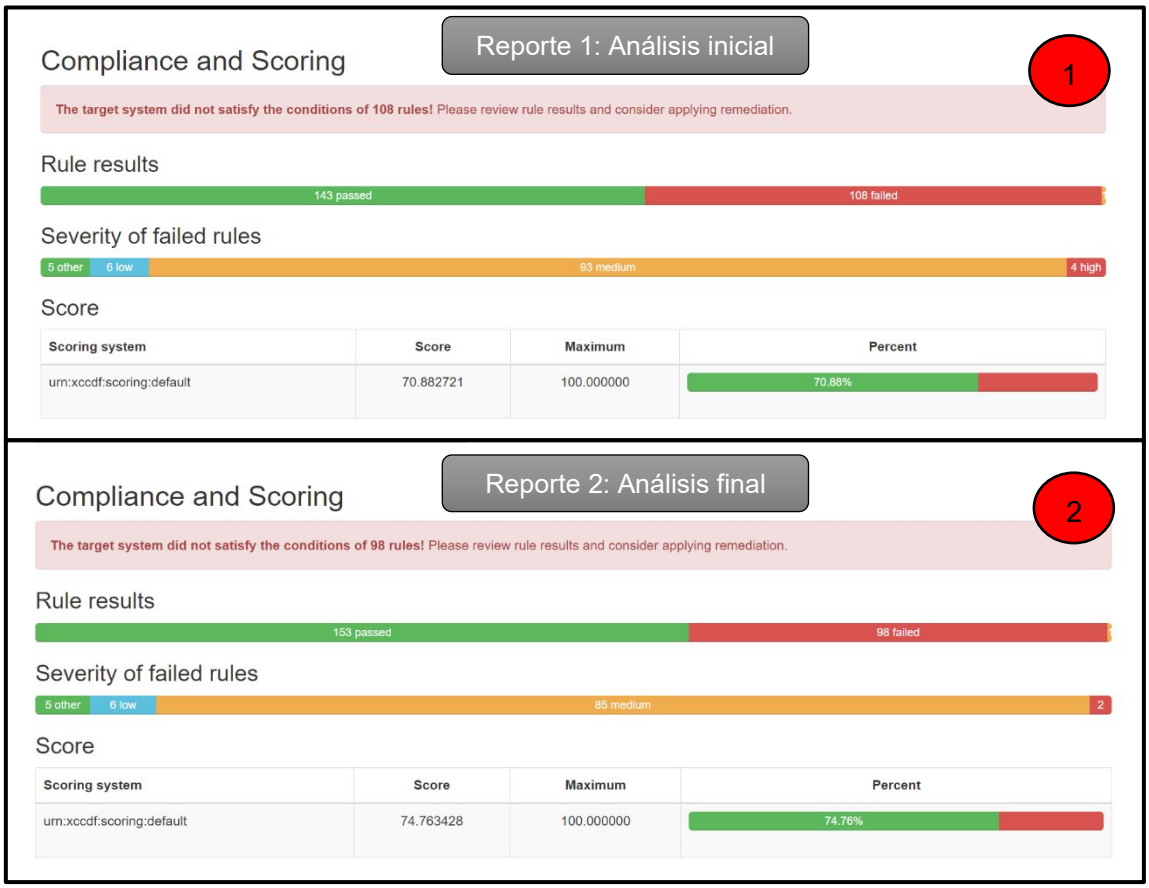

Figura 3.82 Comparación de Resultados del reporte inicial y final

A continuación, se detallan las modificaciones realizadas por las tareas del *playbook*, después de su ejecución en el sistema operativo de servidor Alma Linux. Los cambios en el sistema se dan por las recomendaciones de SCAP *Workbench*, con el propósito de que cumplan 10 vulnerabilidades de las 108 (dentro de las 252 evaluadas) que no cumplían con el marco de referencia CIS nivel 1.

En primer lugar, se muestran los resultados de las vulnerabilidades abordadas de forma dinámica, después los resultados correspondientes de las vulnerabilidades solventadas estáticamente.

Resultados de las vulnerabilidades solucionadas de forma dinámica

En la Tabla 3.2 se presentan las 5 vulnerabilidades abordadas de manera dinámica de las 108 identificadas en el reporte del análisis inicial, Figura 3.12. De las 108 vulnerabilidades, 93 fueron clasificadas por la herramienta con un nivel de riesgo medio. Las 5 vulnerabilidades presentadas en la tabla corresponden a ese mismo nivel y como se indica han sido superadas; además, incluye los encabezados: nombre de la vulnerabilidad y la razón fundamental en donde se da una descripción del motivo de la recomendación, explicando cómo puede afectar a la seguridad del sistema. La razón fundamental de cada vulnerabilidad proviene de la información que la herramienta SCAP *Workbench* generó como “Rationale”, vista en la Figura 3.43.

Tabla 3.2 Vulnerabilidades solventadas de forma dinámica

No.	Nombre de la vulnerabilidad	Razón fundamental	Nivel de riesgo / Resultado
1	*Set* SSH *authentication attempt limit*	Reducir el número de intentos de autenticación permitidos a un nivel bajo disminuye significativamente la probabilidad de éxito en ataques de fuerza bruta dirigidos al servidor SSH.	Medio / Superada
2	*Set Password Maximum Age*	Esta modificación asegura que los usuarios estén obligados a cambiar sus contraseñas, ya que el sistema operativo limita su vida útil. El cambio regular de las contraseñas ayuda a mitigar el riesgo de que estas sean descifradas con el tiempo.	Medio / Superada
3	*Set* SSH *MaxSessions limit*	Establecer el número máximo de sesiones abiertas al servidor SSH, permitidas simultáneamente desde una única conexión, protege al sistema contra ataques de denegación de servicio (DoS).	Medio / Superada

No.	Nombre de la vulnerabilidad	Razón fundamental	Nivel de riesgo / Resultado
4	*Set Account Expiration Following Inactivity*	Establecer la expiración de una cuenta inactiva ayuda a mitigar el riesgo de acceso no autorizado. Las cuentas de usuario inactivas representan un peligro para los sistemas, ya que los atacantes pueden explotar estas identificaciones y obtener acceso no detectado al sistema.	Medio / Superada
5	*Uninstall* avahi *Server Package*	Desinstalar los paquetes asociados al servidor Avahi es recomendable, ya que reduce la superficie de ataque del sistema. Los protocolos que implementa esta herramienta son utilizados para el descubrimiento y la resolución de servicios en redes locales, pero para el funcionamiento del sistema normalmente no son necesarios estos servicios.	Medio / Superada

Seguidamente, se detallan algunos de los cambios realizados por el *playbook* respecto a las 5 vulnerabilidades mostradas en la Tabla 3.2, para que cumplan con el marco de referencia del análisis inicial.

- ***Set* SSH *authentication attempt limit*:**
  En la Figura 3.83 se puede observar el archivo de configuración "sshd_config". En la sección 1 de la misma figura se muestra el parámetro "MaxAuthTries" con 6, además no está habilitado. Mientras que en la sección 2 se muestra el parámetro ya habilitado con el valor de 4, que son los intentos de autenticación permitidos en SSH.

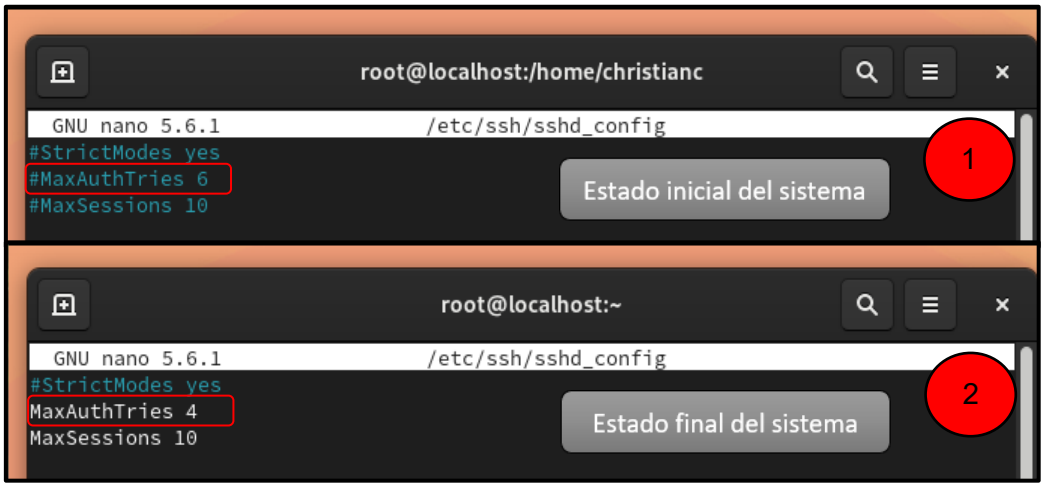

Figura 3.83 Cambios, archivo de configuración "sshd_config"

- ***Set Password Maximum Age**:*

  En la Figura 3.84 se visualiza el archivo de configuración “login.defs”. En la sección 1, el parámetro “PASS_MAX_DAYS” se muestra con un valor de "99999", que determina el número de días de duración para las contraseñas de los usuarios del sistema. Por otra parte, en la sección 2, dicho valor disminuye a "365", con el objetivo de concordar con la recomendación del perfil del análisis.

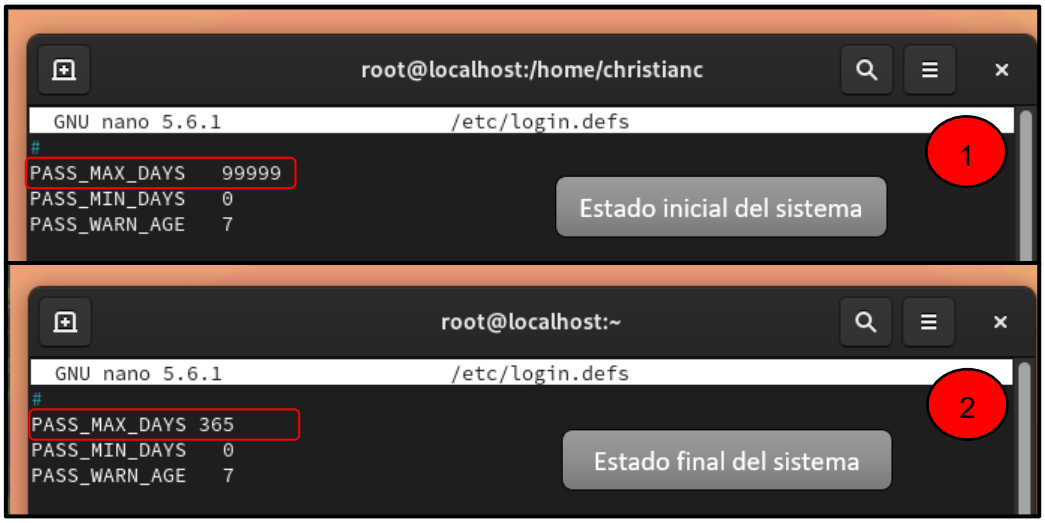

Figura 3.84 Cambios de configuración, archivo “login.defs”

- ***Set* SSH *MaxSessions limit**:*

  La Figura 3.85 presenta el archivo de configuración "sshd_config". En su primera sección, se muestra el parámetro “MaxSessions” con un valor de 10 y sin habilitar. Por el contrario, en la sección 2, el mismo parámetro está activado con un valor de 10, que indica el límite máximo de sesiones simultáneas a través del servicio SSH.

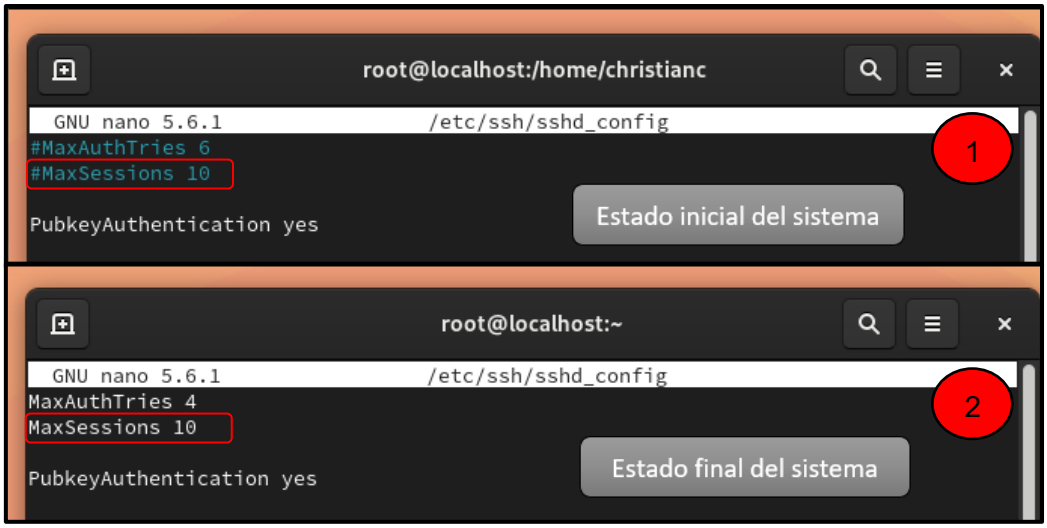

Figura 3.85 Ajustes de configuración en "sshd_config"

- ***Set Account Expiration Following Inactivity***:

  Se puede observar en la Figura 3.86 el archivo de configuración “useradd”. En la sección 1, se muestra el parámetro “INACTIVE” con un valor de "-1", para establecer el número de días para inhabilitar una cuenta por inactividad del usuario. Sin embargo, en la sección 2, el mismo parámetro está cambiado a un valor de "30", cumpliendo así con el perfil de seguridad.

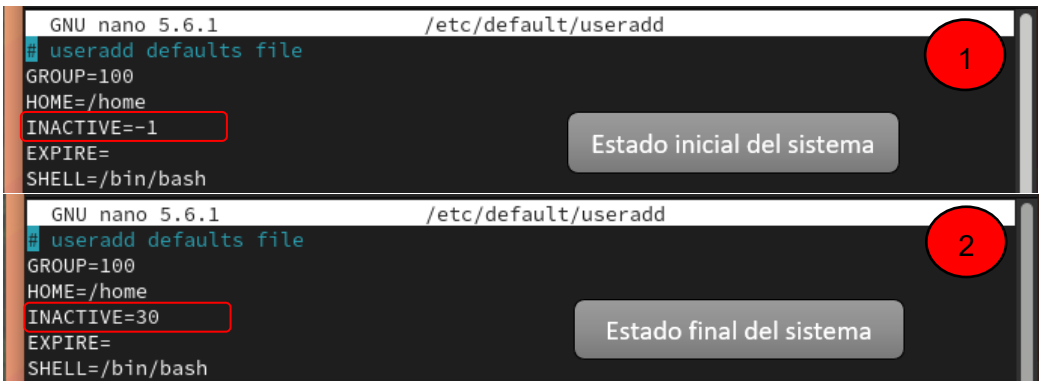

Figura 3.86 Cambios de configuración, archivo “useradd”

- ***Uninstall* avahi *Server Package***:

  La Figura 3.87 está relacionada con la desinstalación del servidor "avahi". En la sección 1, se muestra la versión instalada del servidor. En la sección 2, el mismo servicio se encuentra eliminado, reflejando la desinstalación exitosa mediante el *playbook*. Se hizo uso de los comandos 1 y 2 en la terminal de Alma Linux para visualizar el historial de movimientos acerca de instalación y desinstalación en el sistema.

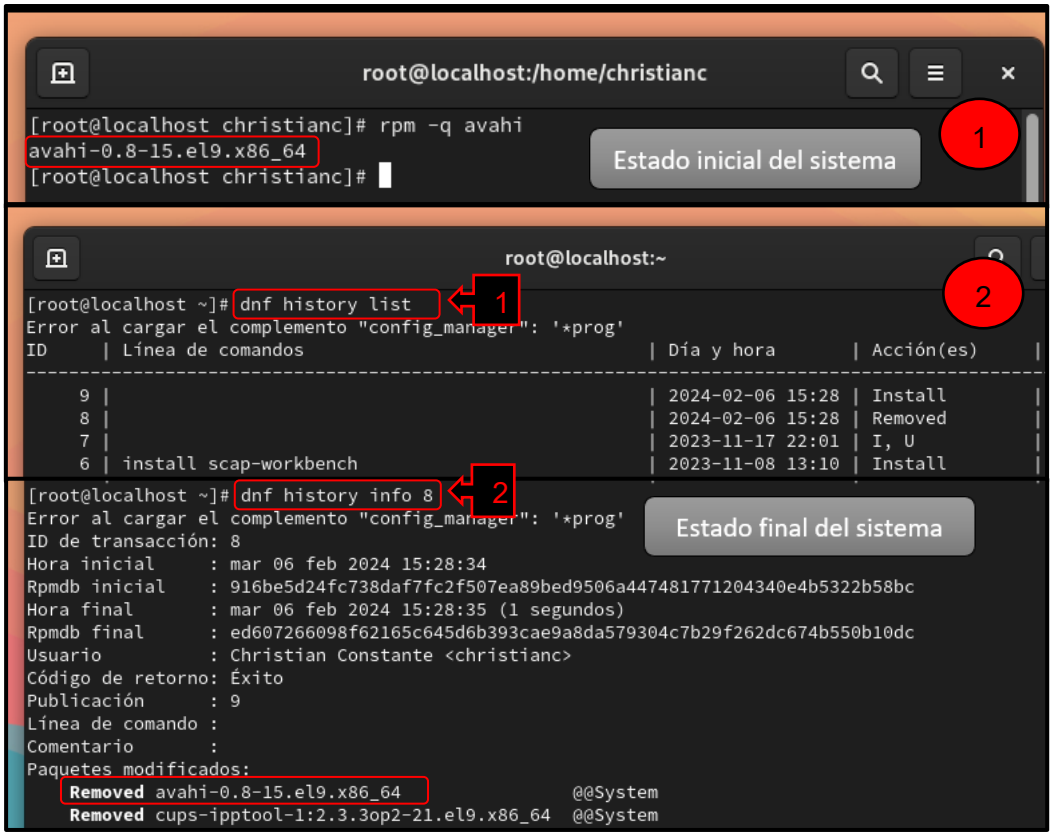

Figura 3.87 Verificación de la desinstalación, servidor Avahi

Resultados de las vulnerabilidades solucionadas de forma estática

En la Tabla 3.3 se incluye las cinco vulnerabilidades que han sido solventadas de manera estática. Al decir de manera estática, quiere decir que los cambios para resolver estas vulnerabilidades no dependen de la intervención del usuario durante la ejecución del *playbook* para ingresar datos. A diferencia de las vulnerabilidades solventadas dinámicamente, donde se requiere la participación del usuario para corregir el problema.

Como se evidenció en el reporte del análisis inicial, presentado en la Figura 3.12, de las 108 reglas fallidas, 93 fueron clasificadas con un nivel de riesgo medio, mientras que 4 reglas fueron de nivel de riesgo alto. De esas 108 reglas fallidas, se seleccionaron otras 5 reglas para solventarlas de forma estática, de las cuales, como se observa en la Tabla 3.3, las vulnerabilidades 6, 9 y 10 tienen un nivel de riesgo medio, mientras que las vulnerabilidades 7 y 8 presentan un nivel de riesgo alto.

En la Tabla 3.3 aparte del nivel de riesgo, también se muestra el nombre de la vulnerabilidad, la razón fundamental del cambio y el resultado. Con relación al resultado todos los problemas de seguridad han sido superados.

Tabla 3.3 Vulnerabilidades solventadas de forma estática

No.	Nombre de la vulnerabilidad	Razón fundamental	Nivel de riesgo / Resultado
6	*Disable Kernel Parameter for Accepting Secure* ICMP *Redirects on all* IPv4 *Interfaces*	Aceptar redirecciones ICMP seguras debe desactivarse, ya que mantener este parámetro activo innecesariamente representa un riesgo de seguridad. Es recomendable desactivarlo a menos de que exista alguna necesidad especifica que justifique su uso en el sistema.	Medio / Superada
7	*Disable* XDMCP *in* GDM	Desactivar XDMCP en GDM es importante, ya que XDMCP es un protocolo que proporciona acceso remoto no cifrado a través de GDM. Esto implica que, si un usuario con privilegios inicia sesión utilizando XDMCP, existe el riesgo de que la contraseña y las interacciones sean interceptadas por un atacante, ya que las pulsaciones de teclas transmitidas a través de esa sesión estarían en texto plano.	alto / Superada
8	*Disable* SSH *Access via Empty Passwords*	Restringir el inicio de sesión SSH mediante contraseñas vacías proporciona una capa adicional de seguridad en el sistema respecto al acceso remoto, ya que no se permite la conexión SSH sin una contraseña.	alto / Superada
9	*Configure* Firewalld *to Trust Loopback Traffic*	Configurar el Firewalld para confiar en el tráfico de bucle local es fundamental para el funcionamiento de algunas aplicaciones y servicios del sistema. Este tráfico se produce entre procesos en la máquina y al confiar se garantiza que este tipo de tráfico no sea bloqueado por el firewall.	Medio / Superada
10	*Install* AIDE	La instalación del paquete AIDE es imprescindible para fortalecer la seguridad del sistema. AIDE es una herramienta especializada en el monitoreo y detección de modificaciones no autorizadas en los archivos del sistema, lo cual resulta fundamental para garantizar su integridad y protegerlo contra posibles amenazas.	Medio / Superada

A continuación, se exponen algunas de las modificaciones ejecutadas mediante el *playbook* respecto a 4 de las 5 vulnerabilidades presentadas en la Tabla 3.3, con el fin de ajustarlas con el marco de referencia del análisis. Con relación a la vulnerabilidad número 9 que no se encuentra presentada, como se expuso en la explicación dada en la Figura 3.70, la solución solo implica la ejecución de dos comandos sugeridos por la herramienta de escaneo. Por lo que los comandos fueron ejecutados con dos tareas, una tarea para confiar en el tráfico de bucle local y un *handler* para recargar las reglas del *firewall*. Además, en las salidas asociadas a cada tarea mostradas en la Figura 3.78 se pudieron evidenciar los cambios efectuados por estas en el sistema Alma Linux.

- ***Disable Kernel Parameter for Accepting Secure* ICMP *Redirects on all* IPv4 *Interfaces*:**

  La Figura 3.88 presenta el archivo de configuración "99-sysctl.conf". En la sección 1, se muestra la estructura del archivo de configuración sin la activación de ningún parámetro. Mientras tanto, en la sección 2, se implementa y habilitada la línea de texto visible dentro del recuadro azul, que sirve para que la configuración respecto a deshabilitar el parámetro del *kernel* se mantenga constante cada vez que el sistema arranque.

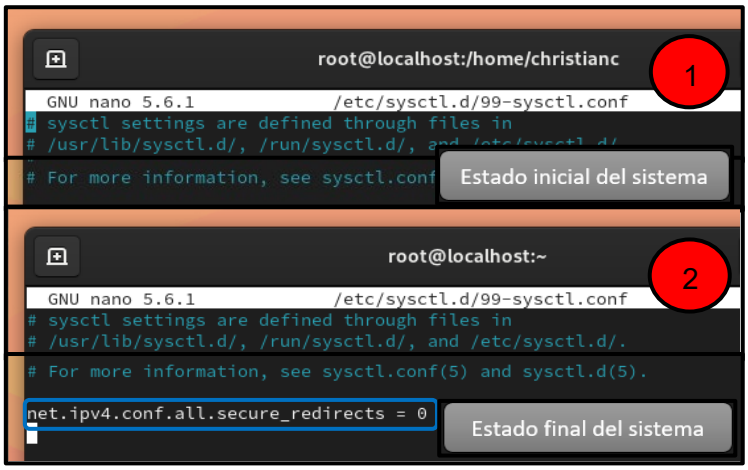

Figura 3.88 Cambios de configuración, archivo "99-sysctl.conf"

- ***Disable* XDMCP *in* GDM*:***

  Se puede observar en la Figura 3.89 el archivo "custom.conf". En la primera sección, se encuentra la configuración inicial que incluye la sección [xdmcp]. En la segunda sección, se reflejan las acciones realizadas mediante el *playbook*, donde se ha eliminado y colocado nuevamente la sección [xdmcp] utilizando el módulo Blockinfile, pero además se ha colocado bajo la sección la condición "Enable=false" para desactivar el protocolo XDMCP.

Figura 3.89 Cambios de configuración, archivo "custom.conf"

- ***Disable* SSH *Access via Empty Passwords***:

  La Figura 3.90 muestra el archivo de configuración "00-complianceascode-handlreha.conf". En la sección 1, se puede apreciar que el archivo está en blanco, la razón es debido a que el archivo no existía inicialmente en el sistema, se creó posteriormente con una tarea del *playbook*. Por otro lado, en la sección 2, dicho archivo de configuración esta creado, además se colocó el parámetro visible dentro del recuadro azul, su función es deshabilitar el ingreso al servicio SSH para cuentas configuradas con contraseñas vacías.

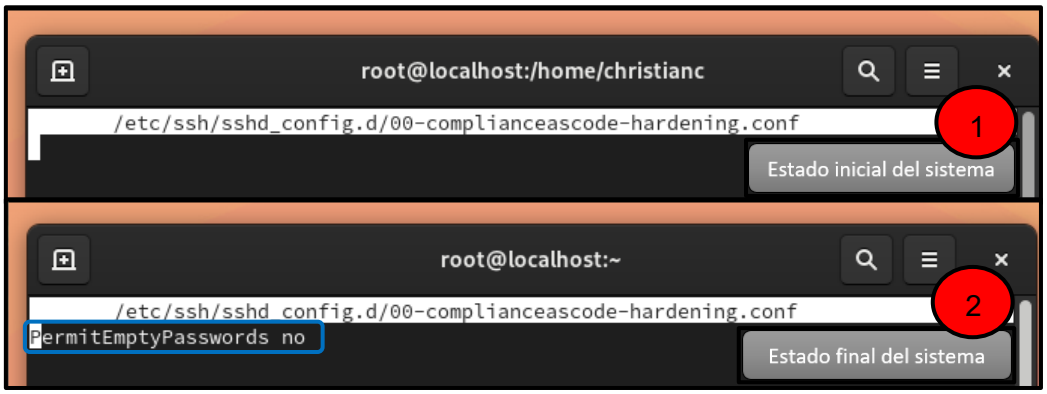

Figura 3.90 Ajustes, archivo "00-complianceascode-handlreha.conf"

- ***Install* AIDE***:*

  En la Figura 3.91 se presenta la instalación del servidor AIDE. En la sección 1, al ejecutar el comando "aide --version", no se mostró la versión debido a que el servicio aún no estaba instalado. Sin embargo, en la sección 2, se puede observar la instalación del servidor al utilizar los comandos 1 y 2 para revisar en detalle la instalación.

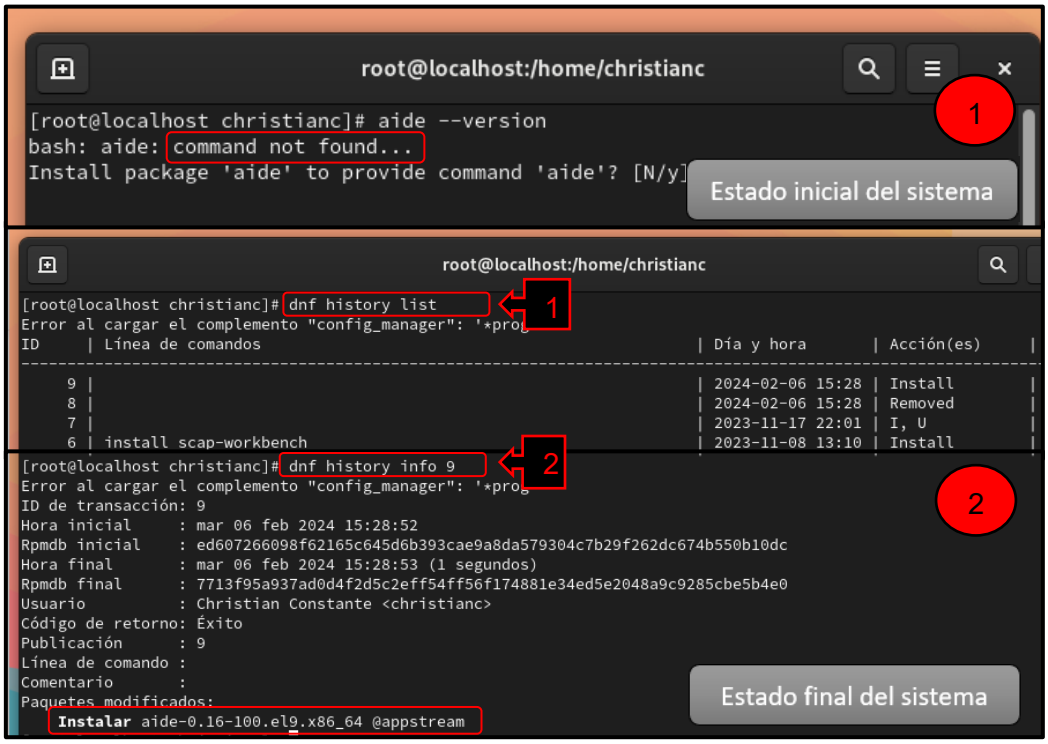

Figura 3.91 Verificación de la instalación, servidor AIDE

Con los datos obtenidos en los análisis efectuados por SCAP *Workbench* y los resultados expuestos de las modificaciones hechas por el *playbook* en el sistema operativo de servidor Alma Linux, se puede evidenciar se logró solventar las 10 reglas seleccionadas que no habían pasado la prueba del perfil de seguridad CIS nivel 1, lo que ha contribuido significativamente a mejorar la seguridad del sistema operativo de servidor Alma Linux.

4 CONCLUSIONES

- La implementación de *hardening* mediante la herramienta de DevOps Ansible ha sido llevada a cabo con éxito. A través de la automatización proporcionada por Ansible, se logró fortalecer la seguridad del sistema operativo de servidor de base *Red Hat*, Alma Linux de manera eficiente. Las tareas de configuración y endurecimiento, creadas en el *playbook* a partir de las recomendaciones de

seguridad obtenidas mediante SCAP *Workbench*, se ejecutaron correctamente sin errores. Con lo que se puede decir que este enfoque es efectivo para alcanzar un nivel óptimo de seguridad en un sistema operativo de servidor, ya que ofrece una base sólida para mitigar posibles vulnerabilidades.

- El análisis inicial efectuado en el sistema operativo de servidor sin políticas de seguridad, realizado mediante la herramienta de escaneo de configuración y vulnerabilidades SCAP *Workbench*, permitió identificar los puntos de debilidad en la seguridad del sistema, proporcionando una base sólida para la implementación de medidas de *hardening*.

- Las recomendaciones de seguridad proporcionadas por SCAP *Workbench*, bajo el perfil CIS Nivel 1 para servidor, han sido fundamentales para identificar y abordar vulnerabilidades comunes en el sistema operativo de servidor Alma Linux. Estas recomendaciones han servido como una guía valiosa para garantizar el cumplimiento de marcos de referencia de seguridad reconocidos como CIS.

- La instalación exitosa de Ansible en el nodo de control Fedora, haciendo uso de un entorno virtual como el hipervisor VirtualBox, facilitó la gestión centralizada y la automatización de las tareas para fortalecer la seguridad del sistema operativo de servidor Alma Linux, cumpliendo así con el requerimiento de disponer de la herramienta Ansible y garantizar que funcione adecuadamente en el nodo de control para posteriormente aplicar el proceso de *hardening*.

- El proceso de conexión SSH, con autenticación de clave pública para tener acceso al nodo controlado desde el nodo de control, es fundamental para establecer una conexión segura y autorizada durante las operaciones de administración remota. Este método de conexión, implementado con éxito durante el proyecto, ha eliminado la necesidad de ingresar la contraseña del usuario del Alma Linux en cada conexión, proporcionando un flujo de información seguro y eficiente entre el nodo de control y el nodo controlado.

- La comunicación establecida mediante SSH entre el nodo de control Fedora y el nodo controlado Alma Linux permitió la ejecución correcta del *playbook* diseñado para fortalecer la seguridad del sistema Alma Linux. Además, gracias a la utilización del proceso de conexión SSH con autenticación de clave pública se proporcionó mayor protección en la implementación, al ser más difíciles de adivinar las claves generadas en relación con las contraseñas de usuario

usadas en el método de conexión de autenticación por contraseña, lo que contribuye a fortalecer la seguridad del entorno.

- La creación y configuración del archivo de inventario ha sido exitosa, permitiendo establecer la conexión con el nodo controlado de manera efectiva. El formato .ini utilizado para el inventario proporciona una estructura clara y organizada, facilitando la gestión de la configuración del nodo controlado. Además, todos los parámetros que conforman su estructura han sido correctamente especificados, ya que se logró verificar su funcionamiento y conexión con el nodo controlado mediante el comando PING de Ansible.

- El uso del directorio "/etc/ansible/hosts", ruta predeterminada por Ansible para el inventario, puede considerarse la elección más común para almacenar los inventarios de Ansible. Sin embargo, se ha demostrado que la organización de los archivos de inventario y *playbook* en otros directorios y subdirectorios diferentes al predeterminado, también es una opción válida que ayuda a mejorar la organización del proyecto.

- La comparación entre los reportes inicial y finales obtenidos mediante la herramienta de escaneo de vulnerabilidades SCAP *Workbench*, ha sido fundamental para evaluar el impacto de las medidas de *hardening* implementadas. El análisis detallado de las vulnerabilidades solucionadas ha evidenciado una mejora significativa en la seguridad del sistema operativo de servidor, proporcionando una visión clara de estos problemas de seguridad que han sido abordados con éxito, validando así la eficacia de las medidas implementadas.

5 RECOMENDACIONES

- Tomando como base la efectividad demostrada del proceso de conexión SSH con autenticación de clave pública, para asegurar un acceso seguro y autorizado al nodo controlado desde el nodo de control, se recomienda implementar este método de autenticación en los proyectos futuros relacionados con Ansible. Esta práctica, aunque no es requerida por Ansible, ya que es suficiente la conexión SSH con autenticación por contraseña, representa una medida adicional de seguridad para garantizar la integridad y confidencialidad de las comunicaciones durante la administración remota.

- Se recomienda utilizar un inventario estático en proyectos similares de administración remota con Ansible. El inventario estático es la mejor opción en entornos donde los nodos controlados se conocen previamente y son pocos, o solo uno como en el presente proyecto. A diferencia del inventario dinámico, que requiere *scripts* para gestionar y actualizar la información de los nodos, el inventario estático se define en un archivo .ini y las actualizaciones manuales para agregar un nuevo nodo son sencillas. Lo que simplifica la gestión y proporciona una estructura clara, especialmente en entornos de prueba pequeños.
- Para realizar pruebas de funcionamiento del inventario, se recomienda utilizar herramientas como el módulo PING de Ansible, así se garantiza la correcta configuración y disponibilidad de los nodos controlados. Esta prueba ayuda a identificar posibles problemas en la configuración del inventario.
- Considerando la necesidad de establecer comunicación entre máquinas virtuales en VirtualBox durante el proceso de instalación y configuración de Ansible, se recomienda utilizar la configuración de red NAT. La configuración de red NAT permite que las máquinas virtuales estén en la misma red interna, lo que facilita la comunicación entre ellas y permite el acceso a Internet. Además, cuenta con la opción de habilitar el DHCP.
- Para modificar bloques de texto en archivos de configuración por medio de *playbooks* de Ansible, se recomienda identificar previamente las líneas de texto específicas que deben ser editadas. La mejor opción es primero eliminarlas del archivo de configuración utilizando tareas que contengan módulos como “lineinfile”, con el estado *absent*. Posteriormente, se puede proceder a insertar del bloque de texto completo utilizando el módulo “blockinfile”, asegurando así que la configuración deseada se aplique de manera efectiva y sin redundancias.
- Para garantizar la correcta ejecución y funcionalidad de los *playbooks* de Ansible, se recomienda utilizar el comando “ansible-playbook --syntax-check nombre_del_playbook.yml” antes de ejecutarlos. Este comando permite verificar la sintaxis del código del *playbook* y detectar posibles errores de formato, así como la utilización incorrecta de parámetros dentro de los módulos de Ansible. Al realizar esta verificación previa, se pueden identificar y corregir los errores asegurando un despliegue exitoso de las tareas en los nodos controlados.

6 REFERENCIAS BIBLIOGRÁFICAS

[1] «¿Qué es Ansible y para qué sirve?,» Red Hat - We make open source technologies for the enterprise, 9 Noviembre 2023. [En línea]. Available: https://www.redhat.com/es/technologies/management/ansible/what-is-ansible. [Último acceso: 11 Noviembre 2023].

[2] «Ansible para principiantes: conceptos básicos de Ansible y cómo funciona,» Geekflare, 21 8 2023. [En línea]. Available: https://geekflare.com/es/ansible-basics/. [Último acceso: 11 11 2023].

[3] I. Lemus, «¿Qué es ANSIBLE?,» Conocimiento Libre, 5 Noviembre 2019. [En línea]. Available: https://conocimientolibre.mx/que-es-ansible/. [Último acceso: 16 Noviembre 2023].

[4] «Installing Ansible,» Ansible Documentation, 5 Noviembre 2023. [En línea]. Available: https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html. [Último acceso: 11 Noviembre 2023].

[5] S. De Luz, «Los mejores escáner de vulnerabilidades gratis para hackers,» RedesZone, 1 Octubre 2023. [En línea]. Available: https://www.redeszone.net/tutoriales/seguridad/mejores-escaner-vulnerabilidades-gratis-hacker/. [Último acceso: 11 Noviembre 2023].

[6] H. Ashtari, «What Is Security Content Automation Protocol (SCAP)?,» Spiceworks, 31 Mayo 2023. [En línea]. Available: https://www.spiceworks.com/it-security/vulnerability-management/articles/what-is-security-content-automation-protocol-scap/. [Último acceso: 11 Noviembre 2023].

[7] «Using SCAP Workbench Red Hat Enterprise Linux 6,» Red Hat Customer Portal, [En línea]. Available: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/security_guide/sect-using_scap_workbench. [Último acceso: 12 Noviembre 2023].

[8] «Escanear el sistema con un perfil personalizado utilizando SCAP Workbench Red Hat Enterprise Linux 8,» Red Hat Customer Portal, [En línea]. Available:

https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/security_hardening/scanning-the-system-with-a-customized-profile-using-scap-workbench_scanning-the-system-for-configuration-compliance-and-vulnerabilities. [Último acceso: 17 Noviembre 2023].

[9] K. Gauza, it-swarm-es.com, 16 Enero 2019. [En línea]. Available: https://www.it-swarm-es.com/es/python/instalacion-del-paquete-ansible-python-en-windows/806388595/. [Último acceso: 26 Noviembre 2021].

[10] «Escanear el sistema con un perfil personalizado utilizando SCAP Workbench,» Red Hat Customer Portal, [En línea]. Available: https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/security_hardening/scanning-the-system-with-a-customized-profile-using-scap-workbench_scanning-the-system-for-configuration-compliance-and-vulnerabilities. [Último acceso: 17 Noviembre 2023].

[11] «Qualys Threat Protection: IT Threat Management Tool,» Qualys, Inc., 2023. [En línea]. Available: https://www.qualys.com/apps/threat-protection/. [Último acceso: 18 Noviembre 2023].

[12] N. Gutierrez, «Marcos de Ciberseguridad: La Guía Definitiva,» Prey, 4 Junio 2020. [En línea]. Available: https://preyproject.com/es/blog/marcos-de-ciberseguridad-la-guia-definitiva. [Último acceso: 18 Noviembre 2023].

[13] N. Gibbon, «CIS Benchmark Profile Levels,» Medium, 18 Agosto 2021. [En línea]. Available: https://medium.com/pareture/cis-benchmark-profile-levels-3b5dc045635d. [Último acceso: 18 Noviembre 2023].

[14] C. Villamizar, «¿Qué es COBIT y para qué sirve?,» GlobalSuite Solutions, 25 Septiembre 2023. [En línea]. Available: https://www.globalsuitesolutions.com/es/que-es-cobit/. [Último acceso: 18 Noviembre 2023].

[15] «Instalación de Alma Linux 9,» Nettix Perú – Servicios en nube privada y soporte especialista., [En línea]. Available: https://www.nettix.com.pe/documentacion/administracion/linux-administracion/instalacion-de-alma-linux-9/. [Último acceso: 11 Noviembre 2023].

7 ANEXOS

ANEXO I: Certificado de Originalidad

ANEXO II: Enlaces

ANEXO III: Códigos Fuente

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 19 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, MEDIANTE ANSIBLE, EN UN SISTEMA OPERATIVO DE SERVIDOR LINUX DE BASE RED HAT asociado al proyecto IMPLEMENTACIÓN DE HARDENING MEDIANTELA HERRAMIENTA DE DEVOPS ANSIBLE elaborado por el estudiante CHRISTIAN PATRICIO CONSTANTE VACA 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.

Turnitin Sr Christian Constante V.pdf

Atentamente,

Gabriela Katherine Cevallos Salazar

Docente

Escuela de Formación de Tecnólogos

ANEXO II: Enlaces

El video demostrativo del Trabajo de Integración Curricular está disponible en Video Trabajo de Integración Curricular ChristianConstante.mp4 o al escanear el siguiente código QR.

Anexo II.I Código QR del video demostrativo

El reporte del análisis inicial efectuado por la herramienta de escaneo SCAP Workbench está disponible en Reporte análisis inicial_SCAPWorkbench_profile_cis_server_l1.pdf o al escanear el siguiente código QR.

Anexo III.II Código QR del análisis inicial, SCAP *Workbench*

El reporte del análisis final efectuado por la herramienta de escaneo SCAP *Workbench* está disponible en Reporte análisis final_SCAPWorkbench_profile_cis_server_l1.pdf o al escanear el siguiente código QR.

Anexo IIII.III Código QR del análisis final, SCAP *Workbench*

ANEXO III: Códigos Fuente

```
---

##############################################################################
##############################################################################
##############################################

#                Este playbook se creó tomando como base las soluciones planteadas por SCAP Workbench a partir de la evaluación

#                 de perfil CIS AlmaLinux OS 9 Benchmark para el nivel 1: servidor, con el propósito de solucionar los problemas

#                 de seguridad en algunas reglas que no pasaron la prueba.

##############################################################################
##############################################################################
##############################################


- name: Playbook para realizar hardening en Alma Linux 9.2

  hosts: alma_linux_servers

  become: true


##############################################################################
##############################################################################
##############################################

##############################################################################
##############################################################################
##############################################


  vars_prompt:


##############################################################################
##############################################################################
##############################################
```

```
- name: indicacion_general
  prompt: |
    "Después de realizar un escaneo previo con SCAP Workbench en el sistema operativo Alma Linux 9.2, siguiendo el marco de referencia
    CIS Level 1 - Server se identificaron vulnerabilidades. Para abordar estos problemas de seguridad, se aplicarán los cambios sugeridos por este.
    Leer detenidamente las recomendaciones de los parámetros sugeridos para cada problema por esta herramienta para solucionarlos.

    A continuación, se presentan algunas de estas vulnerabilidades que SCAP Workbench detectó y que serán solucionadas dinámicamente mediante este playbook."

    Presione ENTER para continuar.
  private: no

# Vulnerabilidad 1 (0) Set SSH authentication attempt limit

- name: max_auth_tries
  prompt: |
    "VULNERABILIDAD #1: Set SSH authentication attempt limit

    "Este problema de seguridad se refiere a restringir el número máximo de intentos de autenticación permitidos antes de que el servidor SSH desconecte al cliente.
```

La solución sugerida implica permitir hasta 4 intentos de autenticación antes de desconectar el cliente. Si el número de intentos de autenticación fallidos alcanza

la mitad de este valor, es decir 2, cualquier intento adicional se registrará como fallido.

Ingrese el límite de intentos de autenticación SSH a ser permitidos (por ejemplo, 4)"

private: no

Vulnerabilidad 2 (0) Set Password Maximum Age

- name: pass_max_days

prompt: |

"VULNERABILIDAD #2: Set Password Maximum Age

Este problema de seguridad se refiere a la gestión de contraseñas en el sistema. El ajuste determina la cantidad máxima de días que una contraseña puede estar

en uso antes de que el sistema solicite que el usuario la cambie.

Por requisito del perfil, la solución implica que las contraseñas tendrán una duración máxima de 365 días.

Indique el número de días para establecer la duración máxima de la contraseña (por ejemplo, 365)"

private: no

Vulnerabilidad 3 (0) Set SSH MaxSessions limit

```
  - name: max_sessions
    prompt: |

      "VULNERABILIDAD #3: Set SSH MaxSessions limit

      Este problema de seguridad se refiere a la configuración de la cantidad de sesiones simultáneas permitidas para un usuario remoto que se conecte a este
      sistema a través de SSH. La solución implica limitar el número máximo de sesiones a 10, lo que significa que un usuario no podrá tener más de 10 sesiones SSH
      abiertas de forma simultánea.

      Ingrese el número máximo de sesiones permitidas desde una conexión SSH (por ejemplo, 10)"
    private: no


# Vulnerabilidad 4 (0) Set Account Expiration Following Inactivity

  - name: inactive_days
    prompt: |

      "VULNERABILIDAD #4: Set Account Expiration Following Inactivity

      Este problema de seguridad se refiere a la administración de cuentas de usuario. Esta solución implica establecer un número de días después de que una
      contraseña expire hasta que una cuenta se desactive de forma permanente debido a la inactividad del usuario.
```

La sugerencia es 30 días, es decir, si una contraseña está a punto de caducar, quedarán 30 días hasta que la cuenta se desactive automáticamente.

Ingrese el número de días para la inactividad de la cuenta (por ejemplo, 30)"

private: no

Vulnerabilidad 5 (0) Uninstall avahi Server Package

- name: desinstalar_avahi

prompt: |

"VULNERABILIDAD #5: Uninstall avahi Server Package

Este problema de seguridad se refiere a desinstalar del sistema los paquetes avahi-autoipd y avahi, relacionados con el servidor Avahi. Avahi es una

herramienta que implementa los protocolos de descubrimiento de servicios DNS Service Discovery (DNS-SD) y Multicast DNS (mDNS). Estos protocolos son

utilizados para el descubrimiento y la resolución de servicios en redes locales.

La recomendación implica desinstalar el servidor Avahi, ya que para la funcionalidad del sistema normalmente no se requiere de estos servicios.

Eliminar estos paquetes reduce la superficie de ataque.

Este playbook desinstalará los paquetes Avahi. ¿Desea remover Avahi del sistema? (Sí/No)"

private: no

Indicación para continuar con el proceso de ejecución y aplicar los cambios

- name: indicacion_aplicar_cambios

  prompt: |

"Se ha completado el ingreso de todos los datos necesarios para realizar las configuraciones de las 5 vulnerabilidades que serán solucionadas de forma dinámica.

Las 5 vulnerabilidades restantes serán resueltas mediante tareas estáticas, por lo que no se requiere la introducción de ningún dato adicional por parte del usuario."

Por favor, presione ENTER para continuar con el proceso de ejecución y aplicar los cambios de endurecimiento al sistema Alma Linux.

  private: no

##
##
##

##
##
##

 tasks:

##
##
##

TAREAS DINÁMICAS

```
##########################################################################
##########################################################################
##########################################
# Vulnerabilidad 1 (1) Set SSH authentication attempt limit
# Establecer límite de intentos de autenticación SSH
# Nivel de riesgo: Medio
##########################################################################
##################################################

  - name: Agregar o modificar la línea (MaxAuthTries) en el archivo sshd_config
    lineinfile:
      path: /etc/ssh/sshd_config
      regexp: '^.*MaxAuthTries.*$'
      line: 'MaxAuthTries {{ max_auth_tries }}'
    notify: Reiniciar servicio SSH

##########################################################################
################################################
# Vulnerabilidad 2 (1) Set Password Maximum Age
# Establecer un límite máximo de edad para las contraseñas de los usuarios del sistema
# Nivel de riesgo: Medio
##########################################################################
################################################

  - name: Agregar o modificar la línea (PASS_MAX_DAYS) en el archivo login.defs
    lineinfile:
      path: /etc/login.defs
      regexp: '^.*PASS_MAX_DAYS.*$'
```

```
      line: 'PASS_MAX_DAYS {{ pass_max_days }}'
    notify: Reiniciar el servicio de gestión de contraseñas SSSD

##########################################################################
##################################################

# Vulnerabilidad 3 (1) Set SSH MaxSessions limit

# Establecer un límite máximo de sesiones simultáneas permitidas para un usuario al conectarse

# al sistema a través de SSH

# Nivel de riesgo: Medio

##########################################################################
##################################################


  - name: Agregar o modificar la línea (MaxSessions) en el archivo sshd_config
    lineinfile:
      path: /etc/ssh/sshd_config
      regexp: '^.*MaxSessions.*$'
      line: 'MaxSessions {{ max_sessions }}'
    notify: Reiniciar servicio SSH


##########################################################################
##################################################

# Vulnerabilidad 4 (1) Set Account Expiration Following Inactivity

# Establecer el número de días después de que una contraseña expire por inactividad del usuario

# Nivel de riesgo: Medio

##########################################################################
##################################################
```

```
  - name: Agregar o modificar la línea (INACTIVE) en el archivo useradd
    lineinfile:
      path: /etc/default/useradd
      regexp: '^.*INACTIVE.*$'
      line: 'INACTIVE={{ inactive_days }}'

#########################################################################
#################################################

# Vulnerabilidad 5 (1) Uninstall avahi Server Package
# Desinstalar el paquete del servidor avahi
# Nivel de riesgo: Medio

#########################################################################
#################################################

  - name: Confirmación del usuario para desinstalar los paquetes Avahi
    pause:
      prompt: "Por favor, ingrese 'Si' o 'No' nuevamente para confirmar la desinstalación de los paquetes de Avahi."
    register: desinstalar_avahi

  - name: Detener la ejecución del playbook si la respuesta ingresada no es válida
    fail:
      msg: "Respuesta no válida. Por favor, ingrese 'Si' o 'No'."
    when: desinstalar_avahi.user_input | lower not in ['si', 'no']

  - name: Desinstalar paquetes Avahi si se ha confirmado
    package:
      name: "{{ item }}"
```

```
    state: absent
  loop:
    - avahi-autoipd
    - avahi
  when: desinstalar_avahi.user_input | lower == 'si'

##############################################################################
##############################################################################
##############################################

#                                    TAREAS ESTÁTICAS

##############################################################################
##############################################################################
##############################################

# Vulnerabilidad 6 (1) Disable Kernel Parameter for Accepting Secure ICMP Redirects on all IPv4 Interfaces

# Deshabilitar el parámetro del núcleo para aceptar redirecciones ICMP seguras en todas las interfaces IPv4

# Nivel de riesgo: Medio

# Descripción: Esta recomendación de seguridad implica deshabilitar la aceptación de redirecciones ICMP seguras en todos los interfaces IPv4 del kernel. ICMP es un protocolo

#          que se utiliza para el control y el manejo de mensajes en la capa de red del modelo OSI. Las redirecciones ICMP seguras son mensajes usados por los routers

#          para notificar a los hosts acerca de rutas más eficientes o preferidas. No obstante, hay situaciones en las que son objeto de uso indebido de manera maliciosa.

##############################################################################
##############################################################################
####################################
```

```
  - name: Desactivar aceptación de redirecciones ICMP seguras en tiempo de ejecución
    sysctl:
      name: "net.ipv4.conf.all.secure_redirects"
      value: "0"
    become: true

  - name: Asegurar que la configuración sea persistente cada vez que el sistema se inicie
    lineinfile:
      dest: /etc/sysctl.d/99-sysctl.conf
      line: "net.ipv4.conf.all.secure_redirects = 0"
    become: true

  - name: Recargar configuración de sysctl para aplicar los cambios.
    command: sysctl -p
    become: true

#########################################################################
#########################################################################
####################################

# Vulnerabilidad 7 (1) Disable XDMCP in GDM

# Deshabilitar XDMCP en GDM

# NIvel de riesgo: Alto

# Descripción: Esta recomendación de seguridad implica deshabilitar el XDMCP (X Display Manager Control Protocol) en el gestor de visualización GDM (GNOME Display Manager).

#          XDMCP es un protocolo que se usa para administrar sesiones gráficas remotas en entornos del sistema X Window System, sin embargo, la conexión no tiene cifrado
```

```
#           y autenticación por lo que representa un riesgo de seguridad.
#####################################################################
#####################################################################
#####################################

  - name: Verificar y eliminar la sección [xdmcp] si existe
    lineinfile:
      path: /etc/gdm/custom.conf
      state: absent
      regexp: '^\[xdmcp\]'

  - name: Modificar la configuración en el archivo custom.conf
    blockinfile:
      path: /etc/gdm/custom.conf
      block: |
        [xdmcp]
        Enable=false
    notify: Reiniciar el servicio de GDM

#####################################################################
#####################################################################
#####################################
# Vulnerabilidad 8 (1) Disable SSH Access via Empty Passwords
# Deshabilitar el acceso SSH para cuentas con contraseñas vacías
# NIvel de riesgo: Alto
# Descripción: Esta recomendación de seguridad implica deshabilitar el acceso SSH para cuentas de usuario configuradas con contraseñas vacías, representa un riesgo de
#           seguridad ya que no se necesita autenticación para acceder al sistema.
```

```
##########################################################################
##########################################################################
####################################

  - name: Crear archivo de configuración 00-complianceascode-hardening.conf
    file:
      path: /etc/ssh/sshd_config.d/00-complianceascode-hardening.conf
      state: touch
      mode: '0644'

  - name: Agregar la línea (PermitEmptyPasswords no) al archivo 00-complianceascode-hardening.conf
    lineinfile:
      path: /etc/ssh/sshd_config.d/00-complianceascode-hardening.conf
      line: "PermitEmptyPasswords no"
    notify: Reiniciar servicio SSH

##########################################################################
##########################################################################
####################################
# Vulnerabilidad 9 (1) Configure Firewalld to Trust Loopback Traffic
# Configurar Firewalld para confiar en el tráfico de bucle local (loopback)
# Nivel de riesgo: Medio
# Descripción: Esta medida de seguridad implica modificar la configuración del firewall para autorizar el tráfico en la interfaz de bucle local (loopback). Esto quiere decir
#              que, después de realizar este cambio en la configuración, el tráfico de bucle local podrá pasar a través del firewall sin restricciones, lo cual resulta crucial
#              para facilitar la comunicación interna en la máquina.
```

```
##########################################################################
##########################################################################
####################################

  - name: Asignar interfaz de loopback a la zona confiable de firewalld
    command: firewall-cmd --permanent --zone=trusted --add-interface=lo
    notify: Recargar las reglas de firewalld

##########################################################################
##########################################################################
####################################
# Vulnerabilidad 10 (1) Install AIDE
# Instalar AIDE
# Nivel de riesgo: Medio
# Descripción: Esta recomendación de seguridad implica instalar el paquete AIDE (Advanced Intrusion Detection Environment) en el sistema operativo Alma Linux. AIDE es una
#              herramienta de detección de intrusiones que lleva a cabo revisiones de integridad en los archivos del sistema, ofreciendo la capacidad de detectar
#          cambios no autorizados.
##########################################################################
##########################################################################
####################################

  - name: Instalar el paquete AIDE
    dnf:
      name: aide
      state: present
```

```
##############################################################################
##############################################################################
##########################################

##############################################################################
##############################################################################
##########################################

 handlers:

##############################################################################
##############################################################################
##########################################

# Vulnerabilidad 1 (2) Set SSH authentication attempt limit

# Vulnerabilidad 3 (2) Set SSH MaxSessions limit

# Vulnerabilidad 8 (2) Disable SSH Access via Empty Passwords

   - name: Reiniciar servicio SSH

     service:

       name: sshd

       state: restarted

# Vulnerabilidad 2 (2) Set Password Maximum Age

   - name: Reiniciar el servicio de gestión de contraseñas SSSD

     service:

       name: sssd

       state: restarted
```

```
# Vulnerabilidad 7 (2) Disable XDMCP in GDM


    - name: Reiniciar el servicio de GDM
      systemd:
        name: gdm
        state: restarted


# Vulnerabilidad 9 (2) Configure Firewalld to Trust Loopback Traffic


    - name: Recargar las reglas de firewalld
      command: firewall-cmd --reload
```