Universidad de Cuenca

Facultad de Ingeniería

Carrera de Ingeniería en Telecomunicaciones

Implementación de un Framework de pentesting para el laboratorio de Micro-Red de la Universidad de Cuenca

Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones

Autores:

Jaime Fabricio Malla Zhunio

Diego Mauricio Narvaez Paucar

Director:

Darwin Fabián Astudillo Salinas

ORCID: 0000-0001-7644-0270

Cuenca, Ecuador

2024-08-28

Resumen

Los *Industrial Control System* (ICS) son esenciales para gestionar infraestructuras críticas en sectores como energía, tratamiento de agua y manufactura. La creciente interconexión de estos sistemas con redes amplias, como Internet, los ha convertido en blancos atractivos para ciberataques, con potenciales repercusiones económicas y ambientales graves. Para mitigar estas amenazas, la Universidad de Cuenca, a través de su laboratorio de Micro-Red, desarrolló e implementó un *framework* de *pentesting White Box*, diseñado para identificar vulnerabilidades en los activos del laboratorio, que opera con fuentes de energía solar, eólica y térmica. La primera fase del proyecto incluyó la catalogación de activos y topología del laboratorio utilizando herramientas como *Gestionnaire Libre de Parc Informatique* (GLPI) y OpenCVE. A continuación, se adaptó el *framework* Daniz-Red para realizar *pentesting* en dispositivos de operación tecnológica, incorporando la gestión de protocolos como Modbus y *Simple Network Management Protocol* (SNMP), y mejorando las herramientas de escaneo de vulnerabilidades. Los resultados preliminares indicaron las correcciones necesarias para reforzar la seguridad sin interrumpir las operaciones. Además, se llevó a cabo un proceso de *hardening* en los servidores críticos del laboratorio, aplicando *scripts* de auditoría y remediación basados en las recomendaciones del *Center for Internet Security* (CIS). Este esfuerzo resultó en la optimización de la seguridad del laboratorio, abordando eficazmente todas las vulnerabilidades detectadas, mejorando la confidencialidad, integridad y disponibilidad de los sistemas, y fortaleciendo la infraestructura de seguridad del laboratorio según los principios fundamentales de la seguridad informática.

Palabras clave del autor: pruebas de penetración, corrección de vulnerabilidades, redes industriales, sistemas de control industrial

Abstract

Industrial Control System (ICS) are essential for managing critical infrastructure in sectors such as energy, water treatment, and manufacturing. The increasing interconnection of these systems with broader networks, such as the Internet, has made them attractive targets for cyberattacks, with potentially severe economic and environmental repercussions. To mitigate these threats, the University of Cuenca, through its Micro-Grid laboratory, developed and implemented a White Box pentesting framework, designed to identify vulnerabilities in the laboratory's assets, which operate with solar, wind, and thermal energy sources. The first phase of the project involved cataloging the laboratory's assets and topology using tools like *Gestionnaire Libre de Parc Informatique* (GLPI) and OpenCVE. Subsequently, the Daniz-Red framework was adapted to perform pentesting on operational technology devices, incorporating the management of protocols such as Modbus and *Simple Network Management Protocol* (SNMP), and enhancing vulnerability scanning tools. Preliminary results indicated the necessary corrections to strengthen security without disrupting operations. Additionally, a hardening process was carried out on the laboratory's critical servers, applying audit and remediation scripts based on *Center for Internet Security* (CIS) recommendations. This effort resulted in the optimization of the laboratory's security, effectively addressing all detected vulnerabilities, improving the confidentiality, integrity, and availability of the systems, and strengthening the laboratory's security infrastructure in line with fundamental principles of information security.

Author Keywords: penetration testing, vulnerability remediation, industrial networks, industrial control systems

Índice de contenido

Índice de figuras

Índice de tablas

Agradecimientos.

Después de un intenso periodo, hemos llegado a la culminación de una de las etapas más importantes de nuestras vidas. Es por ello que deseamos expresar nuestros más profundos agradecimientos a todas las personas que han sido parte de nuestro trabajo de titulación.

En primer lugar, deseamos expresar nuestra profunda gratitud a nuestro tutor, Ing. Fabián Astudillo, cuya invaluable orientación, experiencia, paciencia y dedicación han sido fundamentales para la realización de este trabajo. Sin su guía y apoyo incondicional, no hubiéramos culminado nuestro proyecto de manera exitosa.

En segundo lugar, queremos agradecer a todo el personal del Laboratorio de Micro-Red de la Universidad de Cuenca. Desde el primer día, nos abrieron las puertas y mostraron completa disposición para enseñarnos los aspectos técnicos operacionales del laboratorio, además de facilitarnos los recursos necesarios para ejecutar pruebas y desarrollar cada uno de los servicios requeridos.

En tercer lugar, agradecemos a nuestros compañeros, quienes a lo largo de nuestra trayectoria universitaria se han convertido en amigos. Gracias por todas las horas de estudio, risas, conversaciones y consejos, y por su incondicional acompañamiento en este trayecto. Sin ustedes, no hubiera sido posible llegar hasta este punto.

Finalmente, pero no menos importante, queremos agradecer a todos nuestros profesores, quienes han impartido sus conocimientos y, algunos de ellos, nos han ofrecido su amistad. Esto ha fomentado un cariño y una motivación especial para avanzar en el campo de las telecomunicaciones.

Para cada una de las personas mencionadas en este documento, ha sido un honor conocerlos y les deseamos éxitos en todos sus futuros proyectos.

Dedicatoria.

Dedico este trabajo a todas las personas que me acompañaron en esta trayectoria, tan desafiante como hermosa. En primer lugar, a mis padres, cuyo apoyo y cariño incondicional fueron mi fuente de inspiración y fortaleza. Sus esfuerzos y sacrificios me impulsaron a superar cada adversidad.

A mi hermana y su familia, quienes me brindaron su cariño y apoyo constante. A mis amigos de la universidad, con quienes forjé vínculos indelebles; gracias a su compañía, sus ocurrencias y los buenos momentos, este viaje ha sido inolvidable. Siempre tendrán un lugar especial en mi corazón.

También extiendo mi gratitud a mis amigos fuera de la universidad, quienes han sido pilares fundamentales en mi vida.

A todos ustedes, les debo un millón de gracias por hacer este sueño posible.

Fabricio

Este trabajo de titulación está, en primer lugar, dedicado a los dos pilares fundamentales de mi vida: mis padres, María Elena y Geovanny. Ellos, con su sacrificio y amor incondicional, han sido una motivación constante para llegar hasta este punto. Me han enseñado que, a pesar de cualquier dificultad, hay que seguir adelante y nunca rendirse. Me han brindado la mayor riqueza que una persona puede tener: la educación.

A mis abuelas, María y Rosario, dos mujeres trabajadoras que me han enseñado que con trabajo y dedicación no existen metas inalcanzables. Su cariño ha sido esencial a lo largo de mi vida. También a mis tíos y primos, quienes han aportado su granito de arena. Sin ninguna duda, su apoyo constante y su presencia en cada etapa de mi vida han sido fundamentales para llegar hasta aquí.

A ti, Fernanda, nuestros caminos se cruzaron hace varios años y desde ese momento has sido una de las personas que más ha creído en mí. En los momentos en los que parecía que estaba yendo a tocar fondo, has hecho todo lo posible por ayudarme; has sido mi luz, mi inspiración. Gracias por ser parte de mi vida y de mis sueños. Este logro también es tuyo.

Finalmente, pero no menos importante, quiero dedicar esto a las personas más cercanas que han estado a mi lado y han sido fundamentales. Gracias por sus risas, consejos y por hacer que el camino hasta alcanzar este logro sea mucho más alegre y llevadero.

Diego

Glosario

API *Application Programming Interface*. 31, 41, 42, 49, 51, 69, 83

CIS *Center for Internet Security*. 2, 3, 17, 19, 37, 65, 83

CISA Agencia de Seguridad de Infraestructura y Ciberseguridad. 26

CPS *Cyber Physical Systems*. 25

CSV *Comma-Separated Value*. 67

CVE *Common Vulnerabilities and Exposures*. 8, 19, 41, 51, 71, 83, 97, 102–105

CVSS *Common Vulnerability Scoring System*. 8, 38–40

DCS *Distributed Control Systems*. 24, 29

DMZ Zona Desmilitarizada. 28, 29

GLPI *Gestionnaire Libre de Parc Informatique*. 2, 3, 8, 10, 19, 41, 46–49, 66, 69–71, 83, 91–94, 102

HMI *Human–machine interface*. 23, 24

HTTP *Hypertext Transfer Protocol*. 51, 73, 77

HTTPS *Hypertext Transfer Protocol Secure*. 60, 77

ICS *Industrial Control System*. 2, 3, 5, 24–27, 29, 30, 44–46, 53, 57, 83, 84

IEC *International Electrotechnical Commission*. 29, 45

IEDs *Standard for Intelligent Electronic Devices*. 30

IEEE *Institute of Electrical and Electronics Engineers*. 30

Industria 4.0 *Fourth Industrial Revolution*. 24

IoT Internet Industrial de las Cosas. 16, 25, 31, 32

ISA *International Society of Automation*. 29

ISMS *information security management systemss*. 30

ISO *International Organization for Standardization*. 30, 45

LAN *Local Area Network*. 28

LCSA *Legal-Client-to-Server*. 45

MFA *multi-factor authentication*. 35

MITM Man-in-the-Middle. 45, 77

NIST *National Institute of Standards and Technology*. 29

NSE *Nmap Scripting Engine*. 42

NVT *Network Vulnerability Tests*. 43, 57, 83, 97

OPC OLE for Process Control. 17

OSINT *Open Source Intelligence*. 42

PLC *Programmable Logic Controller*. 23, 24, 57, 72

PTES *Penetration Testing Execution Standard*. 8, 33, 34, 53

RBAC *using role-based access control*. 36

RTU Remote Terminal Unit. 23, 57

SCADA *Supervisory Control and Data Acquisition*. 4, 8, 16–19, 22, 23, 29, 45, 95, 99

SNMP *Simple Network Management Protocol*. 2, 3, 47, 55, 64, 73

TI Tecnologías de la Información. 19, 21, 24, 29, 34, 41, 45, 71, 83

TO Tecnologías de Operación. 17, 19, 21, 24, 26, 29, 32, 46, 70, 71, 75, 83

VPN *Virtual private network*. 28, 75

1. Generalidades

1.1. Introducción

Una Micro-Red es un sistema energético diseñado para generar electricidad de manera autónoma y eficiente, con elementos de comunicaciones y medición inteligente para control y monitoreo automáticos [1]. Actualmente, estas redes son objetivo de numerosos ataques cibernéticos debido a su importancia en la sostenibilidad y funcionamiento de sistemas energéticos. En la Universidad de Cuenca, la Micro-Red presenta un nivel de madurez en ciberseguridad nulo o casi inexistente, según una evaluación previa [2]. Una solución para mejorar la seguridad es implementar pruebas de penetración (pentesting) y prácticas de endurecimiento (hardening) en todos los activos de la red. El pentesting permite evaluar la robustez de la infraestructura tecnológica al exponer vulnerabilidades y desafiar la resiliencia de los sistemas [3, 4]. Estas prácticas, aunque asociadas con grandes organizaciones, también son valiosas para sistemas más pequeños, donde la seguridad no puede pasarse por alto. Es importante recordar que el pentesting no es una solución única, sino parte de un enfoque integral de seguridad [3].

El control y monitoreo en la Micro-Red actualmente se realiza a través de un sistema *Supervisory Control and Data Acquisition* (SCADA); por lo que, es necesario enfocarse en la protección de este recurso. Implementar endurecimiento (*Hardening*) en este servidor y en los activos de la red implica fortalecer las defensas mediante medidas como la configuración de servicios y el cierre de puertos innecesarios, la gestión y control de acceso, así como actualizaciones de software y hardware [4]. Estas prácticas se vuelven críticas en un mundo cada vez más interconectado.

1.2. Antecedentes y alcances del proyecto

Los sistemas y dispositivos dentro de una Micro-Red, al estar interconectados en una red, se vuelven susceptibles a diversas amenazas, tanto internas como externas. Además, con la creciente adopción del Internet Industrial de las Cosas (IoT) y la expansión de la conectividad, los sistemas y dispositivos se han vuelto cada vez más vulnerables a ataques externos [5].

Ante la constante necesidad de estar conectados a Internet los sistemas y dispositivos pertenecientes a la Micro-Red estos se han expuesto a ciberataques y vulnerabilidades, lo que subraya la necesidad de experiencia especializada en ciberseguridad [6]. Es esencial proteger activos críticos en los sistemas como en el caso del sistema SCADA de la Micro-Red y

garantizar que estén a salvo de amenazas internas y externas. En este contexto, el *pentesting* desempeña un papel crucial al identificar vulnerabilidades potenciales y amenazas que podrían explotarse [7].

Estos estudios muestran que el *pentesting* es esencial para evaluar y fortalecer la seguridad de una Micro-Red. Dada la naturaleza crítica de estos sistemas y dispositivos y su exposición constante a riesgos cibernéticos, se plantea la necesidad de implementar un enfoque de seguridad proactivo. En este contexto, la creación de un marco de trabajo (*framework*) personalizado para *pentesting* en la Micro-Red se presenta como una solución integral que permitirá evaluar y mejorar la seguridad de los sistemas y dispositivos, agregando posibles mejoras al *Framework* implementado en un trabajo previo [8] y una etapa de filtrado enfocada en dispositivos de las Tecnologías de Operación (TO).

Una vez que se hayan obtenido los informes de vulnerabilidades, resulta esencial llevar a cabo una fase de *hardening* de la seguridad, como se detalla en el trabajo [2]. Este trabajo proporciona una visión actual de la topología de la red en el laboratorio de Micro-Red, donde se encuentra una máquina virtual ejecutando el sistema operativo Windows Server 2012 R2 y operando el sistema SCADA con el apoyo del Servidor OLE for Process Control (OPC). Otro servidor que debemos considerar es el de seguridad que ofrece el servicio GLPI y Guacamole, el cual se implementó con la finalidad de tener un inventario de los activos del laboratorio y una conexión remota al SCADA. Por lo tanto, debemos implementar medidas que garanticen la protección de los datos y disponibilidad del servicio.

Actualmente, el *Center for Internet Security* (CIS) es una organización sin fines de lucro dedicada a la creación de estándares de seguridad, como los CIS *Controls* y CIS *Benchmarks* [9], que proporcionan pautas detalladas para fortalecer la protección de sistemas y redes contra las amenazas cibernéticas [10]. Con el objetivo de mejorar la seguridad y reducir la exposición a posibles ataques, es posible utilizar el lenguaje de scripting PowerShell para administrar registros del sistema operativo y llevar a cabo diversas tareas como la administración de contraseñas, control de acceso, cierre de puertos, entre otros.

1.3. Metodología

El presente trabajo de titulación está enfocado en la implementación de un *framework* que sea capaz de detectar y solventar vulnerabilidades basándose en *pentesting* y *hardening* de seguridad hacia los activos y dispositivos que pertenece a la Micro-Red, es por eso que es

fundamental seccionar el trabajo en tres fases importantes.

Figura 1.1: Esquema de implementación de *Framework* de *pentesting* y *hardening* en el sistema SCADA.

El diagrama de la Figura 1.1 ilustra las tres etapas fundamentales: Gestión de Activos, Implementación del *Pentesting* y Corrección de Vulnerabilidades con *Hardening*. Estas etapas están respaldadas por subetapas complementarias que facilitarán la implementación de los *frameworks*. A continuación, se presenta una introducción de cada una de estas fases, una explicación más detallada se realizará en el Capítulo 3.

1.3.1. Fase 1 (Gestión de activos)

El primer paso para llevar a cabo el *pentesting* en un sistema SCADA consiste en comprender el estado de la red. Existen tres tipos diferentes de pentesting: *Black Box*, *Gray Box* y *White Box*.

El enfoque de este trabajo es *White Box*, dado que se necesita tener conocimiento y acceso total de la red para identificar los activos Tecnologías de la Información (TI) y TO del laboratorio. El objetivo en esta fase es administrar una herramienta de código abierto, como *Gestionnaire Libre de Parc Informatique* (GLPI), que nos ayudará en la gestión de activos y la identificación de vulnerabilidades conocidas a través de la extracción de informes *Common Vulnerabilities and Exposures* (CVE).

1.3.2. Fase 2 (Implementación de *pentesting*)

Como siguiente paso, se procederá con el desarrollo del *framework* de *pentesting*, una estructura que abarca múltiples fases. En un principio, se llevará a cabo un análisis exhaustivo del *framework* actualmente en uso, evaluando las herramientas y técnicas que lo componen. Se considera la posibilidad de actualizar las herramientas y módulos existentes y se contempla la integración de nuevas herramientas para mejorar la capacidad de detección de vulnerabilidades. Posteriormente, se lleva a cabo el *pentesting* utilizando el *framework* actualizado, generando un informe detallado que documenta las vulnerabilidades identificadas. Luego, se procede a una fase de filtrado, aprovechando la gestión de activos previamente realizada, con el propósito de identificar y seleccionar los activos directamente vinculados al sistema SCADA, marcando el camino para el siguiente paso.

1.3.3. Fase 3 (Corrección de vulnerabilidades con *hardening*)

Una vez se obtenga el informe y se realice la etapa de filtrado de la fase de *pentesting* y gestión de activos, se evaluarán las vulnerabilidades críticas para identificar aquellas cuya mitigación no comprometerá la integridad de la Micro-Red. Para la mitigación de las vulnerabilidades, se seguirán las prácticas recomendadas por los *benchmarks* del CIS. Estas prácticas se materializarán en los scripts de *hardening* que se aplicarán sobre los dispositivos, asegurando un enfoque sistemático y validado en la mitigación de riesgos.

Una vez completado este proceso, se procederá a la elaboración de un manual con recomendaciones destinado a nuevos usuarios que deseen conectarse a la red. Este enfoque se adopta para prevenir problemas de seguridad relacionados con la ingeniería social que puedan afectar el funcionamiento de la Micro-Red en su totalidad.

1.4. Impactos y beneficios

La implementación de un *framework* centrado en el *pentesting* conlleva varios impactos y beneficios significativos para la gestión de activos y la seguridad de una Micro-Red:

1. **Mejora en la gestión de activos y vulnerabilidades conocidas:** El *framework* permite una gestión más efectiva de los activos al identificar y abordar las vulnerabilidades conocidas. Esto previene posibles ataques derivados de configuraciones incorrectas, falta de actualizaciones de firmware y otros problemas.
2. **Revelación de riesgos potenciales:** A través del *pentesting*, se identifican los puertos, servicios y otros elementos que podrían representar un riesgo para la correcta operación y disponibilidad del sistema. Estos hallazgos se documentan en un informe detallado.
3. **Aumento del nivel de seguridad:** En un entorno donde los sistemas operativos antiguos ya no reciben soporte por parte de sus desarrolladores, el *pentesting* permite elevar la seguridad o mitigar posibles amenazas. Esto es esencial para prevenir ataques comunes, especialmente aquellos relacionados con configuraciones erróneas o mal uso por parte de los administradores.

El propósito general de este enfoque es elevar el nivel de seguridad de la Micro-Red, que en la actualidad se encuentra en estado de madurez en ciberseguridad inexistente. Esto es especialmente relevante en el contexto del laboratorio de Micro-Red de la Universidad de Cuenca, donde se busca prevenir amenazas como la suplantación de identidad, ataques de denegación de servicio y otros riesgos potenciales. La implementación del *framework* proporciona un enfoque proactivo para abordar estos desafíos y garantizar la integridad y disponibilidad de los sistemas críticos.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar, implementar y evaluar un *framework* de *pentesting* para el laboratorio de Micro-Red de la Universidad de Cuenca, con el propósito de identificar vulnerabilidades, corregirlas mediante *hardening* y suministrar una herramienta para la detección y mitigación de amenazas cibernéticas.

1.5.2. Objetivos específicos

- Realizar un escaneo exhaustivo de la red en el laboratorio para identificar y catalogar los activos TI y TO (por ejemplo: controladores industriales, ordenadores, servidores con su respectivo sistema operativo, etc), para posteriormente generar un reporte que muestre las vulnerabilidades encontradas en los dispositivos.
- Gestionar y mejorar un *framework* de *pentesting* para evaluar la seguridad de los activos pertenecientes al laboratorio de Micro-Red de la Universidad de Cuenca, con el fin de identificar vulnerabilidades y generar informes detallados al respecto.
- Desarrollar *scripts* de *hardening* para mitigar vulnerabilidades en los activos de la Micro-Red, y validar su eficacia mediante pruebas exhaustivas en la mejora de la seguridad de dichos activos.
- Evaluar el impacto de la implementación del *framework* en términos de seguridad y proponer recomendaciones finales basadas en los resultados obtenidos y posibles mejoras o expansiones del *framework*.

2. Marco teórico

En este capítulo se abordarán todos los conceptos teóricos considerados de importancia, los cuales permitieron desarrollar el presente trabajo de titulación.

2.1. Micro-Redes y SCADA

2.1.1. Introducción a las Micro-Redes

Una Micro-Red eléctrica se concibe como un sistema de generación eléctrica bidireccional, que posibilita la distribución de electricidad desde los proveedores hasta los consumidores mediante una red de energía a nivel de distribución, limitada en extensión con una capacidad menor a 100 kW [11]. Estos sistemas son totalmente autónomos y son considerados como cargas para la red principal, pudiendo operar tanto en conexión con la red más amplia como de manera aislada. La integración de tecnología digital en las Micro-Redes favorece la incorporación de fuentes de generación de origen renovable, con el fin último de ahorrar energía, reducir costos e incrementar la fiabilidad [12].

2.1.2. Elementos que constituyen una Micro-Red

Los elementos que constituyen una Micro-Red inteligente incluyen: sistemas de generación distribuida; sistemas de almacenamiento de energía; técnicas para la gestión de cargas; sistemas de monitorización y control del flujo de potencia; y técnicas y procedimientos de mantenimiento preventivo. Estos elementos trabajan en conjunto para ofrecer una solución energética más eficiente y confiable, representando una evolución significativa en la gestión y distribución de energía eléctrica [12]. Las Micro-Redes, por tanto, ofrecen grandes oportunidades para integrar a pequeña escala las fuentes de energía renovable a los sistemas de potencia local, adaptándose a las necesidades específicas de las comunidades que sirven y contribuyendo a un futuro energético más sostenible [11].

2.1.3. Sistema *Supervisory Control and Data Acquisition* (SCADA) en Micro-Redes

Un sistema SCADA en una micro-red desempeña varias funciones. Como las de monitorear y controlar la generación y distribución de distintas fuentes de energía, además de gestionar dispositivos de almacenamiento y servicios auxiliares. Este sistema recopila datos en tiempo real

de diversos dispositivos y sensores distribuidos en una Micro-Red, procesa estos datos para proporcionar información útil a los operadores y ofrece una interfaz visual para supervisar el estado de la Micro-Red, permitiendo realizar ajustes necesarios. Las principales operaciones de control son ejecutadas automáticamente por Remote Terminal Unit (RTU) y/o *Programmable Logic Controller* (PLC), pero un operador experto puede intervenir manualmente si es necesario. Los datos se recopilan y procesan en tiempo real y se ponen a disposición de los operadores a través de una *Human–machine interface* (HMI), permitiendo la supervisión y ajustes necesarios [13].

2.1.4. Estructura de un sistema SCADA

La estructura básica de un sistema SCADA en una Micro-Red se compone de varios componentes clave que trabajan juntos para garantizar el control y la supervisión eficientes de la red, tal y como se observa en la Figura 2.1. La estación de programación de SCADA se encuentra en un ordenador, donde se desarrollan y gestionan las configuraciones del sistema [14].

Este ordenador se comunica con la HMI, que permite a los operadores monitorear y controlar los procesos de la Micro-Red a través de una interfaz gráfica intuitiva. La HMI se conecta a los PLC, los cuales reciben y procesan datos desde el equipo en campo, como generadores y robots industriales, y ejecutan las instrucciones de control necesarias [14].

Los PLC envían datos operativos a la HMI y reciben comandos de control, facilitando una comunicación bidireccional. Esto permite la supervisión en tiempo real y la intervención manual por parte de los operadores si es necesario, mejorando la eficiencia y seguridad del sistema. En conjunto, estos componentes aseguran un control efectivo y automatizado de los diversos elementos de la Micro-Red [14].

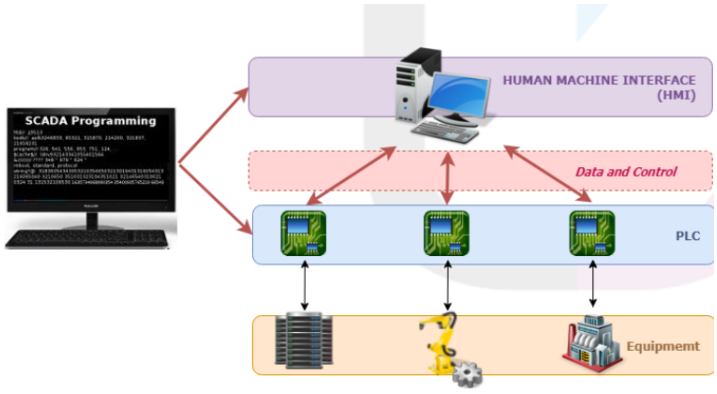

Figura 2.1: Estructura básica de un sistema SCADA [14].

2.2. ICS y Tecnologías Operacionales

2.2.1. ¿Qué es un ICS?

Un *Industrial Control System* (ICS) es un conjunto de dispositivos, redes y sistemas que gestionan, comandan y regulan el funcionamiento de los equipos industriales y procesos operativos. Estos sistemas son importantes en varios sectores industriales debido a su capacidad para monitorear y controlar operaciones en tiempo real, asegurando así la eficiencia, la seguridad y la fiabilidad de las plantas industriales y las infraestructuras críticas [15]. Los ICS incluyen distintos tipos de sistemas como los sistemas glsSCADA, *Distributed Control Systems* (DCS), y otros componentes como PLC y HMI. Estos componentes trabajan de manera integrada para facilitar la automatización y el control continuo de los procesos industriales [16].

2.2.2. Tecnologías operacionales

Las Tecnologías de Operación (TO) comprenden el conjunto de hardware y software dedicado a la monitorización y control de dispositivos físicos. En el contexto de los ICS, las TO se refiere a las tecnologías que se aplica directamente en la gestión y operación de componentes industriales esenciales, como los encontrados en los sectores de manufactura, energía y servicios públicos. Las TO son pilares en la gestión de infraestructuras críticas y operaciones industriales, proporcionando la interfaz y el control necesarios para asegurar la continuidad y eficiencia de los sistemas físicos [16].

2.2.3. Integración de ICS y TO con TI

La convergencia de las Tecnologías de la Información (TI) y las TO ha sido una tendencia creciente, especialmente con el auge de la *Fourth Industrial Revolution* (Industria 4.0), que busca integrar sistemas de TI más avanzados con plataformas operativas tradicionales para mejorar la eficiencia operativa y la toma de decisiones basada en datos. Esta integración ha sido beneficiosa en términos de gestión remota y recopilación de datos avanzada, pero también ha presentado nuevos desafíos de seguridad. La exposición de sistemas de las TO a las redes de las TI ha aumentado el riesgo de ataques cibernéticos, ya que muchos sistemas de las TO no fueron diseñados inicialmente con defensas cibernéticas robustas en mente. Esto hace que los ICS sean susceptibles a un rango de amenazas cibernéticas, desde *malware* hasta intrusiones sofisticadas diseñadas para controlar o dañar equipos y procesos [15].

2.3. Ciberseguridad en redes industriales

En la era de la transformación digital, la ciberseguridad ha dejado de ser una preocupación exclusiva de usuarios de computadoras e Internet para convertirse en un asunto crítico que abarca todos los sectores de la sociedad. Esta expansión del impacto de la ciberseguridad se debe a la creciente dependencia de redes y sistemas computacionales que interactúan directamente con el mundo real a través de sensores y actuadores. Estos sistemas, conocidos colectivamente como *Cyber Physical Systems* (CPS), Internet Industrial de las Cosas (IoT), Internet Industrial, entre otros, representan una fusión de capacidad digital y operativa que es fundamental para operaciones críticas en diversos sectores [17].

En particular, los ICS se han identificado como áreas de interés prioritarias dentro de la ciberseguridad debido a su papel esencial en la gestión y operación de infraestructuras críticas en sectores como energía, agua y manufactura. Un ataque cibernético que logre perturbar estos sistemas puede tener consecuencias devastadoras para la salud, seguridad y el medio ambiente, resaltando la relevancia de estos sistemas como blancos principales para actividades maliciosas [18].

2.3.1. Vulnerabilidades en ICS

Los sistemas ICS compuestos por componentes de *hardware* y *software*, enfrentan amenazas de seguridad significativas que pueden comprometer su integridad y confiabilidad. A continuación, se listan las principales categorías de vulnerabilidades asociadas con los ICS [19]:

- **Vulnerabilidades de arquitectura y diseño:** La seguridad de los ICS puede verse comprometida desde la fase de diseño, donde la falta de incorporación adecuada de medidas de seguridad puede dejar a estos sistemas abiertos a explotaciones futuras. Los diseños arquitectónicos que no delimitan claramente los perfiles de seguridad crean brechas que pueden ser explotadas por actores maliciosos. La ausencia de un perímetro de seguridad definido y una inadecuada recolección de datos de eventos históricos complican la detección y respuesta ante incidentes, dejando a los sistemas susceptibles a ataques prolongados sin detección.
- **Vulnerabilidades de configuración y mantenimiento:** Las configuraciones inadecuadas y la falta de mantenimiento regular son fuentes significativas de vulnerabilidades en ICS. Los sistemas que operan con *software* o sistemas operativos desactualizados pue-

den incluir vulnerabilidades sin parchear que facilitan los ataques. Además, la instalación de *malware* a través de ingeniería social o brechas de seguridad puede llevar a un control no autorizado de los sistemas. La gestión deficiente del acceso remoto, junto con procedimientos inadecuados de respaldo y recuperación, aumenta el riesgo de incidentes de seguridad que pueden causar daños irreversibles.

- **Vulnerabilidades de desarrollo de software:** Errores en el desarrollo de software, como la validación inadecuada de datos, pueden introducir vulnerabilidades explotables en los ICS. Estas vulnerabilidades permiten a los atacantes realizar ataques de inyección de código, desbordamiento de *buffer* y otros *exploits* que comprometen la seguridad del sistema.
- **Vulnerabilidades de comunicación y configuración de red:** La configuración de red insegura y el uso de protocolos de comunicación no cifrados permiten a los atacantes interceptar o manipular datos transmitidos. La falta de *firewalls* adecuados, registros detallados y medidas de autenticación robusta pueden facilitar el acceso no autorizado y la manipulación de los sistemas de ICS, así dejando una brecha a la provocación de ataques o filtración de datos.
- **Vulnerabilidades físicas** El acceso físico no autorizado a las instalaciones de ICS puede permitir a los atacantes manipular o dañar directamente equipos críticos. Esta categoría de vulnerabilidad destaca la importancia de la seguridad física integrada en la estrategia general de protección de infraestructuras críticas.

2.3.2. Amenazas en ICS

A partir de las vulnerabilidades, los ICS enfrentan amenazas cibernéticas que pueden explotar dichas debilidades. Estas amenazas han evolucionado en complejidad y frecuencia, convirtiéndose en un foco significativo para la seguridad nacional e internacional. La Agencia de Seguridad de Infraestructura y Ciberseguridad (CISA) ha delineado prioridades clave que enfatizan la necesidad de proteger activamente estos sistemas [20].

2.3.2.1. Incidentes de ataques a ICS

En 2021, el panorama de amenazas para los ICS se intensificó notablemente, con un incremento del 50 % en las vulnerabilidades identificadas en dispositivos de TO y sus sistemas de

gestión. Los grupos de *ransomware*, en particular, han continuado con su enfoque persistente hacia la fabricación y la infraestructura crítica, intensificando la frecuencia y severidad de los ataques. Estos ataques no solo exponen debilidades sistémicas sino que también pueden causar daños irreparables, como se evidencia en varios incidentes significativos [20]:

- **Ataque de *ransomware* a Colonial Pipeline (2021):** Este ataque provocó una escasez severa de combustible y un aumento en los precios. El incidente fue causado por el aprovechamiento de una cuenta VPN inactiva, lo que obligó a la empresa a pagar millones para recuperar el control de sus sistemas.
- **CPC Corp. Taiwan—Ataque de *ransomware* (2020):** Un ataque que se realizó mediante una unidad USB para violar la red de la empresa, que en sí no afectó directamente a la producción, pero si impacto sobre otras operaciones críticas, como lo fue en el sistema de pago.
- **Ataque al sistema de energía de Ucrania (2015):** Representó uno de los primeros incidentes conocidos de *malware* utilizado para causar una interrupción a gran escala del suministro de energía, afectando a millones de residentes y resaltando la amenaza de los ciberataques a la infraestructura nacional.

2.3.3. Políticas de seguridad

Las políticas de seguridad de la red son esenciales para proteger la integridad y la operatividad de los ICS. Estas políticas no solo abordan la protección de *hardware* y *software*, sino también la gestión de conductas humanas y respuestas automatizadas a amenazas potenciales. A continuación, se detallan las principales políticas [21]:

- **Política de seguridad de dispositivos**
  - Implementación de controles de seguridad en dispositivos como *routers* y *switches* para gestionar el flujo de tráfico.
  - Firma obligatoria de un acuerdo de no divulgación por parte de los empleados sobre los detalles de los dispositivos de seguridad internos.
  - Aplicación constante de parches de seguridad y actualizaciones críticas.
  - Desactivación de servicios innecesarios para minimizar vulnerabilidades.
- **Política de acceso a Internet**

- Restricción automática de acceso a sitios web considerados inapropiados.
- Adaptación del acceso a Internet según las funciones laborales específicas del empleado.

- **Política de *Virtual private network* (VPN)**
  - Restricción del uso de VPN al sistema informático de la organización y a empleados aprobados.
  - Aseguramiento de que el acceso remoto a la red corporativa se realice a través de conexiones VPN seguras y autorizadas.
- **Política de comunicación de puertos**
  - Bloqueo de puertos no esenciales para prevenir accesos y servicios no autorizados.
- **Política de *Local Area Network* (LAN) inalámbrica**
  - Implementación de medidas de autenticación robustas y reemplazo de protocolos de seguridad obsoletos.
- **Política de conexión remota**
  - Restricciones severas sobre el acceso remoto directo a servidores críticos, permitido sólo bajo protocolos de cifrado robustos.
- **Política de reglas de *firewall***
  - Configuración de *firewalls* para filtrar y gestionar el tráfico de acuerdo a las necesidades de la red.
- **Política de detección de intrusiones**
  - Empleo de sistemas de detección de intrusiones para vigilancia y notificación de actividades sospechosas.
- **Política de servidores *proxy***
  - Regulación del tráfico de la red mediante servidores *proxy* correctamente configurados y actualizados.
- **Política de comunicación segura**
  - Uso de cifrado para proteger la transmisión de datos sensibles a través de la red.
- **Política de Zona Desmilitarizada (DMZ)**

- Uso de una DMZ para aislar y proteger los sistemas que requieren acceso público.

2.3.4. Estándares y normativas

Varias organizaciones nacionales e internacionales desarrollan estándares y normativas que guían la implementación y gestión de medidas de seguridad en entornos industriales. Estas entidades abarcan un amplio espectro de industrias y tecnologías, estableciendo *frameworks* que ayudan a proteger la infraestructura crítica frente a amenazas emergentes y continuas. Entre estas organizaciones, algunas de las más influyentes incluyen [22–25]:

- ***International Society of Automation* (ISA):** Desarrolla normas y guías para la ciberseguridad en sistemas de control y automatización industrial.
  - **ISA/IEC 62443:** Establece requisitos y procesos para implementar y mantener ICS electrónicamente seguros. Estas normas definen las mejores prácticas de seguridad y ofrecen un método para evaluar el nivel de desempeño de la seguridad. Adoptan un enfoque holístico para los desafíos de la ciberseguridad, cerrando la brecha entre las TO y TI, así como entre la seguridad de procesos y la ciberseguridad.
- ***International Electrotechnical Commission* (IEC):** Proporciona estándares internacionales para la ciberseguridad de sistemas de automatización y control industrial.
  - **IEC 61508:** Es el estándar principal de seguridad funcional y es aplicable a todas las industrias. Cuando no existen estándares de seguridad funcional específicos para productos en una industria, se puede aplicar IEC 61508. Este estándar proporciona normas de seguridad funcional para todo el ciclo de vida de sistemas y productos eléctricos, electrónicos o programables. Se enfoca en aquellas partes de un dispositivo o sistema que realizan funciones de seguridad automatizadas, como sensores, lógica de control, actuadores y microprocesadores.
- ***National Institute of Standards and Technology* (NIST):** Publica guías y estándares para mejorar la ciberseguridad de infraestructuras críticas, incluyendo ICS.
  - **NIST SP 800-82:** Proporciona lineamientos para mejorar la seguridad de los ICS, que incluyen sistemas SCADA, DCS, y otros tipos de sistemas de control utilizados en operaciones industriales, de infraestructura y de fabricación. Este estándar ofrece una guía detallada sobre cómo proteger los ICS frente a amenazas y vulnerabilidades mientras se mantiene su disponibilidad, integridad y confidencialidad.

- ***International Organization for Standardization* (ISO):** Desarrolla estándares para la gestión de la seguridad de la información, aplicables a ICS.
    - **ISO/gIsIEC 27001:** es el estándar más reconocido mundialmente para *information security management systemss* (ISMS). Establece los requisitos que debe cumplir un ISMS y ofrece orientación a empresas de cualquier tamaño y de todos los sectores de actividad para establecer, implementar, mantener y mejorar continuamente un sistema de gestión de seguridad de la información.
- ***Institute of Electrical and Electronics Engineers* (IEEE):** Desarrolla estándares técnicos para la ciberseguridad en redes industriales y sistemas de control.
    - **IEEE 1686:** especifica los requisitos de seguridad para los *Standard for Intelligent Electronic Devices* (IEDs) utilizados en los sistemas de control y automatización de subestaciones eléctricas. Su principal objetivo es proporcionar una guía sobre las funciones de seguridad que deben ser implementadas en los IEDs para proteger los sistemas eléctricos de posibles ciberataques. El IEEE 1686 cubre aspectos como la gestión segura de acceso, la auditoría de eventos de seguridad, la gestión de la configuración de seguridad y la respuesta a incidentes de seguridad, entre otros.

2.4. Principios de *pentesting*

El *pentesting*, es una técnica fundamental de seguridad cibernética que involucra la evaluación de la seguridad de sistemas de información a través de la simulación de ataques por parte de actores malintencionados. Esta estrategia busca identificar y explotar vulnerabilidades en un ambiente controlado para comprender mejor los riesgos potenciales antes de que sean explotados de manera maliciosa. El alcance del *pentesting* abarca diversas pruebas, como exámenes externos, pruebas internas, evaluaciones de aplicaciones web y pruebas de ingeniería social, cada una diseñada para detectar vulnerabilidades específicas que podrían ser utilizadas para comprometer la integridad, disponibilidad y confidencialidad de los datos y sistemas [26].

El objetivo primordial de un *pentest* es doble: el primero, es identificar vectores de ataque, vulnerabilidades y problemas de configuración o control que podrían poner en riesgo los sistemas de información. Mientras que el segundo, está diseñado para fortalecer la seguridad del sistema objetivo al proporcionar correcciones y recomendaciones específicas para cerrar brechas de seguridad detectadas durante la prueba [27].

Los *pentests* son llevados a cabo por expertos en seguridad ofensiva, conocidos como "*pentesters*", que utilizan su habilidad, experiencia e imaginación para descubrir debilidades, tanto técnicas como humanas. Esta evaluación puede abarcar no solo las infraestructuras y redes, sino también aplicaciones móviles y web, y otros dispositivos conectados [26]. Además un *pentest* proporciona un informe detallado que incluye todos los hallazgos, una lista de riesgos de seguridad identificados, y orientación detallada sobre cómo mitigar los riesgos más críticos para la organización [28].

2.4.1. Tipos de *pentesting*

Existen varios tipos de *pentesting*, cada uno diseñado para evaluar diferentes sistemas y componentes dentro de una infraestructura. Estas evaluaciones involucran un ataque simulado contra los sistemas informáticos de una empresa, enfocándose en distintos activos empresariales según el tipo de evaluación. A continuación se presentan los posibles tipos de *pentesting* que se pueden realizar [29] [26]:

- ***Pentesting* de aplicaciones:** Evalúan vulnerabilidades en aplicaciones y sistemas incluyendo aplicaciones web, móviles, de IoT, en la nube y *Application Programming Interface* (API). Estas pruebas no solo detectan problemas comunes como inyecciones de código y fallos de autenticación, sino también deficiencias menos comunes y específicas de cada aplicación, abarcando desde fallos criptográficos hasta problemas de configuración de seguridad. Además, el *pentesting* de aplicaciones web revisa las vulnerabilidades asociadas a la configuración de las infraestructuras que alojan los servicios, como servidores y entornos en la nube, incluyendo *software* desactualizado y errores de configuración.
- ***Pentesting* de red:** Evalúan la seguridad de toda la red informática de una empresa, divididas en pruebas externas e internas. Las pruebas externas simulan el comportamiento de *hackers* externos para identificar problemas de seguridad en activos expuestos a Internet, como servidores, *routers*, sitios web y computadoras de empleados. Estas pruebas se realizan desde fuera de la red. Por otro lado, las pruebas internas imitan el comportamiento de personas malintencionadas dentro de la empresa, como empleados con intenciones nocivas o *hackers* con credenciales robadas, con el objetivo de descubrir vulnerabilidades que podrían ser explotadas desde dentro, como el abuso de privilegios de acceso para robar datos sensibles. .
- ***Pentesting* de *hardware*:** Evalúan la seguridad de dispositivos conectados a la red, co-

mo *laptops*, dispositivos móviles, IoT y TO. Estas pruebas buscan fallos de *software* que podrían permitir a los *hackers* accesos remotos, así como vulnerabilidades físicas, como centros de datos inadecuadamente asegurados que podrían ser infiltrados por actores maliciosos.

- ***Pentesting* del personal:** Evalúan las debilidades en la higiene cibernética de los empleados, enfocándose especialmente en cómo una empresa puede ser vulnerable a ataques de ingeniería social. Estos *pentests* incluyen tácticas como *phishing*, *vishing* (*phishing* por voz) y *smishing* (*phishing* por SMS) para engañar a los empleados y hacer que divulguen información sensible. Además, estas pruebas pueden incluir la evaluación de la seguridad física de las oficinas, por ejemplo, intentando infiltrarse en un edificio disfrazados de repartidores, técnica conocida como "*tailgating*", comúnmente utilizada por criminales en el mundo real.

2.4.2. Técnicas de *Pentesting*

Las técnicas de *pentesting* se pueden clasificar en 3 tipos: caja negra, caja gris y caja blanca, a continuación se describen cada una de ellas [28] [29]:

- ***Pentesting* de caja negra:** Simula un ataque cibernético real, proporcionando al *pentesters* la mínima información sobre las credenciales y recursos de la organización. Este tipo de *test* es ideal para organizaciones con estrategias de seguridad maduras y controles establecidos, dado que el *pentesters*, al no tener información previa, debe realizar un reconocimiento exhaustivo para determinar posibles caminos de ataque, lo cual puede prolongar la duración de la prueba comparado con otros tipos de *pentests*.
- ***Pentesting* de caja blanca:** Proporcionan al *pentester* la mayor cantidad de información disponible antes de realizar las pruebas. Los *pentesters* reciben acceso a documentos internos, credenciales de usuarios y administradores, código fuente, y datos completos sobre la red y sistemas de la organización. Esta amplitud de información permite a los *hackers* éticos enfocarse en explotar objetivos específicos sin la necesidad de utilizar métodos de ataque externos para ganar entrada inicial, configurando un escenario de brecha asumida.
- ***Pentesting* de caja gris:** Combina elementos de las pruebas de caja negra y caja blanca, proporcionando al *pentester* más información que en una caja negra pero mucho menos que en una caja blanca. Típicamente, se proveen algunos detalles como credenciales

de inicio de sesión, mientras que otra información privilegiada se retiene. Este enfoque es útil para evaluar cómo un ataque podría desarrollarse si un cibercriminal consiguiera información de acceso pero sin conocer la estructura detallada de la red y los sistemas de la organización.

2.4.3. *Red team*

Un *red team* representa una forma especializada de *pentesting* que difiere significativamente de un *pentest* tradicional en términos de objetivos y métodos. Mientras que un *pentest* convencional se enfoca en identificar y explotar vulnerabilidades dentro de un conjunto definido de sistemas de la compañía, un *red team* tiene un enfoque más orientado a objetivos específicos y busca acceder a ellos explotando debilidades relevantes en cualquier parte de una organización, sin la intención de listar todas las vulnerabilidades existentes [30].

El valor principal de un *red team* radica en su capacidad para simular cómo un adversario real podría atacar a la organización y cómo responde el *blue team* a tales ataques. Las tácticas, técnicas y procedimientos utilizados por un *red team* se basan en los de actores de amenazas maliciosas del mundo real, con el objetivo de identificar deficiencias en la respuesta de seguridad de la organización [30].

2.4.4. PTES

El *Penetration Testing Execution Standard* (PTES) es un conjunto de reglas estandarizadas diseñado para proporcionar un marco común que guíe todos los procesos de *pentesting*. Desarrollado por un equipo diverso de profesionales de la seguridad de la información, incluyendo expertos de instituciones financieras, proveedores de servicios y proveedores de seguridad, PTES responde a la necesidad de un estándar integral y actualizado en el campo del *pentesting* [31].

Este estándar no solo ofrece orientación a los profesionales de seguridad sobre cómo realizar *pentestings* de manera efectiva, sino que también informa a las empresas sobre qué esperar de estas pruebas.

El PTES estable un proceso de *pentesting* de 7 fases, como se observa en la Figura 2.2, a continuación se describen estas fases [32] [33]:

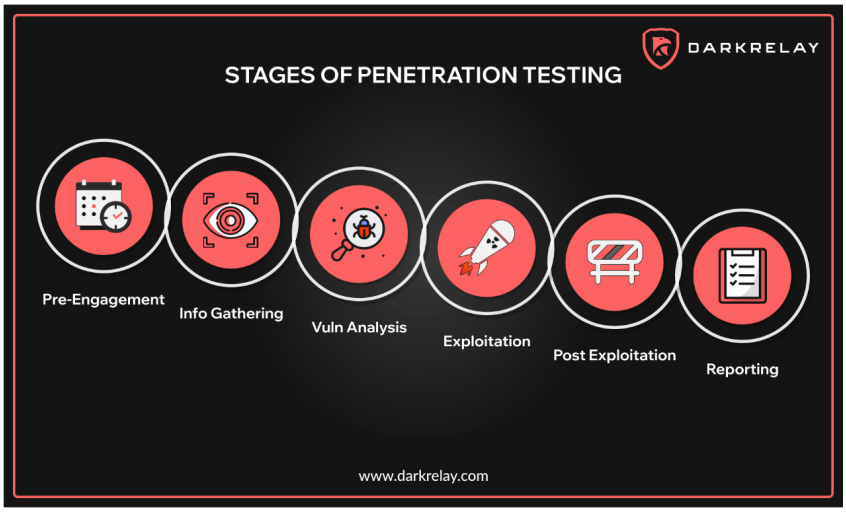

Figura 2.2: Fases de PTES [31].

- **Interacción previa**: Esta fase describe las acciones previas al compromiso y la descripción del alcance, enfocándose en qué considerar en la prueba para proporcionar directrices importantes.
- **Recopilación de información:** Se especifica la recopilación de información necesaria para producir una representación comprensible de la empresa, su infraestructura de TI, empleados, productos y operaciones.
- **Modelado de amenazas:** Basado en las etapas anteriores, esta fase cubre los fundamentos del modelado de amenazas, enfatizando la necesidad de un modelo fiable que describa adecuadamente los riesgos y sus limitaciones para la organización analizada.
- **Análisis de vulnerabilidades:** También conocida como evaluación de vulnerabilidades, esta fase describe los aspectos clave que debe cubrir el análisis de vulnerabilidades, incluyendo la identificación y categorización de las mismas, y la formulación de un plan para abordar los problemas más graves.
- **Explotación:** En esta fase, el *pentester* explota efectivamente el objetivo, utilizando el conocimiento adquirido en las fases anteriores para acceder a datos o información de otra manera inaccesibles, estableciendo el acceso a un servicio o máquina.
- **Post-explotación:** Una vez que el *pentester* tiene control sobre un objetivo, el principal objetivo es permanecer indetectado y asegurar el control futuro sobre los sistemas explotados. Esta fase también incluye cubrir cualquier rastro de las actividades manipuladoras y preparar el sistema explotado para un uso futuro.
- **Reporte:** El *pentester* crea un documento que detalla todas las vulnerabilidades de se-

guridad descubiertas durante la prueba. Este informe está diseñado para informar a los responsables sobre las vulnerabilidades en la infraestructura y el diseño del código, y debe ser tomado en serio por la gestión para mitigar futuras debilidades y exposiciones en sus sistemas.

2.5. Introducción al *hardening*

El *hardening* es un conjunto de prácticas de seguridad diseñadas para fortalecer las defensas de un sistema informático. Este proceso proactivo se centra en minimizar la superficie de ataque al configurar de manera segura los componentes del sistema, aplicar parches de software, y eliminar servicios y funciones innecesarias. La estrategia involucra la implementación de múltiples capas de medidas de seguridad que deben ser superadas antes de que un ataque pueda comprometer un sistema, abarcando desde el nivel de *host*, aplicación, y sistema operativo, hasta las configuraciones de usuario y las capas físicas. Al endurecer un sistema, las organizaciones protegen sus datos contra vulnerabilidades conocidas y potenciales, mejorando la resiliencia frente a ciberataques y reduciendo significativamente el número de violaciones de seguridad [34].

2.5.1. Tipos de *Hardening*

Aunque existe varias formas y métodos para hardenizar un sistema, parte de estas se han podido agrupar de la siguiente manera [34, 35]:

- ***Hardening* en servidores:** Es un proceso esencial para asegurar las capas de *hardware*, *software* y *firmware* de un sistema. Incluye la aplicación de parches, actualizaciones, *multi-factor authentication* (MFA) y uso de contraseñas robustas. Este proceso se enfoca en proteger los datos, puertos, componentes y permisos del servidor para reducir la superficie de ataque, teniendo en cuenta las necesidades específicas de seguridad de cada servidor, como las diferencias entre un servidor web y uno de bases de datos.
- ***Hardening* en sistemas operativos:** Es fundamental para fortalecer la seguridad en los servidores, controlando los permisos otorgados al software de aplicaciones. Este proceso incluye medidas como la eliminación de controladores innecesarios, la limitación de la creación de usuarios, y la encriptación de discos duros. El objetivo principal del endurecimiento del sistema operativo es proteger tanto el hardware como el software

contra ataques cibernéticos, implementando una gestión de parches que monitorea e instala actualizaciones y paquetes de servicio de manera automática, aumentando así la protección en todos los niveles del sistema.

- ***Hardening* de la red:** Implica configurar *firewalls*, deshabilitar servicios innecesarios, auditar privilegios de acceso y cifrar el tráfico, entre otras medidas. Este proceso busca fortalecer los dispositivos de red para contrarrestar accesos no autorizados y prevenir la explotación de vulnerabilidades en la gestión y configuración de dispositivos. Además, el uso de *software* de prevención y detección de intrusiones es fundamental para monitorear actividades sospechosas y prevenir el acceso no autorizado a la infraestructura de la red.
- ***Hardening* de aplicaciones:** Este proceso busca proteger el software instalado en la red e implica actualizar o reescribir el código de la aplicación para mejorar su seguridad, además de utilizar otras soluciones de seguridad basadas en software como antivirus, gestión de contraseñas, uso de *firewalls* y cifrado. Una parte importante del endurecimiento de aplicaciones es la gestión de parches y actualizaciones para abordar vulnerabilidades, a menudo mediante automatización para garantizar la implementación eficiente de las últimas protecciones.
- ***Hardening* de base de datos:** Implica adoptar una serie de medidas de seguridad destinadas a reducir las vulnerabilidades en las bases de datos. Estas medidas incluyen controlar los privilegios de acceso a la base de datos, deshabilitar funciones innecesarias y cifrar la información almacenada. Además, es fundamental aplicar parches a la base de datos, *using role-based access control* (RBAC), y restringir los privilegios administrativos. El objetivo de estas prácticas es proteger tanto los repositorios de datos como el *software* utilizado para interactuar con dichos datos, fortaleciendo así la seguridad y la integridad de la información crítica.
- ***Hardening* a nivel físico:** Se enfoca en proteger los recursos tecnológicos en un espacio físico. Este tipo de seguridad involucra la implementación de diversas medidas como cerraduras, control de acceso mediante puertas, sistemas de circuito cerrado de televisión y alarmas de incidentes. Además, se utilizan tecnologías de acceso restringido y sistemas de detección de intrusiones perimetrales para prevenir el acceso no autorizado.

2.6. CIS *benchmarks*

Los *Center for Internet Security* (CIS) *benchmarks* son un conjunto de prácticas recomendadas, desarrolladas por el CIS, para la configuración segura de sistemas y tecnologías con el objetivo de fortalecer la seguridad frente a ataques cibernéticos. Estas guías de configuración, avaladas por sectores gubernamentales, empresariales, industriales y académicos, son establecidas a través de un consenso de expertos en ciberseguridad de todo el mundo. Los CIS *benchmarks* proporcionan recomendaciones detalladas para endurecer desde sistemas operativos y dispositivos móviles hasta plataformas de servicios en la nube y bases de datos. Estas directrices no solo mejoran las defensas de seguridad sino que también ayudan a las organizaciones a demostrar el cumplimiento con diversas regulaciones y marcos de la industria [36].

En las Tablas 2.1 y 2.2, se observan ejemplos específicos de estas recomendaciones de seguridad aplicadas a los sistemas operativos Windows y Ubuntu, respectivamente. Estas tablas detallan las prácticas de seguridad más importantes y su aplicación sugerida, facilitando así la adopción de las mismas para lograr configuraciones más seguras y efectivas.

CIS Benchmark	**Descripción**
Configuración de la cuenta de administrador	Establecer una contraseña única y segura para la cuenta de administrador para evitar accesos no autorizados.
Uso de BitLocker	Implementar el cifrado de disco completo con BitLocker para proteger los datos en caso de acceso físico no autorizado.
Configuración de Windows Defender	Optimizar los ajustes de Windows Defender para maximizar la protección contra malware y otras amenazas.
Deshabilitar o restringir RDP	Deshabilitar el Protocolo de Escritorio Remoto (RDP) o restringirlo a redes confiables para reducir la superficie de ataque.
Auditoría y monitoreo de eventos	Habilitar la auditoría de eventos para monitorizar y registrar eventos críticos del sistema, mejorando la capacidad de respuesta ante incidentes.

Tabla 2.1: CIS Benchmarks para Windows [37].

CIS Benchmark	Descripción
Configuración de la cuenta de root	Establecer una contraseña segura para la cuenta root y considerar la desactivación cuando no sea necesaria.
Configuración del cortafuegos UFW	Activar y configurar UFW (Uncomplicated Firewall) para gestionar el tráfico de red y proteger el sistema.
Restricciones de acceso SSH	Restringir el acceso SSH modificando las configuraciones por defecto para evitar accesos remotos no autorizados.
Permisos de archivos y directorios	Asegurar que los permisos de archivos y directorios sean los adecuados para evitar accesos no autorizados.
Cuentas de usuario innecesarias	Eliminar o deshabilitar cuentas de usuario que no sean necesarias para minimizar riesgos de seguridad.

Tabla 2.2: CIS Benchmarks para Ubuntu [38].

2.7. CVSS

Common Vulnerability Scoring System (CVSS) es un estándar abierto de la industria diseñado para evaluar la severidad de las vulnerabilidades de seguridad en sistemas informáticos. CVSS ofrece un marco metodológico que permite calcular la severidad de una vulnerabilidad a través de una fórmula que contempla una serie de métricas. Estas métricas buscan aproximarse tanto a la facilidad de explotación como al impacto potencial de dicha explotación. Este sistema de puntuación ayuda a los profesionales de la seguridad a priorizar adecuadamente sus respuestas y los recursos necesarios frente a las amenazas detectadas, basándose en una metodología consistente y transparente [39].

2.7.1. Métricas de puntuación

Como se representa en la Figura 2.3, las métricas se dividen en 3 partes, las cuales se describen a continuación [41]:

- **Métrica base:** Evalúa las características inherentes a una vulnerabilidad que son independientes del entorno o del tiempo, resultando en el CVSS Base Score. Este puntaje se compone de dos subpuntuaciones: la Subpuntuación de Explotabilidad y la Subpuntuación de Impacto. La Subpuntuación de Explotabilidad se basa en las características del componente vulnerable, evaluando qué tan fácil es explotar dicha vulnerabilidad; mien-

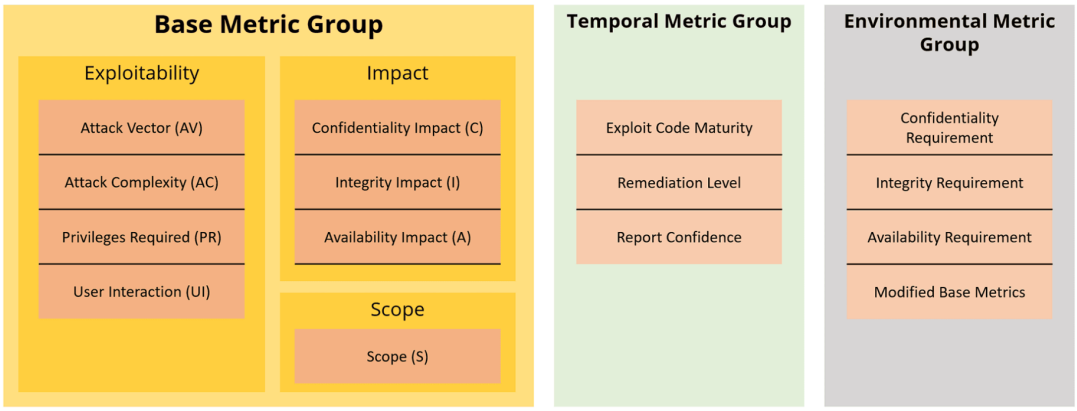

Figura 2.3: Métricas de Puntuación CVSS [40].

tras más fácil, mayor es la puntuación. Las métricas involucradas incluyen el Vector de Ataque, que mide la facilidad con la que un atacante puede explotar una vulnerabilidad; la Complejidad del Ataque, que define los requisitos para que un ataque sea exitoso; los Privilegios Requeridos, que indican el nivel de acceso necesario; y la Interacción del Usuario, que determina si se necesita la ayuda de otro usuario para ejecutar el ataque. Cada métrica se califica según un conjunto de valores propio, contribuyendo al cálculo del puntaje base.

- **Métrica temporal:** Ajusta la puntuación base de una vulnerabilidad tomando en cuenta factores adicionales que pueden variar con el tiempo. Esta métrica incluye la Madurez del Código de Explotación, que evalúa la probabilidad de que una vulnerabilidad sea explotada según la disponibilidad de código de explotación en uso activo, variando desde inexistente hasta código funcional o de alto riesgo. También considera el Nivel de Remediación, que refleja la facilidad con la que se puede solucionar una vulnerabilidad, desde soluciones temporales hasta correcciones oficiales completas. Finalmente, la Confianza del Informe mide la certeza con la que se puede afirmar la existencia y las características de una vulnerabilidad, desde informes contradictorios hasta vulnerabilidades completamente confirmadas y reproducibles. El resultado de incorporar estos factores es la puntuación temporal de CVSS, que ofrece una perspectiva más dinámica y actualizada sobre cómo las amenazas están siendo explotadas y las opciones disponibles para su mitigación.

- **Métrica ambiental:** Ajusta la puntuación base de una vulnerabilidad tomando en cuenta factores específicos del entorno en el que se encuentra. Esta métrica puede aumentar o disminuir la puntuación de CVSS basada en consideraciones como la criticidad de los datos afectados o cambios en la vulnerabilidad debido a una configuración específica

del sistema o la constelación de la red. La métrica ambiental incluye un Subpuntaje de Requerimientos de Seguridad, que se basa en los tres componentes de la puntuación de impacto (Confidencialidad, Integridad y Disponibilidad) como se estiman en un entorno específico. Además, la Puntuación Base Modificada reevalúa las métricas que componen la puntuación base teniendo en cuenta las condiciones particulares del entorno de la organización. Esta métrica ambiental de CVSS refleja cómo las características particulares de un escenario afectan la gravedad y el impacto de una vulnerabilidad, proporcionando una evaluación más precisa y contextualizada de su potencial de daño.

2.7.2. Clasificación de vulnerabilidades

Actualmente, existen varias versiones de la puntuación CVSS, pero la versión 3.1 es la más utilizada. Esta versión se distingue de las anteriores (2.0 y 3.0) por ofrecer una subdivisión más detallada en el rango más alto de severidad, permitiendo un análisis más preciso de las vulnerabilidades. La puntuación se escala de 0 a 10 y se clasifica de la siguiente manera [42, 43]:

- **Logs:** Estos reciben una puntuación de 0.0 y generalmente incluyen registros en los informes de escaneo que documentan actividades no vinculadas directamente con vulnerabilidades. Esto incluye el inicio y cierre de sesión de escaneos, mensajes sobre la configuración de escaneos y cualquier mensaje de error que ocurra durante el mismo.
- **Baja (0.1 - 3.9):** Vulnerabilidades con bajo impacto en la seguridad del sistema que es poco probable que sean explotadas para causar un daño significativo. Estas pueden ser remediadas durante periodos de mantenimiento regular, especialmente si su solución es sencilla y puede implementarse sin interrupciones significativas.
- **Media (4.0 - 6.9):** Incluye vulnerabilidades con un impacto moderado que generalmente requieren condiciones específicas para su explotación, como configuraciones inadecuadas o ciertos privilegios. Estas amenazas deben ser tratadas en un plazo razonable para evitar escaladas o explotaciones en escenarios menos comunes.
- **Alta (7.0 - 8.9):** Vulnerabilidades que en la versión 3.1 varían en este rango, mientras que en versiones anteriores podrían alcanzar hasta 10.0. Estas vulnerabilidades permiten a los atacantes ejecutar código arbitrario o acceder cantidades significativas de datos sensibles, afectando significativamente la operación de un sistema. Requieren intervención inmediata.

- **Crítico (9.0 - 10.0):** Esta subcategoría, específica de la versión 3.1, incluye vulnerabilidades con un impacto extremadamente alto. Permiten a los atacantes tomar control completo de un sistema, robar datos altamente sensibles, realizar modificaciones severas, o permitir acceso no autorizado a partes críticas de la infraestructura. Como las de alta severidad, requieren atención inmediata.

2.8. Recursos y herramientas

2.8.1. GLPI

Gestionnaire Libre de Parc Informatique (GLPI) es una aplicación de código abierto y multiplataforma diseñada para optimizar la gestión de sistemas de información dentro de las organizaciones. Esta herramienta permite organizar inventarios de hardware, software y dispositivos periféricos. Además, maneja de manera eficiente incidentes, problemas, cambios y alertas, facilitando así una gestión integral de las tecnologías de la información [44].

Una característica destacable de GLPI es su API, que proporciona comunicación con otros *softwares*, y su facilidad al integrar diferentes tipos de extensiones según las necesidades de la organización. Para este proyecto, se empleará *GLPI Inventory*, una extensión que automatiza el inventario de los activos TI mediante la implementación de agentes que, una vez instalados en los dispositivos de la red, recopilan información precisa sobre el *hardware* y el *software*. Este procedimiento incluye detalles sobre la CPU, memoria RAM, almacenamiento, sistemas operativos, paquetes de *software* y sus respectivas versiones y actualizaciones [44].

Los datos recolectados por estos agentes son enviados de forma segura y periódica a la base de datos de GLPI. Esto proporciona un control preciso y continuamente actualizado de los activos de la organización, asegurando así una gestión efectiva y eficiente [45].

2.8.2. OpenCVE

OpenCVE es una plataforma de código abierto diseñada para mejorar la gestión y el seguimiento de vulnerabilidades conocidas como *Common Vulnerabilities and Exposures* (CVE). Al integrarse con bases de datos públicas, permite a los usuarios filtrar y organizar la información según sus necesidades específicas, facilitando así la priorización de respuestas a las amenazas más severas [46].

Además de ofrecer funciones de monitoreo, OpenCVE permite personalizar cómo se presentan y gestionan las alertas. Los usuarios pueden configurar notificaciones específicas para ciertos tipos de vulnerabilidades o para aquellas que afecten a activos críticos de la organización [46].

Uno de los puntos fuertes de OpenCVE es su interfaz web accesible, que simplifica la interacción con la plataforma. Además, su API permite una integración fluida con otros servicios, mejorando la capacidad de las organizaciones para gestionar su seguridad de manera efectiva [46, 47].

2.8.3. Nmap

Network Mapper (Nmap) es una herramienta de código abierto ampliamente utilizada en la exploración de redes y la auditoría de seguridad. Esta herramienta utiliza paquetes IP para determinar diversas características de la red, incluyendo [48]:

- **Descubrimiento de *Hosts*:** Nmap es capaz de identificar qué *hosts* están activos en la red, proporcionando información sobre la disponibilidad de los mismos.
- **Identificación de servicios:** Mediante el análisis de puertos, Nmap puede determinar qué servicios están en ejecución, así como las versiones de software y los sistemas operativos de los dispositivos. Esta función es necesaria para los analistas de seguridad que buscan detectar posibles vulnerabilidades.
- ***Scripting* Flexible:** Gracias a su conjunto de *scripts Nmap Scripting Engine* (NSE), Nmap puede ser adaptado para realizar una variedad de tareas en exploración de redes y detección de vulnerabilidades. Esto permite a los usuarios personalizar sus exploraciones de acuerdo a necesidades específicas.

2.8.4. *TheHarvester*

TheHarvester es una herramienta de código abierto ampliamente utilizada para la recolección de información y *Open Source Intelligence* (OSINT). Los analistas de seguridad informática la emplean para identificar y recopilar datos sobre objetivos humanos y digitales durante las etapas iniciales de un *pentesting* o *red teaming*. Esta herramienta es especialmente importante en el reconocimiento pasivo, ya que proporciona datos como correos electrónicos, dominios y subdominios de una organización. También facilita la identificación de nombres y roles dentro de una empresa. Para obtener esta información, *TheHarvester* se integra con diversos motores

de búsqueda como Bing, Yahoo y Duckduckgo. Además, utiliza herramientas de monitoreo de seguridad como VirusTotal, Shodan y Urlscan [49].

2.8.5. Nuclei

Nuclei es una herramienta de código abierto que los analistas de seguridad utilizan ampliamente para realizar exploraciones rápidas y eficientes. Está diseñada para identificar vulnerabilidades en sistemas de aplicaciones web y funciona mediante el uso de plantillas predefinidas, lo que permite detectar vulnerabilidades comunes de forma efectiva [50].

Soporta una variedad de protocolos para realizar sus escaneos, incluyendo TCP, DNS, HTTP y SSL. La flexibilidad de Nuclei en la manipulación de diferentes tipos de protocolos permite a los profesionales de la seguridad evaluar de manera rápida y exhaustiva las vulnerabilidades en sus sistemas [50].

2.8.6. OpenVas

OpenVAS es un escáner de vulnerabilidades de código abierto diseñado para detectar fallas de seguridad o debilidades en activos que pueden ser explotadas. Entre sus funciones, destacan las pruebas autenticadas y no autenticadas, el análisis de protocolos de seguridad y de Internet de alto y bajo nivel, ajustes de rendimiento para exploraciones a gran escala y un potente lenguaje de programación interno para implementar cualquier tipo de prueba de vulnerabilidad [51, 52].

Una vez instalado y configurado, el servidor puede utilizarse de dos maneras: mediante la línea de comandos (OpenVAS CLI) o con la interfaz web (*Greenbone Security Assistant*). Esta última cuenta con dos servicios que interactúan: *OpenVAS Manager* y *OpenVAS Scanner*. El servicio *Manager* se encarga de tareas como el filtrado o clasificación de los resultados de análisis, el control de las bases de datos que contienen la configuración o los resultados y la administración de usuarios, incluyendo roles y grupos. Por otro lado, el escáner ejecuta las *Network Vulnerability Tests* (NVT), que contienen rutinas para comprobar la presencia de problemas de seguridad específicos, conocidos o potenciales, en los sistemas. Estas pruebas se actualizan constantemente, con nuevos registros semanales, lo que permite ejecutar *tests* de vulnerabilidades recién descubiertas [53].

2.8.7. *KeePass 2*

KeePass 2 es un gestor de contraseñas gratuito y de código abierto disponible para diversas plataformas. Permite a los usuarios almacenar y gestionar sus contraseñas de manera segura utilizando encriptación de extremo a extremo para los archivos .kdbx, que son contenedores de contraseñas [54]. Además, incluye una herramienta para generar contraseñas seguras y complejas, lo que facilita la creación de credenciales difíciles de descifrar.

2.9. Trabajos Relacionados

En esta sección se realizó una revisión de la literatura existente sobre las tecnologías y metodologías de seguridad en redes industriales, particularmente aquellas enfocadas en el *hardening* y el *pentesting*. Los trabajos revisados abarcan desde estrategias de defensa cibernética hasta prácticas específicas de seguridad, incluyendo los desarrollos recientes en *hardening* autónomo y *pentesting* adaptativas.

El estudio [55] explora las técnicas de seguridad de red específicas para los ICS, enfocándose en la importancia de los *pentesting* como una técnica defensiva clave. Los autores discuten cómo la integración cada vez mayor de los sistemas de control industrial con Internet ha elevado la necesidad de mejorar las prácticas de seguridad para proteger infraestructuras críticas. El trabajo introduce métodos detallados de *pentesting* que incluyen escaneo de puertos, pruebas de inyección dinámica externa y detección de troyanos de intrusión, resaltando su relevancia para anticipar y mitigar ataques cibernéticos en sistemas industriales. Esta investigación es relevante ya que proporciona un marco comparativo y metodológico para la aplicación de *pentesting* en un contexto de seguridad de redes industriales. Los métodos propuestos por [55] complementan y amplían las técnicas que se planean desarrollar y adaptar para mejorar la seguridad de la micro-red en el laboratorio, destacando la utilidad práctica de combinar diferentes enfoques de prueba para formar un sistema de defensa exhaustivo.

En el *survey* [56] se profundiza en las técnicas avanzadas de seguridad de red aplicadas a los ICS, enfocándose en la identificación y mitigación de riesgos cibernéticos que enfrentan estos sistemas en entornos industriales contemporáneos. Los autores discuten estrategias detalladas para evaluar y fortalecer la infraestructura de seguridad de los sistemas de control, incluyendo la aplicación de *pentesting* como medio para detectar vulnerabilidades críticas y mejorar las medidas de seguridad.

En el ámbito de la seguridad de sistemas SCADA, [57], realizaron un estudio comparativo entre diferentes estándares de ciberseguridad específicos para SCADA y la norma internacional ISO/IEC 17799 (ahora ISO/IEC 27002). Su investigación destaca cómo los estándares específicos de SCADA tienden a enfocarse más en medidas de seguridad técnicas, como *firewalls* y sistemas de detección de intrusiones, mientras que ISO/IEC 17799 se inclina más hacia medidas organizativas. Este trabajo apoya la necesidad de un *framework* de *pentesting* que no solo se base en conocimientos generales de ciberseguridad, sino que también considere las peculiaridades de los sistemas SCADA, adaptando las prácticas y herramientas para evaluar de manera efectiva su seguridad frente a amenazas actuales.

Los autores de [58] investigan las amenazas cibernéticas y las soluciones potenciales para proteger los ICS. La investigación enfatiza los riesgos presentados por las soluciones de *hardware* y *software* de estantería que están comúnmente disponibles y son conocidas por los expertos en seguridad TI, pero que también pueden ser explotadas como vectores de ataque para infiltrar entornos industriales típicamente carentes de medidas de seguridad rigurosas. El estudio utiliza el protocolo de comunicación Modbus de una estación real de compresores para demostrar ataques de Man-in-the-Middle (MITM) y *Legal-Client-to-Server* (LCSA), que explotan las vulnerabilidades inherentes del protocolo. Este artículo proporciona una visión sobre soluciones de seguridad accesibles que podrían ser aprovechadas potencialmente con fines maliciosos, demostrando la necesidad crítica de desarrollar estrategias de *hardening* robustas dentro de los marcos industriales para mitigar dichas vulnerabilidades. Esta investigación podría complementar el *framework* de *pentesting* al proporcionar un ejemplo concreto de cómo las tecnologías existentes y fácilmente accesibles pueden representar amenazas significativas para los sistemas industriales, subrayando la importancia de un enfoque proactivo hacia la ciberseguridad dentro del entorno de la Micro-Red.

El artículo [59] describe un enfoque innovador para mejorar la seguridad de los ICS mediante la autonomía en el *hardening*, una práctica importante para proteger contra amenazas cibernéticas. Los autores presentan un *framework* que permite a los sistemas endurecerse a sí mismos de manera periódica y autónoma, asegurando que cada dispositivo cumpla con una línea base de seguridad adecuada a su función y capacidad. El estudio destaca la importancia de esta autonomía, dada la creciente complejidad y heterogeneidad de los sistemas ICS, junto con frecuentes cambios topológicos y de configuración que hacen inviable la verificación manual de la seguridad del sistema.

3. Metodología

Este capítulo detalla cada una de las fases consideradas para la implementación del *framework* de *pentesting* Daniz-Red, incluyendo la implementación de auditorías de *hardening* y la corrección de vulnerabilidades basadas en los resultados del *pentesting* realizado en el laboratorio de Micro-Red de la Universidad de Cuenca. El capítulo comienza en la Sección 3.1, donde se describe cómo se gestionaron los activos dentro del laboratorio mediante el uso de la herramienta *Gestionnaire Libre de Parc Informatique* (GLPI). En la Sección 3.2, se muestra la implementación del *framework* Daniz-Red en el sistema *Industrial Control System* (ICS). Para finalizar, la Sección 3.3 presenta el proceso para corregir y prevenir fallos de seguridad, basado en auditorías de *hardening* y la corrección de vulnerabilidades mediante el análisis de los resultados de la fase de *pentesting*.

3.1. Gestión de activos

Para realizar un *pentesting* con un enfoque *white box*, es imprescindible tener acceso completo a la arquitectura, diseño y código fuente de los sistemas. Debido a la ausencia de un inventario completo en el laboratorio, fue necesario implementar una herramienta de código abierto que gestionara de manera eficiente el inventario de activos, simplificara su administración y contribuyera a la prevención de vulnerabilidades conocidas, en particular en los activos de Tecnologías de Operación (TO). Para optimizar los recursos y evitar duplicidades, se optó por emplear GLPI, ya utilizado en un estudio previo titulado “Diseño e implementación de una Arquitectura de Ciberseguridad para la Micro-red de la Universidad de Cuenca”. Como este trabajo de titulación no estaba disponible en el repositorio institucional hasta la redacción de este documento, se recomienda contactar directamente al autor, Boris Guachicullca, para obtener los detalles necesarios [60]. La presente sección se estructura en dos fases.

3.1.1. Automatización de inventarios con GLPI

Con la herramienta GLPI operando en los servidores de la Micro-Red, se identificó la necesidad de realizar un inventario de los activos. Para ello, se decidió implementar la extensión GLPI *Inventory*, cuyo proceso de implementación se detalla en la Sección 5.2. Una vez habilitada la extensión, el siguiente paso fue instalar el agente GLPI tanto en servidores como en ordenadores, particularmente en visualizadores y servidores de seguridad, siguiendo las implementaciones y configuraciones mostradas en la Sección 5.2.

Tras la implementación del *Inventory* y los agentes de GLPI, fue necesario registrar el resto de activos del laboratorio. La extensión permite realizar diversas tareas, para las cuales es necesario definir la hora de inicio, la hora de finalización, el número de agentes que se utilizarán y el tipo de tarea, que puede incluir descubrimiento de red, inventario *Simple Network Management Protocol* (SNMP), entre otros, como se observa en la Figura 3.1.

En cuanto al inventariado de activos, se aplicó el método de reconocimiento de red. Esto fue necesario porque algunos activos, como los PLC, carecían de los archivos de programación y configuración necesarios para implementar protocolos como SNMP en su versión 3.

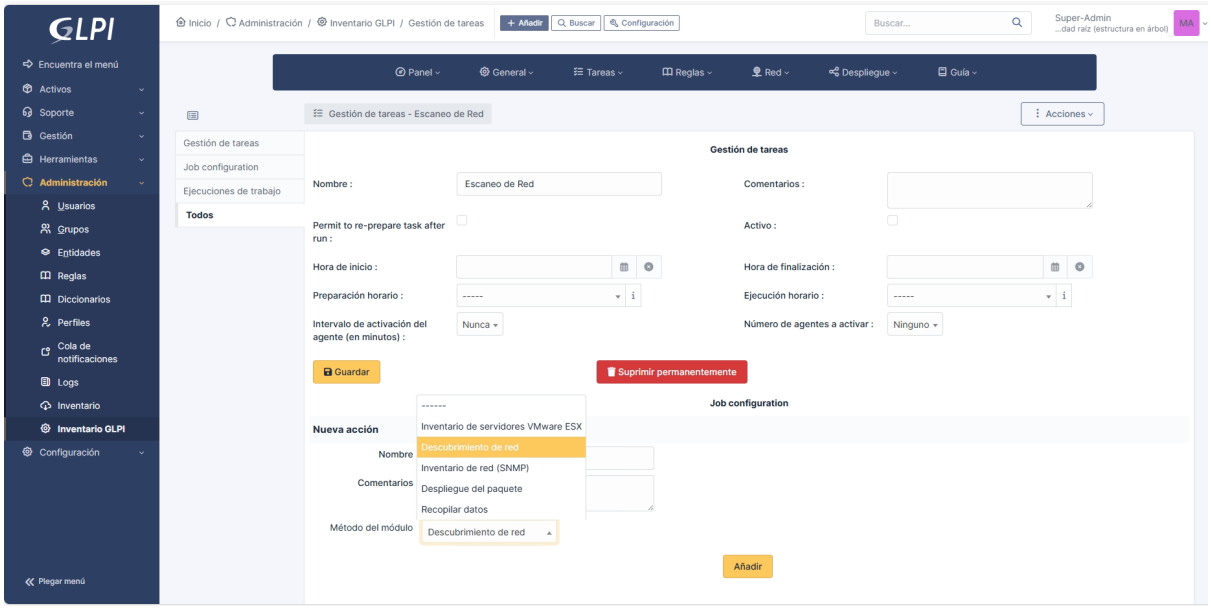

Figura 3.1: Configuración de tareas en GLPI.

Cuando un dispositivo suministra información mediante el método de descubrimiento de red, este pasa directamente a formar parte de la base de datos de las diferentes categorías de activos de GLPI. Cuando esto no ocurre, la extensión ubica los activos en la sección de dispositivos desconocidos, lo que requiere que se realice un inventariado manual de cada uno de los dispositivos. La interfaz para ingresar los datos y convertirlos en un dispositivo de red, computadora, entre otros, se muestra en la Figura 3.2.

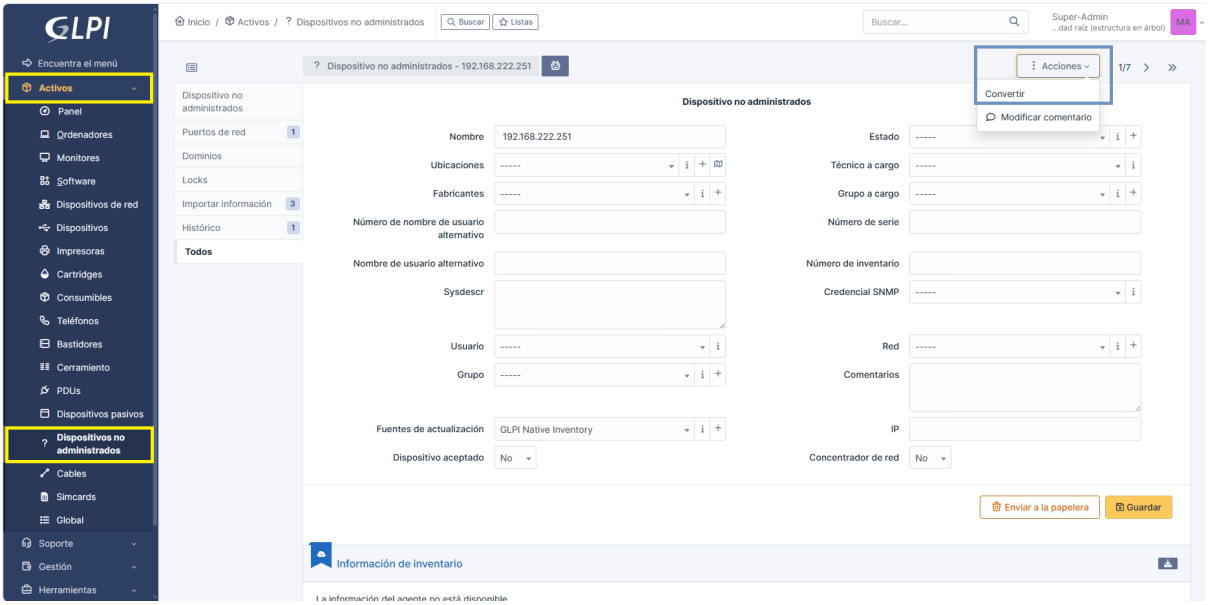

Figura 3.2: Inventario de dispositivos desconocidos en GLPI.

Para levantar la información de los activos en el servidor GLPI, se utilizó el levantamiento de información y topología del laboratorio de Micro-Red, suministrado por [60].

3.1.2. Obtención de CVE con GLPI y OpenCVE

Una vez inventariados todos los activos del laboratorio, se procedió a implementar un sistema capaz de gestionar la información de los activos y sus vulnerabilidades, facilitando así su corrección o configuración adecuada. El diagrama de flujo del sistema se muestra en la Figura 3.3. Para su funcionamiento, es necesario introducir los datos requeridos en un archivo de texto.

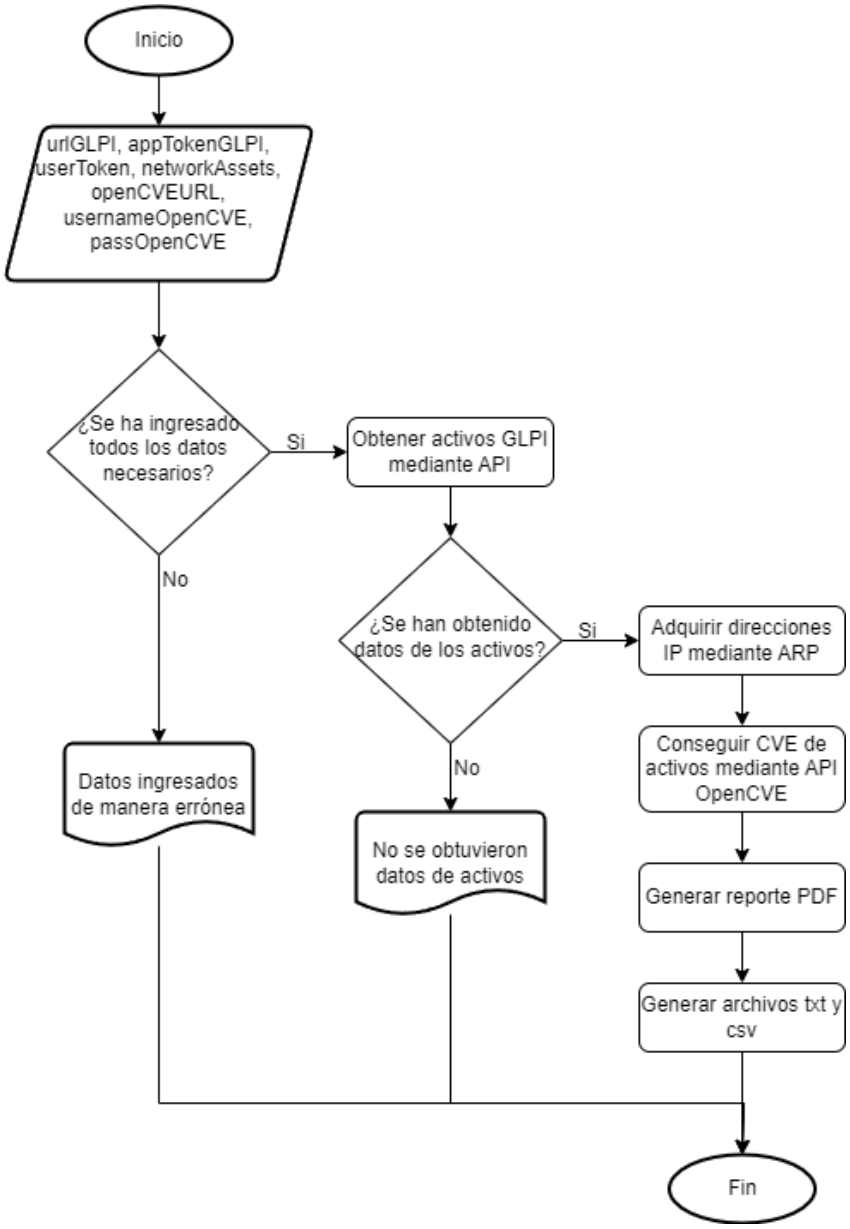

Figura 3.3: Diagrama de flujo para procesamiento de activos y sus vulnerabilidades.

El sistema está compuesto por cuatro módulos. El primero se encarga de realizar y procesar la información de los activos a través de la *Application Programming Interface* (API) de GLPI. Se extrajeron datos de las categorías de computadores, dispositivos de red y periféricos, con el objetivo de proporcionar al administrador información esencial como el tipo, modelo y la dirección IP del *Hardware* del laboratorio. Durante la inspección de los datos suministrados por la API, se observó que en los adaptadores de red no se obtenían direcciones IP, por tanto, se procedió a extraer las direcciones MAC de los dispositivos. El proceso utilizado para esta extracción se detalla en el Algoritmo 1.

Algoritmo 1: Interactuar con la API de GLPI para gestionar información de activos.

Entradas: urlGLPI, appTokenGLPI, userTokenGLPI, categorías
Salida: Datos de activos recopilados por categoría

```
Iniciar sesión con GLPI y obtener un token de sesión.
Si no se obtiene el token de sesión Entonces
    Imprimir ”No se pudo establecer una sesión”Terminar algoritmo.
Definir encabezados para las siguientes peticiones con el token de sesión.
Para cada categoría hacer
    Solicitar los activos de la categoría desde GLPI.
    Si la respuesta es exitosa Entonces
        Ror cada activo en respuesta hacer
            Inicializar la estructura de datos para el activo.
            Si activo tiene modelo específico Entonces
                Obtener y ajustar detalles del modelo.
            Si activo tiene tipo específico Entonces
                Obtener y ajustar detalles del tipo.
            Si activo tiene puertos de red Entonces
                Filtrar direcciones MAC válidas.
            Almacenar datos del activo.
    Sino
        Mostrar error y continuar con la siguiente categoría.
Devolver datos de activos por categoría.
```

El objetivo de extraer las direcciones MAC fue implementar un módulo capaz de obtener las direcciones IP utilizando el protocolo ARP. Para esta tarea, se empleó la librería Scapy, cuya aplicación se detalla en el Algoritmo 2.

Algoritmo 2: Obtener direcciones IP por medio del solicitudes ARP

Entradas: Activos, subredes, categorías

Salida: Direcciones IP de los activos.

```
1  Para cada categoría hacer
2      Obtener la lista de activos de la categoría.
3      Ror cada activo hacer
4          Si el activo tiene direcciones MAC Entonces
5              Ror cada dirección MAC hacer
6                  Ror cada subred hacer
7                      Configurar paquete ARP y Ethernet para broadcast en la subred.
8                      Ensamblar paquete completo con Ethernet y ARP.
9                      Enviar paquete y recibir la respuesta.
10                     Ror cada pareja de paquetes enviados y recibidos hacer
11                         Si MAC recibida es igual a MAC del activo Entonces
12                             Almacenar dirección IP.
13 Devolver direcciones IP de activos por categoría.
```

El siguiente paso consiste en obtener los *Common Vulnerabilities and Exposures* (CVE) de los activos, utilizando la API de OpenCVE. Actualmente, esta API no posee un generador de *tokens* por lo que utiliza un método sencillo de autenticación *Hypertext Transfer Protocol* (HTTP). Para acceder, se introducen las credenciales de una cuenta previamente creada en OpenCVE. El proceso se detalla en el módulo presentado en el Algoritmo 3.

Algoritmo 3: Obtener CVE de modelos de activos

Entradas: datosDeActivos, urlOpenCVE, usuario, contraseña, categorías

Salida: Activos con CVE.

```
1  Ror cada categoría hacer
2      Ror cada Modelo hacer
3          Realizar solicitud GET a OpenCVE, usando autenticación básica HTTP.
4          Si código de estado de la respuesta es igual a 200 Entonces
5              Convertir respuesta a JSON.
6              Ror cada CVE en respuesta JSON hacer
7                  Añadir id de CVE a lista de CVE.
8              Crear diccionario con los modelos y lista de id de CVE.
9              Añadir diccionario a lista de datos de CVE.
10 Devolver lista con datos de CVE.
```

Con la información de los activos, se procede con el cuarto módulo, que se encarga de generar un informe en formato PDF. Este informe consta de dos partes: la primera incluye una tabla con los identificadores de los activos, nombres, modelos, tipos, direcciones MAC e IP. La segunda sección detalla las vulnerabilidades encontradas mediante OpenCVE para los modelos de los activos. Esta sección se ha diseñado de manera independiente porque varios activos comparten el mismo modelo, y su inclusión en la tabla principal la haría extensa y repetitiva. La secuencia de este módulo se detalla en el Algoritmo 4.

Algoritmo 4: Generar informe en formato PDF con información de los activos

Entradas: datosDeActivos, rutaResultados, nombreArchivo

Salida: Archivo PDF creado y almacenado en la ruta.

```
Crear documento PDF con tamaño de página y márgenes establecidos.
Configurar estilos de documento.
Inicializar lista de elementos del documento.
Agregar título del documento al inicio de la lista de elementos.
Inicializar diccionario para relacionar modelos con CVEs.
Ror cada categoría hacer
    Agregar encabezado de categoría a los elementos del documento.
    Inicializar datos de la tabla con encabezados como ID, Nombre, Modelo, Tipo, Direcciones MAC, IP.
    Ror cada Activo hacer
        Agregar una nueva fila con los datos de los activo ajustados al tamaño de la tabla.
        Si el activo tiene CVEs asociados y el modelo no está en el diccionario, añadir al diccionario.
    Establecer estilos de la tabla y agregar la tabla a los elementos del documento.
    Agregar espacio después de la tabla.
Agregar sección de resumen de CVEs por modelo.
Ror cada modelo hacer
    Agregar nombre del modelo y lista de CVEs asociados a los elementos del documento.
Crear y guardar el documento PDF con elementos y encabezado configurado.
```

Los códigos generados en este apartado están disponibles en el repositorio de GitHub en https://github.com/narvydiego/openCVEGLPI.git. Los informes generados son de uso exclusivo del Laboratorio de Micro-Red de la Universidad de Cuenca y serán entregados a sus administradores.

3.2. Adaptación del *framework* Daniz-Red para *pentesting* en Micro-Redes

Daniz-Red es un *framework* de *pentesting* desarrollado en 2021 por Brian Quinde y Cinthya Campoverde. Este *framework* opera sobre un contenedor Docker utilizando la imagen de Kali-linux/Kali-Rolling, y está diseñado para implementar herramientas como The Harvester, Nuclei y Nmap, entre otras [8]. Está programado en Python y, hasta la fecha de este documento, se está trabajando en su segunda versión. Por esta razón, se decidió partir de la primera versión para desarrollar un *framework* de *pentesting* que sea adecuado para redes ICS.

3.2.1. Estructura del *pentesting*

Como se explicó en la Sección 2.4, un *pentesting* se puede dividir en varias fases. La fase de explotación sigue al reconocimiento de vulnerabilidades, pero, como se mencionó en la Sección 2.9, esta puede ser especialmente riesgosa en redes ICS. Por esta razón, se ha decidido estructurar el *pentesting* de manera que se minimicen los riesgos, como se muestra en la Figura 3.4. Cabe mencionar que el reporte final se elaborará a partir de los informes generados por el *framework*, el cual será de uso exclusivo del laboratorio.

Figura 3.4: Estructura de *pentesting* para el laboratorio de Micro-Red de la Universidad de Cuenca.

3.2.2. Arquitectura del *framework*

La arquitectura de Daniz-Red permite realizar pruebas de penetración que no afecten las operaciones de las micro-redes. Por esta razón, se propuso modificar la arquitectura de la versión original, manteniendo el objetivo de emular las fases de *red team* y *Penetration Testing Execution Standard* (PTES). En la Figura 3.5 se pueden visualizar los módulos implementados para cada una de las etapas. El módulo principal, denominado daniz-run.py, ejecuta el módulo pertinente según los requerimientos de cada usuario.

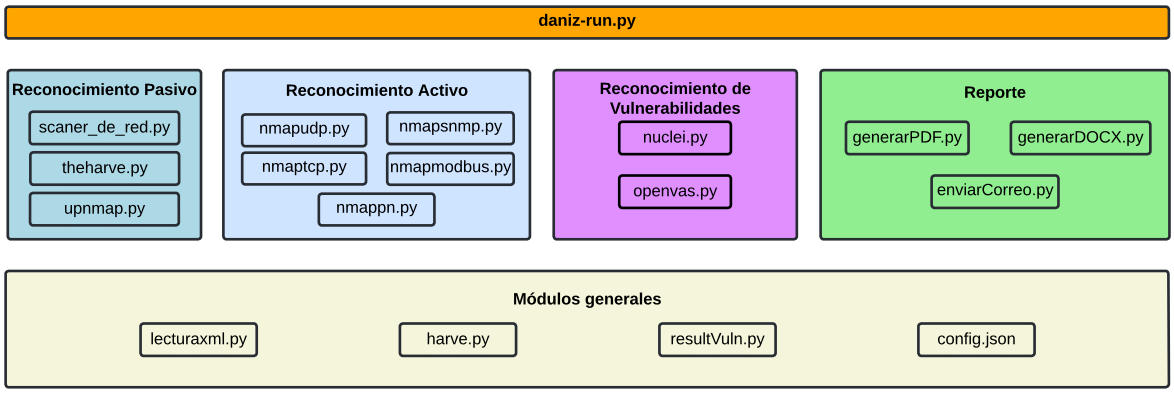

Figura 3.5: Arquitectura de Daniz-Red para versión SCADA

Es importante destacar que el *framework* desarrollado conserva la arquitectura original. En nuestra versión, hemos decidido agregar una nueva sección estructural que incluye directorios específicos para el almacenamiento de los módulos y los resultados obtenidos. En la Figura 3.6, se presenta un diagrama de flujo que ilustra la secuencia de operaciones del *framework*. Además, la primera versión incluía un módulo llamado crontab, utilizado para programar los tiempos de ejecución del *framework* dentro de Docker [8]. Para nuestra versión, es recomendable mantener este módulo desactivado, ya que el tiempo que dura el análisis de vulnerabilidades es variable y requiere una cantidad significativa de recursos computacionales.

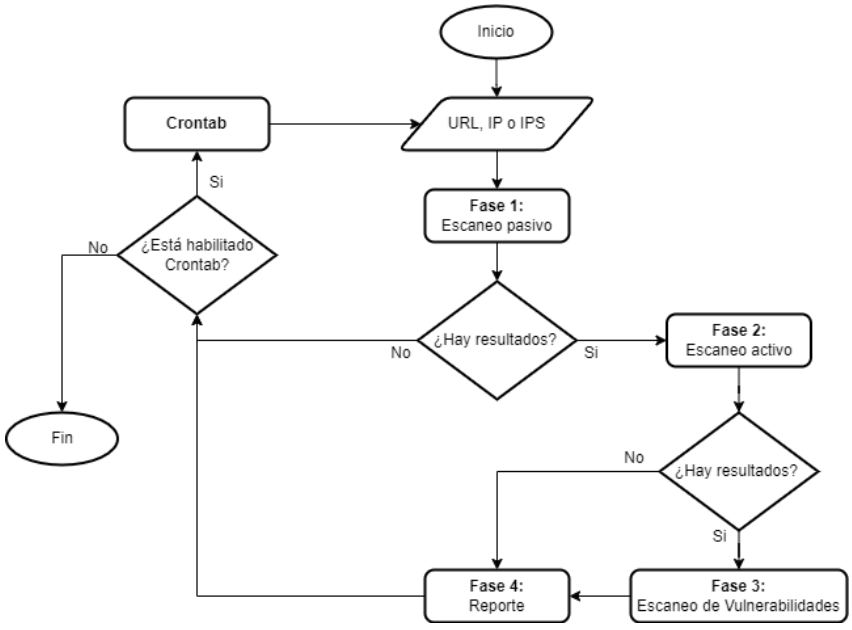

Figura 3.6: Diagrama de flujo del *framework* en su versión SCADA

En la versión SCADA de Daniz-Red, se ha optado por mantener el proceso de obtención de datos desde el archivo de configuración y el procesamiento de resultados en la fase 1, dado que no se han realizado cambios significativos en los módulos. Para un mayor entendimiento de estos procesos, se recomienda revisar la documentación proporcionada en [8].

Basado en los resultados de la fase 1, la finalidad de la fase 2 es recopilar puertos y servicios de cada uno de los objetivos. Dado que el presente *pentesting* está enfocado en una Micro-Red, se identificó la necesidad de adaptar la lectura de activos que operan con los protocolos ModBus y SNMP, que son comúnmente utilizados en la industria para la transmisión de datos y el estado de los dispositivos. Los resultados de estos escaneos se encuentran en formato XML; sin embargo, a diferencia de las lecturas UDP o TCP, estos archivos contienen una estructura diferente, razón por la cual se modificó la lógica, como se observa en el Algoritmo 5.

Algoritmo 5: Procesamiento de resultados de la fase 2

```
Entradas: Ruta de directorio con archivos de resultados de fase 2
Salida: Lista de direcciones y puertos
Leer y parsear el archivo XML a diccionario
Inicializar listas de direcciones y puertos
Si el archivo termina en "modbus.xml" o "snmp.xml" Entonces
    Para cada dirección en el diccionario hacer
        Si es una lista Entonces
            Guardar cada dirección en la lista de direcciones
        Sino
            Guardar la dirección directamente en la lista de direcciones
    Para cada puerto en el diccionario hacer
        Si el puerto está abierto Entonces
            Guardar protocolo y número de puerto en la lista de puertos
Sino
    Para cada dirección en el diccionario hacer
        Si es una lista o un diccionario Entonces
            Guardar todas las direcciones encontradas
    Para cada puerto en el diccionario hacer
        Si el puerto está abierto Entonces
            Guardar protocolo y número de puerto
Retornar puertos y direcciones
```

En la fase 3, implementamos dos herramientas, Nuclei y OpenVAS, cuyos resultados se guardan en archivos JSON y XML, respectivamente. Esto hizo necesario desarrollar un nuevo modelo que permita leer estos resultados, esenciales para la fase de generación de reportes, como se muestra en el Algoritmo 6.

Algoritmo 6: Procesamiento de resultados de la fase 3

```
Entradas: Ruta de directorio con archivos de resultados de fase 3
Salida: Lista de vulnerabilidades
Si la ruta del archivo termina en 'nuclei.json' Entonces
    Leer y parsear el archivo JSON a una estructura de diccionario
    Inicializar contador de vulnerabilidades a cero
    Para cada ítem en el archivo JSON hacer
        Obtener información de vulnerabilidad:
            ID de la plantilla, ruta de la plantilla, nombre, severidad, descripción, host, URL,
             CVE IDs
            Filtrar y transformar CVE IDs si no están en formato de lista
        Si la severidad es 'low', 'medium' o 'high' Entonces
            Imprimir detalles formatados de la vulnerabilidad
            Incrementar contador de vulnerabilidades
    Imprimir el total de vulnerabilidades encontradas en formato JSON
Sino si la ruta del archivo termina en 'openvas.xml' Entonces
    Leer y parsear el archivo XML a una estructura de árbol XML
    Inicializar contador de vulnerabilidades a cero
    Para cada resultado en el archivo XML hacer
        Obtener información de vulnerabilidad:
            Nombre, amenaza, descripción
            Buscar y extraer IDs de CVE relacionados sólo si están disponibles
        Si la amenaza es 'Low', 'Medium' o 'High' Entonces
            Imprimir detalles formatados de la vulnerabilidad, incluyendo todos los CVEs
            Incrementar contador de vulnerabilidades
    Imprimir el total de vulnerabilidades encontradas en formato XML
Sino
    Imprimir "Formato de archivo no soportado."
```

El siguiente y último paso es la generación de informes basados en los resultados obtenidos en las fases anteriores. Para ello, se propuso crear dos tipos de módulos: uno en formato PDF, utilizando la librería ReportLab, y otro en formato DOCX, usando la librería python-docx,

que puede ser modificado según las necesidades del usuario. La lógica para generar estos archivos es la misma y se detalla en el Algoritmo 7. La secuencia para el procesamiento de datos se ha basado en las fases anteriores.

Algoritmo 7: Generación de informes de resultados de *pentesting*

Entradas: Ruta de la carpeta de resultados
Salida: Genera informes en PDF y DOCX

```
Obtener argumentos de línea de comandos
Definir rutas de archivos de salida basadas en la fecha actual
Para cada tipo de archivo en los resultados hacer
    Inicializar documento PDF y estilo de documento
    Leer y analizar datos de archivo (JSON o XML según el caso)
    Para cada entrada de vulnerabilidad encontrada hacer
        Aplicar formatos y agregar información de vulnerabilidad al documento
        Si es archivo JSON Entonces
            Extraer y formatear datos específicos como CVE IDs, severidad, etc.
        Sino si es archivo XML Entonces
            Extraer y formatear datos, manejar referencias y descripciones de amenazas
    Guardar y cerrar el documento
Imprimir ruta del archivo de reporte generado
```

Debido a que el código de programación del *framework* es extenso, no se incluirá en su totalidad en este trabajo de titulación. Sin embargo, está disponible para descarga en su repositorio de GitHub en la rama SCADA. Se puede acceder a través del enlace https://github.com/fabianastudillo/daniz-red.git.

3.2.3. Integración de OpenVas

Durante la fase de análisis de vulnerabilidades, se observó que Nuclei no identificaba suficientes vulnerabilidades ni proporcionaba un análisis adecuado para redes industriales ICS. Por esta razón, se decidió implementar OpenVAS, un escáner que ofrece una amplia gama de *Network Vulnerability Tests* (NVT) adecuados para dispositivos como *Programmable Logic Controller* (PLC) y Remote Terminal Unit (RTU), así como para protocolos como Modbus y DNP3. Esta herramienta se desplegó en contenedores; por lo tanto, se agregaron al Dockerfile de Daniz-Red los servicios necesarios para asegurar el correcto funcionamiento de OpenVAS.

Para integrar la herramienta al framework, es necesario implementar o exponer el socket Unix de gvmd en el contenedor donde se ejecuta Daniz-Red. Es por eso que, desde el archivo docker

-compose, enlazamos los directorios para los servicios de OpenVAS gvmd, gsa y el contenedor con Daniz-Red, como se muestra en la Figura 3.7. Además, para un control desde Python, son necesarias las bibliotecas python-gvm y gvm-tools.

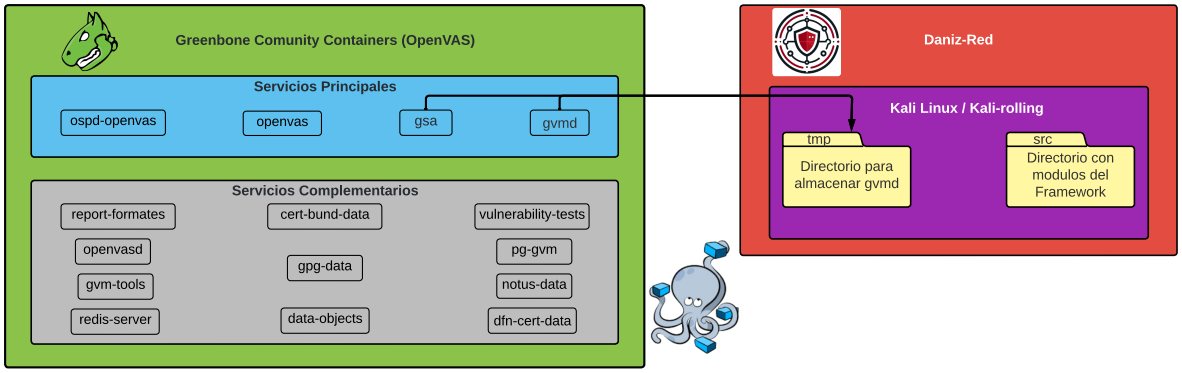

Figura 3.7: Esquema de integración de contenedores en Daniz-Red

Una vez instaladas y configuradas las dependencias, se desarrolló el módulo para el manejo de OpenVAS, cuya secuencia se muestra en la Figura 3.8. Los resultados de estos análisis se devuelven en un archivo en formato XML.

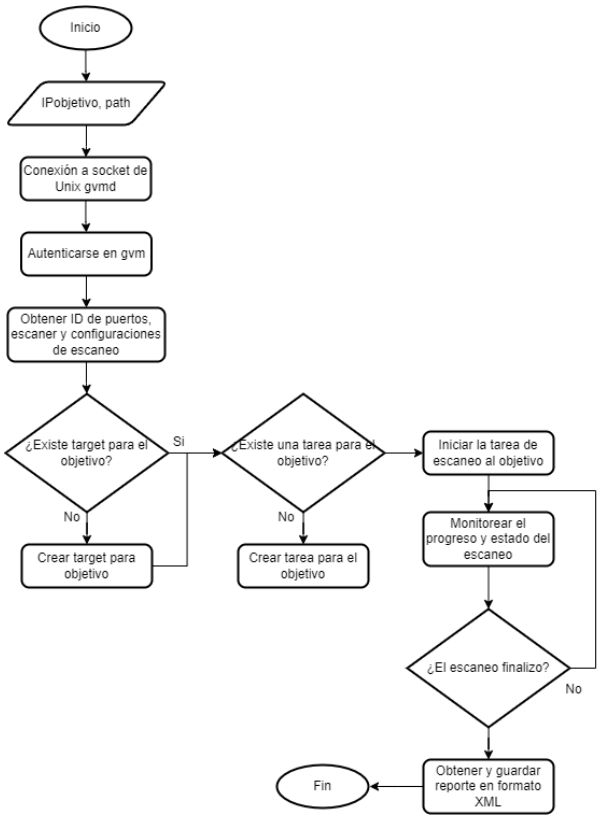

Figura 3.8: Diagrama de flujo para manejo de OpenVAS desde Daniz-Red

3.2.4. Implementación del *framework* en el laboratorio de Micro-Red

Daniz-Red es un *framework* desarrollado dentro de un contenedor Docker. Para su implementación, se recomienda utilizar sistemas operativos Linux, debido a la facilidad de instalación de Docker y Docker-compose. Por esta razón, su implementación se llevó a cabo en una máquina virtual dentro de uno de los ordenadores del laboratorio. Los pasos de instalación se pueden consultar en la Sección 5.2.

3.3. Evaluación de Vulnerabilidades y Explotación

Tras la implementación de la fase de *pentesting*, se obtuvo un panorama detallado de las vulnerabilidades existentes en la Micro-Red. Esto permitió proceder con una evaluación exhaustiva y la subsiguiente clasificación de dichas vulnerabilidades. Una vez evaluadas y categorizadas, se realizaron explotaciones controladas sobre activos específicos, con el objetivo de validar y fortalecer las medidas de seguridad implementadas.

3.3.1. Tipos de vulnerabilidades y filtrado de activos

En esta sección, se presentan las vulnerabilidades detectadas en la red, clasificadas según su nivel de criticidad: alto, medio y bajo riesgo. Estas categorías permiten priorizar las acciones correctivas según el impacto potencial en la seguridad de la red. Cabe destacar que las vulnerabilidades enumeradas en cada subsección representan solo una muestra de las encontradas durante el proceso de auditoría.

3.3.1.1. Vulnerabilidades de alto riesgo

Las vulnerabilidades de alto riesgo son aquellas que pueden permitir a un atacante comprometer la red a nivel de sistema operativo o de aplicaciones de manera significativa. Estas vulnerabilidades se identificaron en diversos componentes, incluidos software, hardware y servicios web.

- ***Apache Guacamole Default Credentials* (HTTP):** Vulnerabilidades por uso de credenciales por defecto, permitiendo fácil acceso no autorizado.
- ***Moxa NPort Unprotected Web Console*:** Consolas web de dispositivos Moxa accesibles sin autenticación, permitiendo control no autorizado sobre los dispositivos.

- ***MikroTik RouterOS Remote Code Execution*:** Vulnerabilidades que permiten ejecución remota de código en versiones específicas de RouterOS, con riesgo crítico de compromiso del sistema.

3.3.1.2. Vulnerabilidades de medio riesgo

Las vulnerabilidades de medio riesgo, aunque menos críticas, presentan riesgos significativos para la integridad y disponibilidad de la red, incluyendo problemas de configuración y el uso de protocolos de seguridad desactualizados.

- **SSL/TLS *Deprecated Protocol Detection*:** Detección de protocolos SSL y TLS obsoletos que no proporcionan una seguridad adecuada contra las intercepciones modernas.
- ***Anonymous FTP Login Reporting*:** Configuraciones de servidor FTP que permiten logins anónimos, potencialmente exponiendo datos sensibles.
- ***Weak Cipher Suites for HTTPS*:** Uso de conjuntos de cifrado débiles en configuraciones *Hypertext Transfer Protocol Secure* (HTTPS) que no garantizan una comunicación segura.

3.3.1.3. Vulnerabilidades de bajo riesgo

Las vulnerabilidades de bajo riesgo son aquellas que, aunque no permiten un compromiso inmediato de la red, podrían ser explotadas en conjunto con otras vulnerabilidades para ampliar un ataque.

- ***ICMP Timestamp Reply Information Disclosure*:** Respuestas ICMP que pueden revelar información sensible sobre la configuración de la red.
- ***TCP Timestamps Information Disclosure*:** Vulnerabilidad que permite a un atacante deducir tiempos de actividad del sistema y otras configuraciones mediante el análisis de los *timestamps* de TCP.
- ***Missing 'HttpOnly' Cookie Attribute*:** Cookies sin el atributo 'HttpOnly', lo que puede permitir a *scripts* del lado del cliente acceder a la información de las cookies, aumentando el riesgo de ataques XSS.

3.3.1.4. Filtrado de activos

Para asegurar la continuidad operativa y la estabilidad de la red, se identificaron y seleccionaron para corrección aquellos dispositivos y sistemas que exhibían las vulnerabilidades más severas, cuidando especialmente que las modificaciones propuestas no comprometieran la integridad general de la Micro-Red. Los dispositivos escogidos incluyen los *switches Weidmuller* IE-SW-VL08MT-5TX-1SC-2SCS y el *router* MikroTik RouterOS. En cuanto a los sistemas, se priorizaron los servidores Windows Server 2012 R2 y Ubuntu 22.04, así como el servicio Apache Guacamole. La elección de estos elementos se basó en el principio de minimizar el impacto operacional mientras se optimizaba la mitigación de amenazas.

3.3.2. Explotación de vulnerabilidades detectadas

Siguiendo la identificación y clasificación de las vulnerabilidades, se procedió a realizar explotaciones sobre los activos filtrados en la Sección 3.3.1.4. Estas pruebas permitieron confirmar la naturaleza y la severidad de las amenazas, así como validar la eficacia de las estrategias de mitigación propuestas.

3.3.2.1. Apache Guacamole

Se efectuó un ataque utilizando las credenciales predeterminadas encontradas durante la fase de *pentesting*. Esta explotación confirmó la vulnerabilidad de autenticación del servicio Apache Guacamole, lo que permitió un acceso no autorizado a través del puerto 8443. Este paso permitió entender la profundidad del acceso que podría obtenerse mediante estas credenciales.

En la Figura 3.9 se muestra cómo se accedió al control del sistema SCADA a través de Apache Guacamole utilizando las credenciales por defecto. Este acceso directo demostró la susceptibilidad del sistema a intrusiones externas, resaltando la necesidad crítica de fortificar la seguridad.

Para una revisión detallada del proceso completo de explotación y las etapas de intervención, véase la Sección 5.2 del anexo.

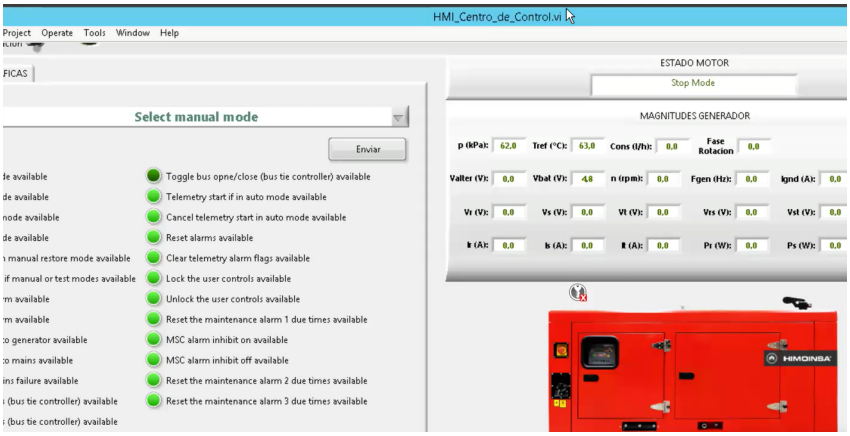

Figura 3.9: Acceso al sistema SCADA mediante Apache Guacamole.

3.3.2.2. Moxa NPort *Web Console*

De manera similar a la intervención de la Sección 3.3.2.1, se llevó a cabo una explotación controlada de la consola web Moxa NPort para demostrar su accesibilidad sin protección, lo que subrayó la urgencia de implementar medidas correctivas. Esta explotación implicó el acceso directo a la interfaz web del dispositivo Moxa mediante un navegador web estándar. Se constató que el acceso a la interfaz era completamente directo y no requería ninguna forma de autenticación previa, lo que pone en evidencia una grave vulnerabilidad de seguridad. Este procedimiento está documentado visualmente en la Figura 3.10.

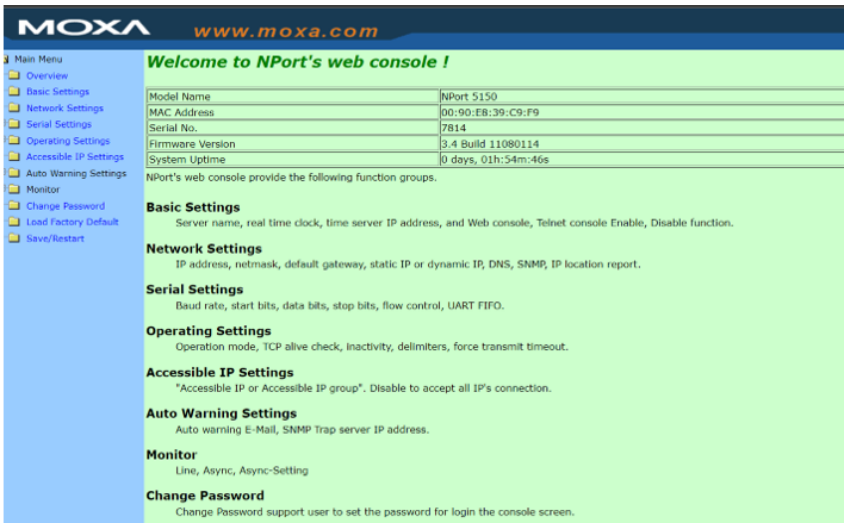

Figura 3.10: Ingreso a la interfaz de Moxa sin previa autenticación.

Estas intervenciones demostraron la posibilidad de controlar estos dispositivos de manera remota sin restricciones, destacando la urgencia de aplicar medidas de seguridad adecuadas para proteger estos puntos de entrada críticos.

3.4. Corrección de vulnerabilidades

En respuesta a las vulnerabilidades identificadas y sus respectivas explotaciones, se implementaron correcciones sobre los dispositivos y sistemas previamente seleccionados, con el objetivo de fortalecer la seguridad de la Micro-Red.

3.4.1. Generación y cambio de contraseña en Apache Guacamole

Para corregir el problema de credenciales por defecto, se realizó el proceso de cambio de contraseña del servicio Apache Guacamole, haciendo uso de la herramienta *KeePass 2*, un gestor de contraseñas robusto y seguro, para generar una nueva contraseña de 20 caracteres, como se observa en la Figura 3.11. Esta nueva contraseña fue diseñada para ser altamente segura, combinando letras mayúsculas, minúsculas y números, lo que aumenta significativamente la resistencia contra ataques de fuerza bruta y adivinación.

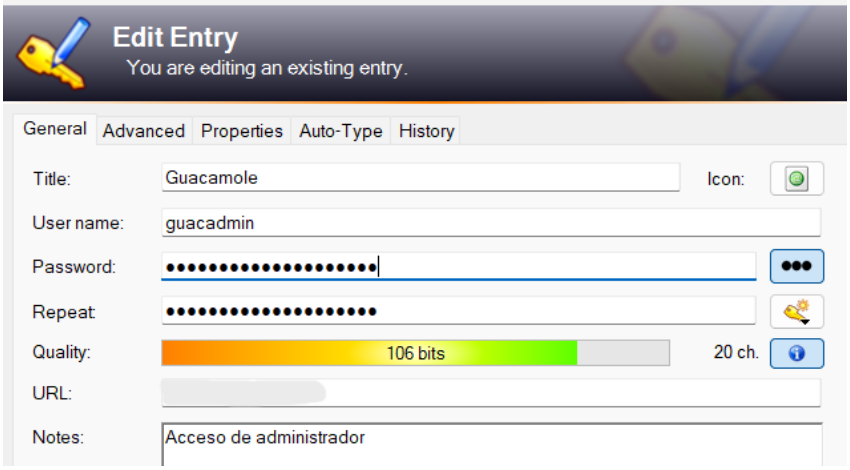

Figura 3.11: Contraseña de 20 caracteres generada por keePass 2.

Detalles adicionales sobre el proceso de cambio de contraseña y las configuraciones de *KeePass 2* se pueden encontrar en el Anexo 5.2, donde se documentan los pasos específicos y las capturas de pantalla del proceso.

3.4.2. Actualización de *firmware* en MikroTik RouterOS

Para la mitigación de vulnerabilidad encontrada en este dispositivo se procedió con la actualización del *firmware* del mismo, pasando de la versión 6.49 a la versión 6.49.15, tal y como se

evidencia en la Figura 3.12. Más detalles de cómo se realizó la actualización se encuentran en el Anexo 5.2.

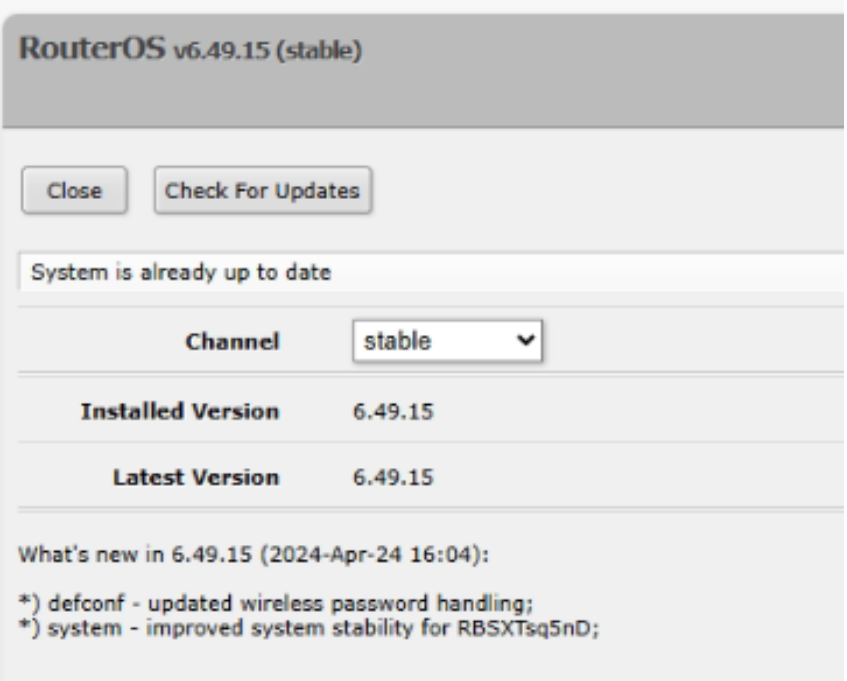

Figura 3.12: MikroTik RouterOS actualizado a su ultima versión.

3.4.3. Habilitación del Protocolo SNMP v3 en *Switches Weidmuller*

A pesar de que no se identificaron vulnerabilidades relacionadas con el protocolo *Simple Network Management Protocol* (SNMP) durante la fase de *pentesting*, una revisión minuciosa de la configuración de los activos reveló que los *switches Weidmuller* operaban utilizando la versión 1 de SNMP, la cual está habilitada por defecto. Esta versión del protocolo es conocida por sus limitaciones significativas en términos de seguridad, incluyendo la falta de cifrado y autenticación adecuada, lo que podría exponer la red a riesgos de interceptación y manipulación de datos.

La versión 3 de SNMP, a diferencia de su predecesora, ofrece mejoras críticas en seguridad, incluyendo autenticación, cifrado y una gestión más robusta de accesos [61]. La implementación de SNMP v3 en estos dispositivos permite asegurar la integridad y la confidencialidad de la información de gestión de la red transmitida entre los dispositivos y los sistemas de administración de red.

La habilitación de SNMP v3 se llevó a cabo con especial cuidado para asegurar que la transición desde la versión 1 no afectara negativamente el funcionamiento de la Micro-Red. La

activación del protocolo se evidencia en la Figura 3.13.

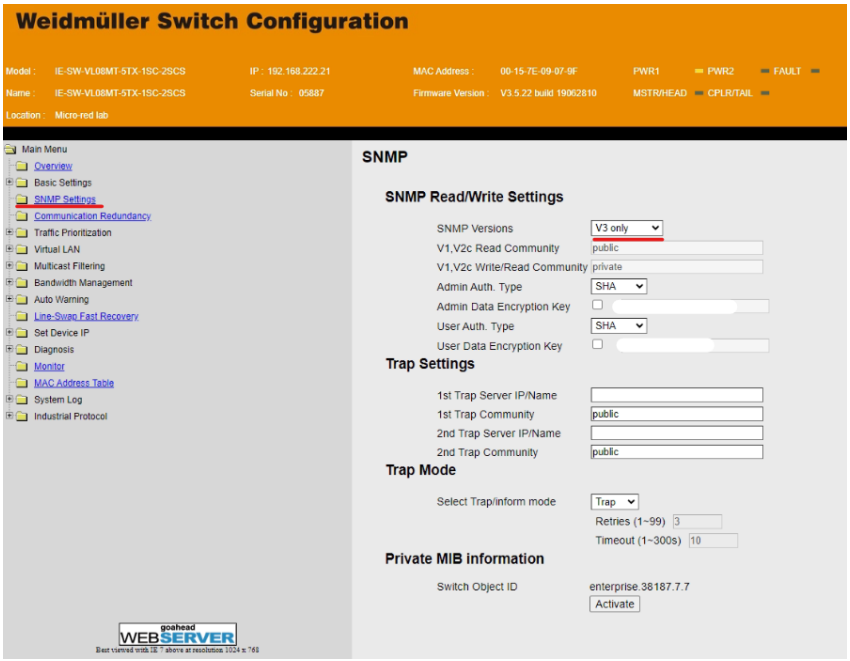

Figura 3.13: Protocolo SNMP v3 habilitado.

3.4.4. *Hardening* en Servidores. Fase: Adaptación de Scripts

Para llevar a cabo el *hardening* de los servidores, se utilizaron *scripts* proporcionados por CE-DIA, los cuales son de uso confidencial y cuya información no se puede divulgar. Estos *scripts* fueron modificados específicamente para adaptarse a las necesidades y requisitos únicos de la Micro-Red. Con el fin de garantizar la seguridad y la eficacia de las adaptaciones sin comprometer la operatividad de los sistemas de la red, se creó un ambiente de pruebas utilizando máquinas virtuales. Este enfoque permitió evaluar y solucionar cualquier inconveniente en un entorno controlado, asegurando que las acciones ejecutadas en los servidores reales no afectaran las funciones críticas de la Micro-Red.

3.4.4.1. Servidor Ubuntu 22.04 LTS

Para el servidor Ubuntu 22.04 LTS, inicialmente, se probó el *script* proporcionado para evaluar su funcionalidad y el impacto sobre el sistema. Durante estas pruebas, se identificaron diversos controles dentro del *script* (similares a los de la Tabla 2.2 de la Sección 2.6), que incluyen acciones como la actualización de políticas de usuario, ajustes en la gestión de permisos y la optimización de los servicios activos. Tras la evaluación inicial, se eliminó el control *Center for*

Internet Security (CIS) 3.2.1 porque su activación generó errores de conectividad que impedían el acceso a los servicios Apache Guacamole y GLPI a través de navegadores, mostrando mensajes de error por falta de conexión con el servidor.

Posteriormente, se realizaron modificaciones en los *scripts* para incorporar un mecanismo de auditoría y remediación integrados, en el cual cada acción identificada por la auditoría era seguida de una sugerencia de remediación inmediata. Esta metodología se representada en el Algoritmo 8, donde se muestra la lógica del *script*.

Además, cabe destacar que se añadió una funcionalidad interactiva al *script* que solicita la confirmación del usuario antes de ejecutar cada control de seguridad. Esta característica permite un mayor control y personalización en la aplicación del *hardening*, ofreciendo a los administradores la opción de omitir controles específicos que pudieran impactar aplicaciones críticas o comprometer la estabilidad del sistema.

3.4.4.2. Servidor Windows

El primer paso para el servidor Windows Server 2012 R2, al igual que con el servidor Ubuntu, consistió en evaluar el *script* de auditoría y remediación para asegurar su alineación con las normativas internas y las mejores prácticas de seguridad específicas de la red de la universidad. Durante la evaluación, se identificaron algunos controles que fueron eliminados por no ser compatibles con la versión del servidor.

Además, se implementaron cambios y mejoras al *script* de auditoría y remediación. Este proceso también incorporó una funcionalidad interactiva que solicitaba confirmación del usuario antes de ejecutar controles de seguridad, permitiendo a los administradores adaptar el *hardening* según las necesidades operativas y evitar impactos en funciones críticas del sistema. La lógica detrás de este procedimiento está detallada en el Algoritmo 8.

3.4.5. *Hardening* en Servidores. Fase: Aplicación de *scripts*

Tras completar satisfactoriamente las fases de adaptación, se procedió a la implementación de los *scripts* modificados en los servidores reales de la Micro-Red. Esta fase traslada las mejoras de seguridad verificadas en el entorno controlado a la infraestructura operativa de la organización.

Los *scripts* de auditoría y remediación, personalizados y probados, fueron ejecutados en sus

Algoritmo 8: Auditoría y remediación de controles

```
Entradas: Registros del sistema con estados esperados
Salida: Archivo de registro con los estados finales de cada configuración
Para cada registro en registros del sistema hacer
    Comparar el estado actual del registro con el estado esperado;
    Si el estado actual es igual al esperado Entonces
        Imprimir y registrar el resultado "OK";
        Continuar con el siguiente registro;
    Sino
        Preguntar al usuario si desea aplicar la remediación;
        Si la respuesta es sí Entonces
            Aplicar la remediación;
            Imprimir y registrar el resultado "OK";
            Continuar con el siguiente registro;
        Sino
            Imprimir y registrar el resultado "FAIL";
            Continuar con el siguiente registro;
Guardar los resultados en un archivo de registro;
```

respectivos servidores Windows Server 2012 R2 y Ubuntu 22.04. Cada *script* se ejecutó siguiendo un protocolo estricto para minimizar interrupciones y garantizar la integridad del sistema durante el proceso.

Durante la ejecución de los *scripts*, se tomaron capturas de pantalla que documentan cada paso del proceso, evidenciadas en las Figuras 3.14 y 3.15. Estas imágenes proporcionan una prueba visual clara del *hardening* en acción para verificar que los *scripts* se aplicaron correctamente en el entorno de producción. Además, las capturas ayudan a ilustrar la implementación efectiva y la funcionalidad interactiva de los *scripts*, asegurando que todos los procedimientos se siguieron como se planificó.

Con la aplicación de los *scripts* en los servidores en producción, se concluye la fase de implementación del *hardening*. La documentación generada y los datos recopilados en los archivos *Comma-Separated Value* (CSV) serán analizados en la siguiente capítulo, dedicado a los resultados.

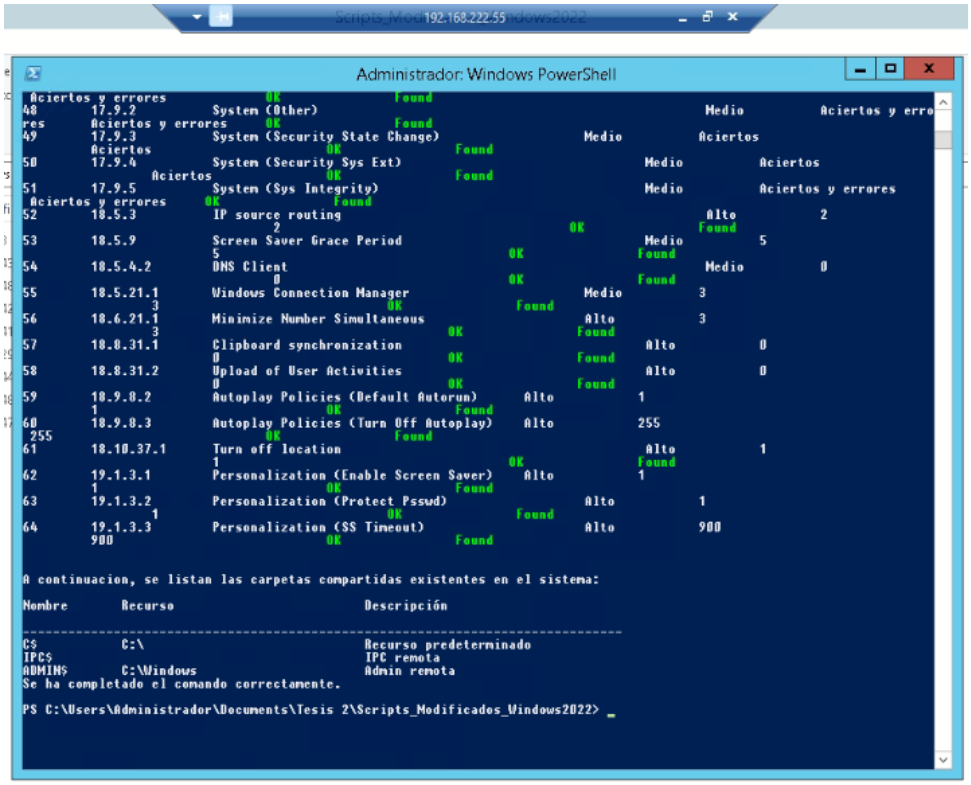

Figura 3.14: Aplicación del *script* auditoría en el servidor Windows de la Micro-Red.

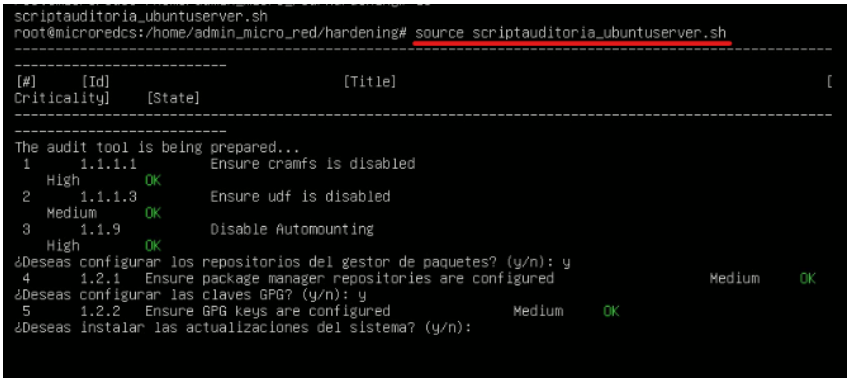

Figura 3.15: Aplicación del *script* Auditoría en el servidor Ubuntu de la Micro-Red.

4. Resultados

En este capítulo se presentan las mejoras operativas y de seguridad obtenidas al implementar el *framework* Daniz-Red y la gestión efectiva de activos. En la Sección 4.1, se muestra cómo la implementación de *Gestionnaire Libre de Parc Informatique* (GLPI), junto con sus extensiones, ha mejorado la optimización de los activos del laboratorio. Mientras tanto, en la Sección 4.2 se evidencian las vulnerabilidades identificadas en el laboratorio tras la implementación del *framework*. Finalmente, en la Sección 4.3 se discuten los resultados obtenidos a partir de las correcciones implementadas después del proceso de *hardening*.

4.1. Evaluación de la gestión de activos

Esta sección detalla cómo la implementación de GLPI, junto con sus extensiones y agentes, permite realizar una gestión de activos de manera más eficiente y automatizada. Además, se discuten las vulnerabilidades encontradas al integrar la *Application Programming Interface* (API) de GLPI con la de OpenCVE.

4.1.1. Inventario de activos

Categoría	**Numero de activos inventariados**	
	Sin *Inventory* y Agente	**Con *Inventory* y Agente**
Ordenadores	6	12
Dispositivos de Red	18	51
Perifericos	0	2
Monitores	0	3
Software	0	3000

Tabla 4.1: Activos inventariados con GLPI.

La implementación de la extensión *Inventory* en el contenedor Docker del servidor Ubuntu, como se detalla en el Anexo 5.2, junto con la activación de los agentes de GLPI descrita en el Anexo 5.2, ha facilitado la recolección de información detallada de los activos. Esto incluye nombre, tipo de equipo, marca, modelo, responsable, interfaz de red, y comentarios específicos

sobre cada activo. Un resumen del número de activos inventariados por categoría se puede observar en la Tabla 4.1.

La extensión y los agentes han posibilitado la obtención de datos más precisos sobre los activos del laboratorio, logrando un inventario completo del software presente en cada ordenador. Además, gracias al descubrimiento de la red, se han identificado los dispositivos disponibles en cada una de las subredes del laboratorio, lo que facilita el registro de los activos Tecnologías de Operación (TO) de la categoría de dispositivos de red, como se muestra en la Figura 4.1.

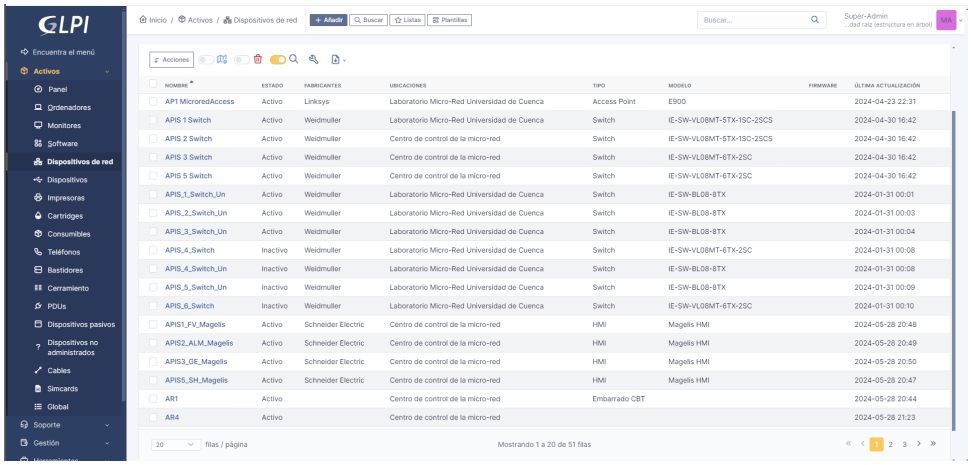

Figura 4.1: Inventario de activos OT en GLPI.

Cabe recalcar que la información de los activos descubiertos ha sido complementada con datos recopilados durante el proceso de levantamiento de información, conforme a lo reportado por [60]. Además, la categoría de software abarca todos los programas instalados en los cuatro activos donde se han desplegado los agentes, incluyendo visualizadores y el servidor Ubuntu. En la Figura 4.2 se puede ver el *dashboard* de GLPI con el resumen de activos almacenados.

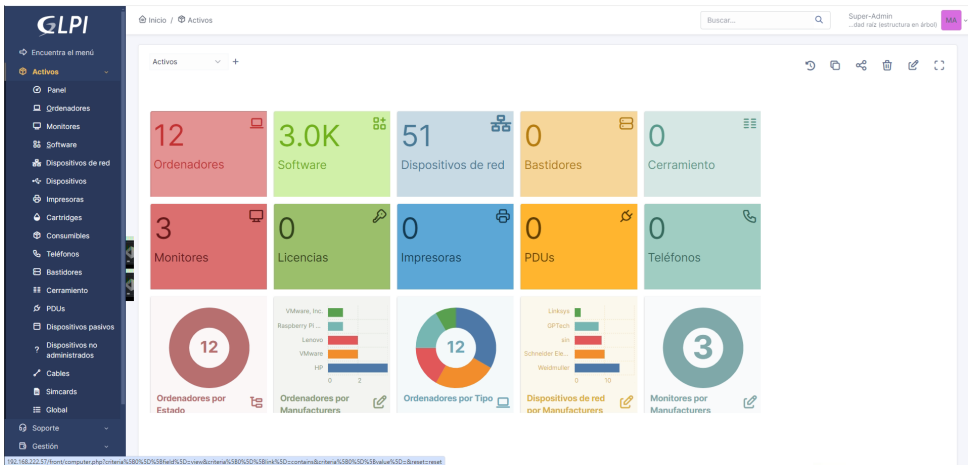

Figura 4.2: Dashboard de activos en GLPI.

4.1.2. Obtención de CVE mediante OpenCVE y GLPI

Una vez registrados todos los activos dentro del servicio GLPI, se implementó un programa para identificar *Common Vulnerabilities and Exposures* (CVE) en cada uno de los dispositivos del laboratorio. Como se explicó en la Sección 3.1, los activos de las categorías de computadores, dispositivos de red y periféricos fueron catalogados. Esto proporciona al administrador una herramienta útil para identificar posibles fallos de seguridad en los activos Tecnologías de la Información (TI) y TO del laboratorio.

Los resultados del análisis se compilan en un archivo PDF que será entregado al administrador del laboratorio. Como se muestra en la Figura 4.3, el análisis revela varias vulnerabilidades en distintos modelos de dispositivos. En la Sección 5.2, se pueden observar con mayor detalle las vulnerabilidades encontradas y los dispositivos específicos del laboratorio que se ven afectados.

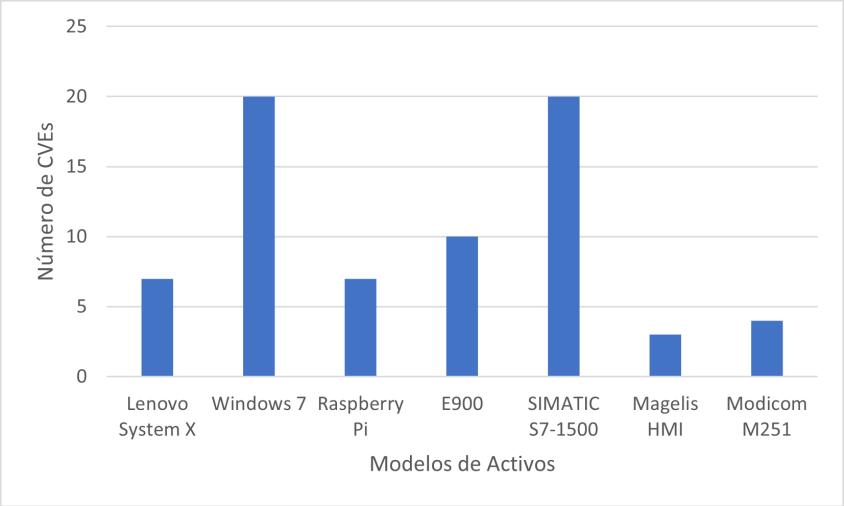

Figura 4.3: CVE encontrado en base información ingresada en GLPI.

4.2. Análisis del reporte de *pentesting* realizado en el laboratorio

El *pentesting* se llevó a cabo con ayuda del *framework* durante 72 horas desde un *host* ubicado dentro de la red interna del laboratorio. Se analizaron las dos subredes pertenecientes a la Micro-Red, las cuales se listan a continuación:

- 192.168.222.0/24
- 192.168.244.0/24

En esta sección, se mostrarán los resultados obtenidos en cada una de las fases del *pentesting*, descrito en 3.2.

4.2.1. Resultados de la fase 1: Reconocimiento pasivo

En la primera fase, se realizó un escaneo de los *hosts* que operan dentro del laboratorio. Se identificaron 54 activos disponibles, entre ellos inversores fotovoltaicos, *Programmable Logic Controller*s (PLCs), *switches* industriales, servidores y ordenadores. Además, se llevó a cabo un análisis de los dominios de la universidad en busca de algún dominio que perteneciera al laboratorio; sin embargo, no se encontró ninguno que pertenezca a la Micro-Red, como se muestra en la Figura 4.4.

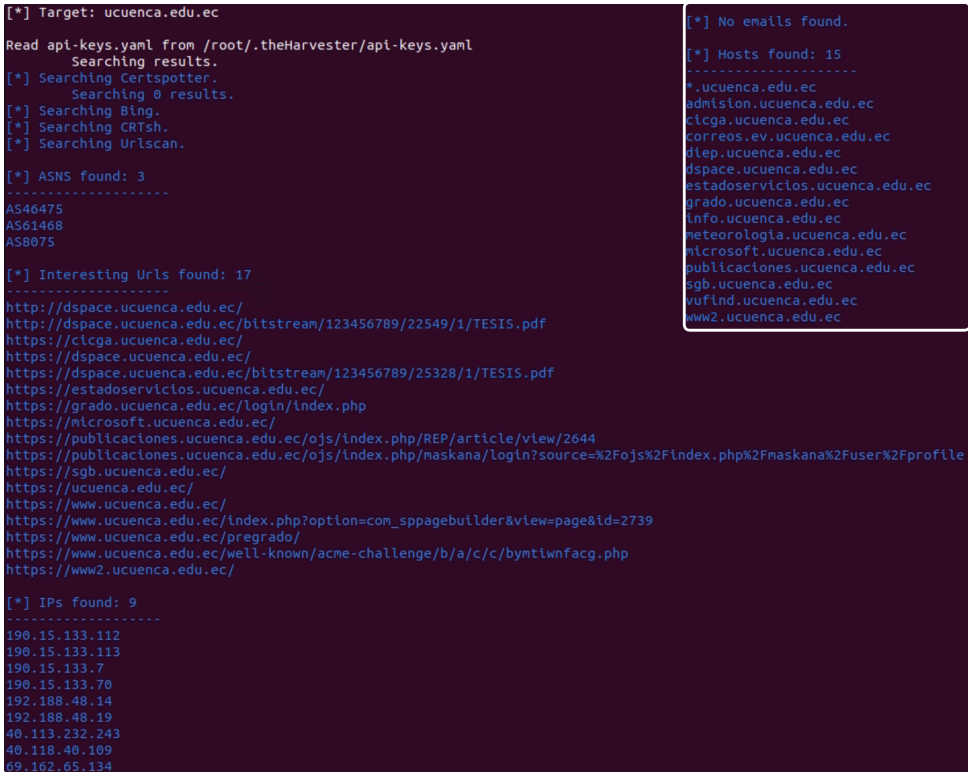

Figura 4.4: Resultados de The Harvester al buscar ”ucuenca.edu.ec”.

4.2.2. Resultados de la fase 2: Reconocimiento activo

Una vez identificados los objetivos del laboratorio, se procedió al reconocimiento activo, que resultó en el levantamiento de 409 servicios. De estos, 348 pertenecen a la subred 192.168.222.0/24 y 61 a la 192.168.244.0/24.

En la Figura 4.5 se puede ver un resumen de los servicios más importantes encontrados. Los principales incluyen: 42 activos operando con el protocolo Modbus, 80 con *Hypertext Transfer Protocol* (HTTP), 28 con *Simple Network Management Protocol* (SNMP), 23 con DHCP y 20 con Telnet, entre otros. Un detalle más exhaustivo se proporciona en los informes preliminares que se entregarán a los administradores del laboratorio.

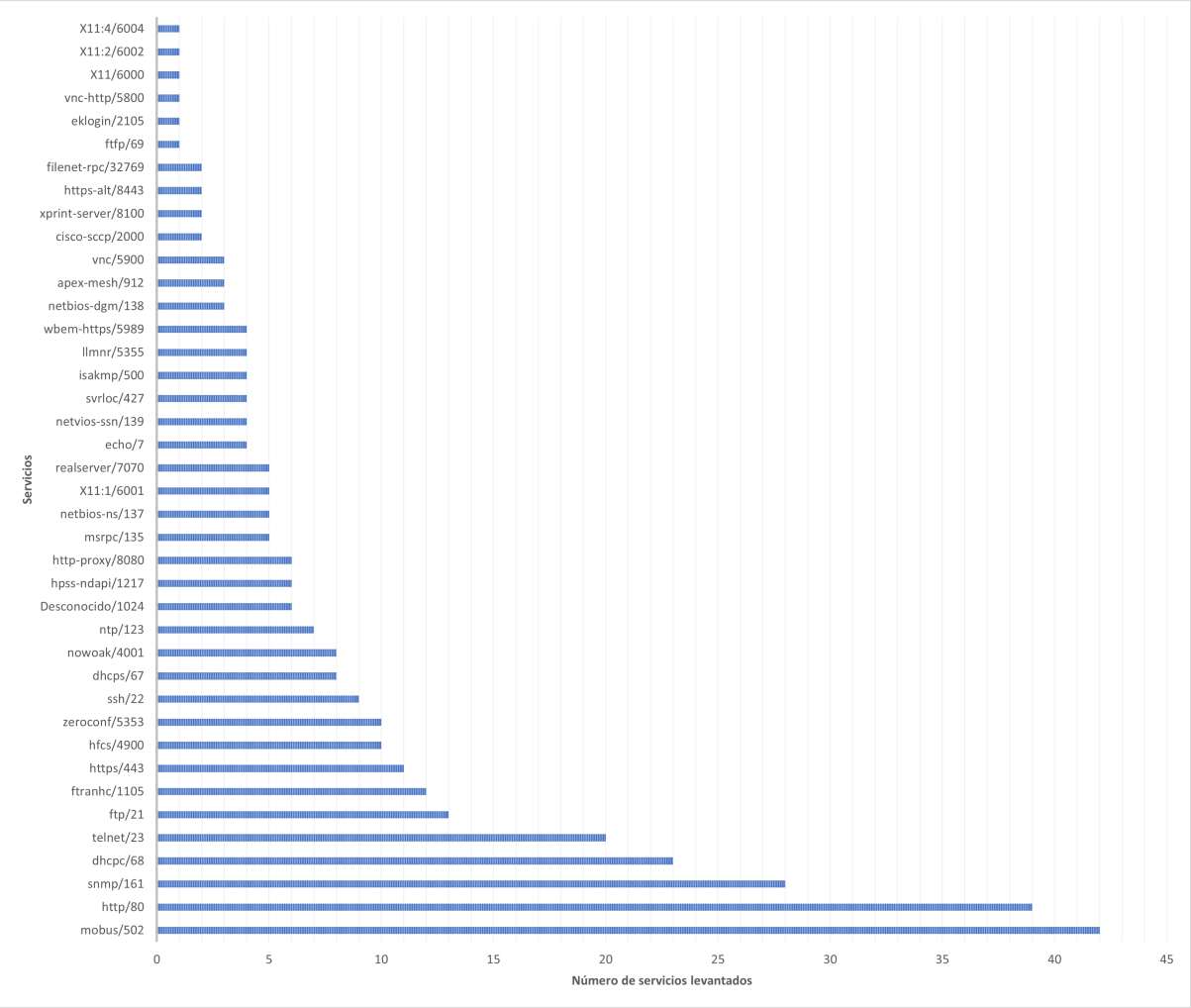

Figura 4.5: Resumen de servicios levantados en los activos de la Micro-Red.

4.2.3. Resultados de la fase 3: Reconocimiento de vulnerabilidades

Con la finalización de la fase de análisis de vulnerabilidades, se pueden obtener los informes preliminares que genera el *framework* Daniz-Red. Basándonos en estos resultados, podemos observar en la Figura 4.6 una visión general de las vulnerabilidades encontradas en el laboratorio de Micro-Red de la Universidad de Cuenca, clasificadas por su nivel de severidad.

Basándonos en los resultados, se elaboró un informe técnico de *pentesting* que detalla las vulnerabilidades encontradas, sus *exploits* y los posibles métodos de remediación. A continuación, se presenta un resumen de las vulnerabilidades identificadas. Para obtener más detalles, se recomienda solicitar este informe a los administradores del laboratorio.

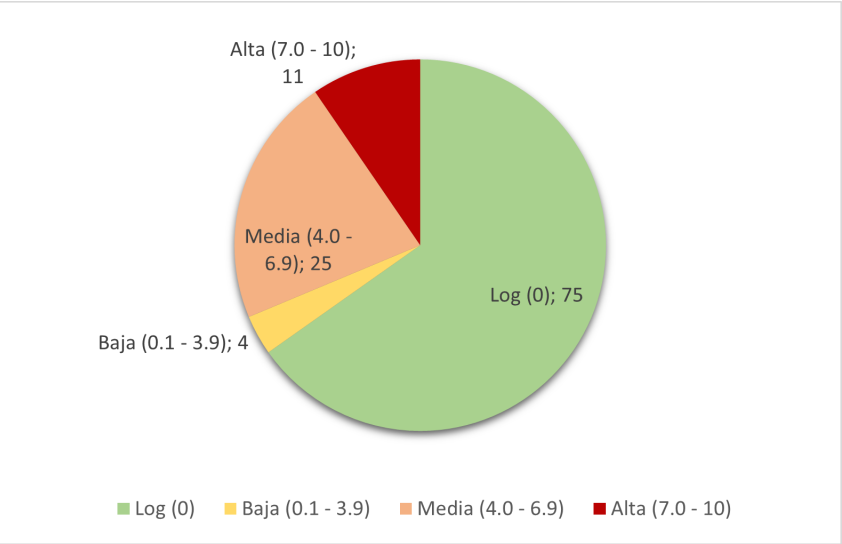

Figura 4.6: Vulnerabilidades encontradas en el laboratorio de Micro-Red.

4.2.3.1. Vulnerabilidades con severidad alta

En la Figura 4.7 se muestran a detalle las vulnerabilidades encontradas, basadas en el número de activos en los que se presentan.

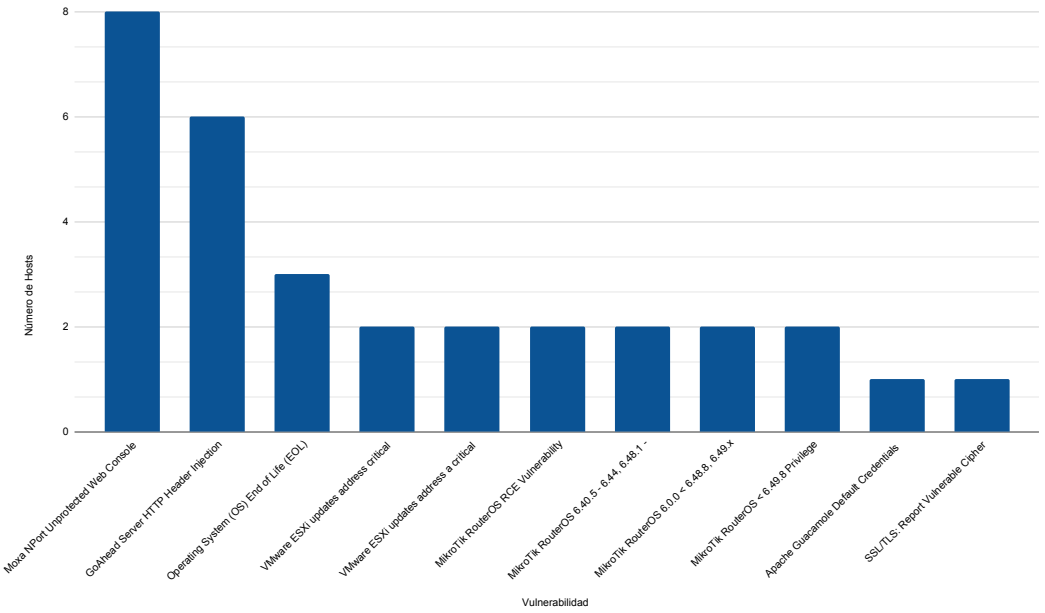

Figura 4.7: Número y tipo de vulnerabilidades de nivel alto presentes en la Micro-Red.

Estas amenazas están distribuidas en 20 activos diferentes del laboratorio. A continuación, se detalla una explicación de cuatro vulnerabilidades que se encuentran en el mayor número de dispositivos.

- ***Moxa NPort Unprotected Web Console:***

  Su severidad alcanza los 10 puntos y está presente en diversos activos de TO, como inversores energéticos y fuentes de alimentación, entre otros. Esta vulnerabilidad puede permitir a un atacante acceder y modificar las configuraciones de los dispositivos. Por esta razón, Moxa ha recomendado deshabilitar puertos y aplicar reglas de *firewall* que solo permitan el acceso a dispositivos de confianza. Además, en caso de necesitar acceso remoto, se recomienda utilizar métodos seguros como *Virtual private network*s (VPNs).

- ***GoAhead Server HTTP Header Injection Vulnerability:***

  La vulnerabilidad tiene una severidad de 8.6 puntos y afecta a 6 *hosts*. Esta vulnerabilidad permite lanzar ataques de *phishing* efectivos, engañando a los usuarios para que revelen información sensible al visitar un sitio web que parece legítimo. Para remediarla, es necesario actualizar a la última versión del servidor GoAhead.

- **Vulnerabilidades relacionados con VMWare ESXi:**

  El servicio VMware ESXi es un software que permite crear y ejecutar máquinas virtuales

en un servidor físico. Estas vulnerabilidades afectan a 2 hosts del laboratorio y tienen un promedio de severidad de 9.2 puntos. Entre los impactos potenciales se encuentra la ejecución de código, lo que podría permitir a un atacante tomar control total del sistema, afectando así la integridad y la confidencialidad de los sistemas involucrados. Para remediar estas vulnerabilidades, es necesario aplicar los parches disponibles en los repositorios de VMware.

- **MikroTik RouterOS:**

  RouterOS es el sistema operativo que utilizan los *routers* de la marca MikroTik. Está presente en un activo del laboratorio, donde se han identificado cuatro vulnerabilidades con un promedio de severidad de 7.6. Los problemas que podrían surgir incluyen la escalación de privilegios, la denegación de servicio y la ejecución de código de forma remota. Para remediar estas vulnerabilidades, es necesario actualizar RouterOS a la versión 6.48.6 o posterior.

4.2.3.2. Vulnerabilidades con severidad media

En la Figura 4.8 se pueden observar las vulnerabilidades encontradas en la Micro-Red.

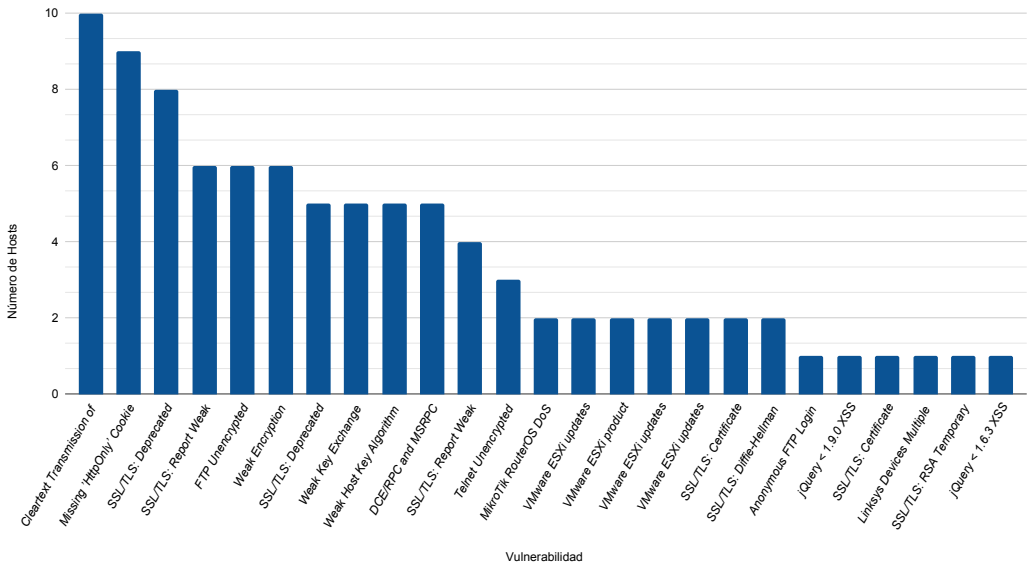

Figura 4.8: Numero y tipo de vulnerabilidades de nivel medio presentes en la Micro-Red.

Son 37 los *hosts* en los que estas vulnerabilidades están presentes. Seguidamente, se mostrará un resumen de las vulnerabilidades con mayor presencia en los activos.

- ***Cleartext transmission of sensitive information via HTTP:***

  La vulnerabilidad consiste en la transmisión de información sensible en texto claro a través de un canal de comunicación no cifrado. Esto ocurre generalmente en protocolos como HTTP, que no tienen seguridad inherente para encriptar los datos enviados entre un cliente y un servidor. Su severidad es de 4.8 puntos y su impacto potencial radica en ataques de tipo *man in the middle* Man-in-the-Middle (MITM), lo que permitiría la divulgación y corrupción de información que podría perjudicar las operaciones del laboratorio. Entre las medidas de remediación, se incluye deshabilitar los protocolos inseguros o utilizar cifrado en la comunicación mediante SSL/TLS, como lo hace el protocolo *Hypertext Transfer Protocol Secure* (HTTPS).

- ***Missing 'HttpOnly' Cookie Attribute (HTTP):***

  Esta vulnerabilidad se debe a la falta del atributo HttpOnly en las *cookies* que almacenan información sensible. Cuando las *cookies* no poseen este atributo en su configuración, un atacante puede explotar esta situación mediante un ataque de *script* entre sitios (XSS por sus siglas en inglés). En este tipo de ataque, el atacante inserta código malicioso en un sitio web que luego accede a las *cookies* sin el atributo HttpOnly. La severidad de esta vulnerabilidad es de 5 puntos. Para remediarla, es necesario activar el atributo HttpOnly y realizar una revisión periódica de las configuraciones de las *cookies*.

- **Vulnerabilidades con protocolos SSL/TLS:**

  Existen seis diferentes tipos de vulnerabilidades relacionadas con los protocolos de cifrado de datos SSL/TLS. Las amenazas de seguridad están ligadas a versiones obsoletas de los protocolos TLS y SSL, certificados expirados, cifrados ineficientes y el uso de claves de un tamaño menor a 2048 bits. El promedio de severidad de estas vulnerabilidades es de 4.8 puntos. Para remediarlas, es necesario revisar las configuraciones en los servidores Apache y Nginx, asegurándose de que las versiones de estos protocolos estén actualizadas a las más recientes.

4.2.3.3. Vulnerabilidades con severidad baja

En la Figura 4.9, podemos observar las vulnerabilidades presentes en el laboratorio.

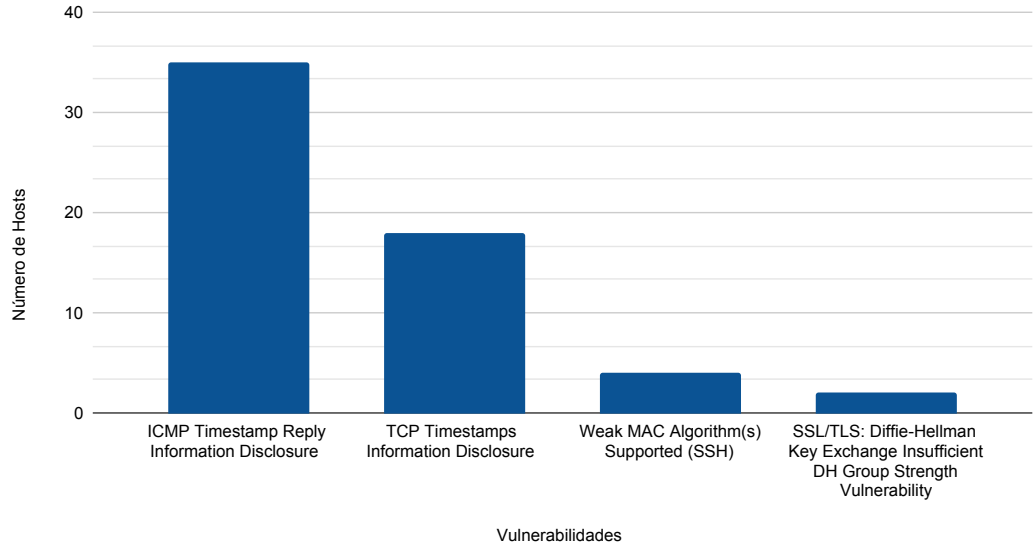

Figura 4.9: Numero y tipo de vulnerabilidades de nivel bajo presentes en la Micro-Red.

Son 39 los activos afectados por esta vulnerabilidad. A continuación, presentamos un resumen de estas amenazas.

- ***SSL/TLS, Diffie-Hellman Key Exchange Insufficient DH Group Strength Vulnerability:***

  Esta vulnerabilidad permite a un atacante interceptar y descifrar datos transmitidos de manera segura entre clientes y servidores. Básicamente, se explota un error en el manejo del relleno de SSLv3 para degradar la calidad del cifrado. Posteriormente, se ejecutan solicitudes repetidas que pueden revelar el texto claro de la comunicación cifrada. Una solución efectiva es deshabilitar esta versión de SSLv3 y actualizar las configuraciones de TLS.

- ***Weak MAC Algorithm(s) Supported (SSH):***

  Esta vulnerabilidad está asociada con el soporte de algoritmos débiles de MAC en SSH, ya que este servidor permite el uso de algoritmos criptográficos como MD5 o HMAC-MD5, que no ofrecen protección adecuada contra ataques criptográficos. Esto permite que un atacante manipule o falsifique los datos de una sesión SSH sin que el receptor se dé cuenta de la manipulación. Para remediar esta amenaza, es necesario configurar el servidor SSH para que opere con algoritmos de cifrado más fuertes y seguros.

- ***TCP Timestamps Information Disclosure:***

Esta vulnerabilidad está relacionada con los *timestamps* de TCP, lo que permite a un atacante deducir información acerca de los sistemas objetivos, como el tiempo de actividad del sistema. Su nivel de severidad es de 2.6 puntos. Esta vulnerabilidad se puede mitigar desactivando los *timestamps* de TCP o utilizando *firewalls* y filtros para bloquear solicitudes de fuentes externas no confiables.

- ***ICMP Timestamp Reply Information Disclosure:***

  Esta vulnerabilidad permite a un atacante obtener información sobre la configuración temporal del sistema de un *host* objetivo, lo cual podría aprovechar para inferir tiempos de actividad, ajustar ataques temporales o explotar vulnerabilidades que dependan de la sincronización. La severidad de esta vulnerabilidad es de 2.1 puntos. Para remediarla, se recomienda bloquear las solicitudes y respuestas de *timestamp* y capacitar al personal sobre los riesgos asociados con la divulgación de información a través de protocolos inseguros.

4.3. Evaluación de correcciones con *hardening*

En esta última sección, se detalla y evalúa la efectividad de las correcciones aplicadas mediante *hardening*, una fase que no solo corrige vulnerabilidades identificadas durante el *pentesting* sino que también establece una defensa robusta contra futuros ataques.

4.3.1. Actualización de *firmware* en MikroTik RouterOS

La actualización de *firmware* en el dispositivo MikroTik RouterOS se llevó a cabo para mitigar las vulnerabilidades identificadas mediante un análisis realizado con la herramienta *OpenVAS*. El escaneo de seguridad previo a la actualización, como se observa en la Figura **??**, reveló tres vulnerabilidades de baja criticidad, cuatro de criticidad media y cuatro de alta criticidad, lo que representaba un nivel de riesgo considerable para la integridad del sistema y la seguridad de la red.

Posteriormente, se actualizó el *firmware* del dispositivo de la versión 6.49 a la versión 6.49.15. Tras la actualización, se realizó un nuevo análisis de vulnerabilidades para evaluar la efectividad de esta medida. Los resultados de este nuevo análisis, presentados en la Tabla 4.2, mostraron una reducción significativa en el número y la criticidad de las vulnerabilidades detectadas: las vulnerabilidades de alta criticidad fueron eliminadas por completo, y las de criticidad

media se redujeron de cuatro a tres.

Tipo de vulnerabilidad	Pre-*Hardening*	Pos-*Hardening*
Baja	3	3
Media	4	3
Alta	4	0

Tabla 4.2: Comparación del nivel de vulnerabilidades en MikroTik RouterOS antes y después de la actualización de *firmware*.

4.3.2. *Hardening* en el Servidor Ubuntu

El proceso de *hardening* en el servidor Ubuntu 22.04, que también aloja el servicio Apache Guacamole, comenzó con una evaluación inicial de controles de seguridad utilizando *scripts* de auditoría. La auditoría inicial mostró que de un total de 139 controles de seguridad, 75 no estaban aplicados. los resultados de esta evaluación pre-*hardening* se muestra en la Tabla 4.3.

Las mejoras en la seguridad involucraron la aplicación de las remediaciones sin implementar, reflejándose en la mejora general de la seguridad del servidor. Las modificaciones resultaron en la aplicación efectiva de todos los controles de seguridad, como se evidencia en la evaluación post-*hardening*, ilustrada en la Tabla 4.3.

Estado de Controles	Pre-*Hardening*	Pos-*Hardening*
Fail	75	0
Ok	64	139

Tabla 4.3: Estado de controles de seguridad en el Servidor Ubuntu.

En términos de vulnerabilidades, el análisis inicial de OpenVAS detectó dos de baja criticidad y dos de alta criticidad en el servidor, incluyendo las asociadas con Apache Guacamole. Después de la aplicación del *hardening* y el cambio de contraseña por defecto a un contraseña robusta generada con *KeePass 2* para el servidor Apache Guacamole, se realizó un nuevo análisis, que confirmó la eliminación de todas las vulnerabilidades detectadas, como se muestra en la Tabla 4.4.

La implementación de las mejoras en Apache Guacamole y el servidor Ubuntu no solo optimizó la configuración de seguridad sino también validó la efectividad de los controles integrados de auditoría y remedio. El enfoque interactivo del *script* de *hardening*, que solicitaba confirmación antes de ejecutar cada control, permitió que los administradores personalizaran el proceso

Tipo de vulnerabilidad	Pre-*Hardening*	Pos-*Hardening*
Baja	2	0
Media	0	0
Alta	2	0

Tabla 4.4: Comparación del nivel de vulnerabilidades antes y después del *hardening* en el Servidor Ubuntu.

según las necesidades operativas específicas del sistema, asegurando así la integridad y estabilidad del mismo.

4.3.3. *Hardening* en el Servidor Windows

El proceso de *hardening* del servidor Windows Server 2012 R2 comenzó con la ejecución de un *script* de auditoría diseñado para alinear la configuración del servidor con las normativas de seguridad internas y las mejores prácticas recomendadas. La auditoría inicial reveló que solo 11 de los 63 controles de seguridad establecidos habían sido aplicados, mientras que 52 controles aún requerían implementación. Este estado inicial se documenta en la Tabla 4.5, que muestra las deficiencias en la configuración de seguridad del servidor.

Siguiendo estos hallazgos, se procedió con la remediación, que aplicó correcciones donde fuese necesario. Tras la implementación de estas medidas de remediación, un análisis posterior mostró que los 63 controles de seguridad habían sido satisfactoriamente establecidos, reflejando la efectividad de las acciones tomadas para fortalecer la seguridad del servidor. El resultado final de este proceso se ilustra en la Tabla 4.5, donde se puede observar que todas las recomendaciones de seguridad ahora están aplicadas.

Estado de Controles	Pre-*Hardening*	Pos-*Hardening*
Fail	52	0
Ok	11	63

Tabla 4.5: Estado de controles de seguridad en el Servidor Windows.

Durante el proceso, se añadió funcionalidad interactiva al *script* de *hardening*, que requería la confirmación del administrador antes de ejecutar cada control de seguridad. Esta característica permitió un mayor control y personalización del proceso de *hardening*, permitiendo a los administradores mantener ciertos controles que podrían afectar operaciones críticas, si lo consideraban necesario.

La eliminación completa de las vulnerabilidades a través de la aplicación efectiva de todos

los controles de seguridad muestra la importancia de una revisión de seguridad continua y adaptativa para mantener la integridad y la protección de los sistemas de información.

Este enfoque narrativo continuo permite destacar la transformación en la seguridad del servidor Windows 2012 R2 de manera fluida y cohesiva, subrayando tanto la metodología aplicada como los resultados obtenidos.

5. Conclusiones y recomendaciones

5.1. Conclusiones

La implementación de *Inventory* y los agentes de *Gestionnaire Libre de Parc Informatique* (GLPI) ha proporcionado una herramienta valiosa para la automatización del inventario de los activos Tecnologías de la Información (TI) y Tecnologías de Operación (TO) del Laboratorio de la Micro-Red, permitiendo obtener detalles de hardware y software de los activos donde los agentes están instalados, así como información sobre el estado de las interfaces de red en los activos que se detectan mediante descubrimiento de red. Además, la integración de un programa que trabaja con las *Application Programming Interface*s (APIs) de GLPI y OpenCVE permite a los administradores del laboratorio identificar posibles vulnerabilidades derivadas de configuraciones erróneas.

Una vez catalogados los activos del laboratorio, se implementó el *framework* Daniz-Red, el cual identificó con éxito las vulnerabilidades y las posibles correcciones necesarias. La clave de su compatibilidad con un sistema *Industrial Control System* (ICS) es la integración de OpenVAS, que, al poseer un mayor número de plantillas *Network Vulnerability Tests* (NVT) en comparación con Nuclei, ha permitido analizar los dispositivos industriales. Además, el *framework* es capaz de generar informes detallados en formato PDF y DOCX, lo que facilita la elaboración de reportes técnicos completos con información sobre la severidad, detalles de la vulnerabilidad, *Common Vulnerabilities and Exposure*s (CVEs) asociados y posibles soluciones.

La implementación del proceso de *hardening* en los servidores del laboratorio ha sido crucial para reforzar la seguridad de la infraestructura crítica. Este proceso se centró en corregir las vulnerabilidades detectadas sin requerir un período de mantenimiento disruptivo, permitiendo una operatividad continua y eficiente. Los *scripts* de *hardening*, diseñados según las recomendaciones del *Center for Internet Security* (CIS), han sido efectivos para remediar fallas existentes y prevenir futuras amenazas.

Al corregir las vulnerabilidades y volver a analizar los activos remediados, se observó una disminución significativa en el nivel de severidad. Este resultado muestra la capacidad del *framework* para mejorar la seguridad de la infraestructura crítica del laboratorio.

El diseño e implementación del *framework* de *pentesting* ha cumplido con éxito su objetivo de identificar vulnerabilidades criticas. El proceso de *hardening* aplicado ha demostrando ser una estrategia efectiva para mejorar la seguridad del laboratorio sin interrumpir las operaciones del

mismo. Esta mejora ha incrementado el nivel de madurez en ciberseguridad a la Micro-Red, estableciendo una base sólida para futuros trabajos.

5.2. Recomendaciones

En respuesta a los hallazgos y experiencias adquiridas durante la implementación del *framework* de *pentesting* y *hardening* en el laboratorio de Micro-Red, se sugieren las siguientes recomendaciones estratégicas para mejorar la seguridad de los sistemas de información críticos.

Es recomendable coordinar la aplicación del proceso de *pentesting* y *hardening* con los períodos de mantenimiento del ICS. Dado que algunas operaciones críticas no pueden detenerse sin consecuencias significativas, realizar estos procedimientos de seguridad durante mantenimientos programados minimizará el impacto operacional. Esto también permitirá la implementación de correcciones que podrían ser potencialmente disruptivas, garantizando que todas las medidas necesarias se apliquen sin afectar la continuidad del servicio.

Asimismo, se debe establecer un registro detallado que contenga toda la documentación relevante sobre cada activo, incluyendo manuales, configuraciones, especificaciones técnicas y cualquier otra información pertinente. Este registro facilitará la identificación precisa de vulnerabilidades y mejorará la eficacia de las estrategias de mitigación aplicadas. Además, un registro bien mantenido apoya la gestión de activos y la planificación de la seguridad a largo plazo, proporcionando una base de datos consolidada que puede ser rápidamente consultada para tomar decisiones informadas sobre la seguridad.

Finalmente, es fundamental asegurar la continuidad en la efectividad de las estrategias de seguridad, por lo que se debería desarrollar un *framework* que automatice las tareas de *pentesting* y *hardening*. Este sistema debería operar de manera periódica, ejecutando análisis de seguridad y aplicando medidas de mitigación automáticamente. Además, es importante que las herramientas utilizadas dentro del *framework* sean regularmente actualizadas para combatir las amenazas emergentes de manera proactiva. También se recomienda la integración de herramientas para entrenamiento contra amenazas, como GoPhish y Mimecast, así como la implementación de un *front-end* con interfaz gráfica. Este enfoque no solo incrementará la eficiencia y efectividad de las operaciones de seguridad, sino que también fortalecerá la capacidad de respuesta ante incidentes de seguridad potenciales.

Referencias

[1] S. Pascual de Vega, “Micro redes y redes inteligentes,” 2018, accepted: 2018-09-05T12:32:37Z. [En línea]. Disponible: https://uvadoc.uva.es/handle/10324/31400

[2] D. D. Jiménez Mendieta y J. A. Sumba Naula, “Construcción e implementación de un modelo para diagnosticar el nivel de la ciberseguridad en una micro-red. Caso de Estudio: micro-red de la Universidad de Cuenca,” May 2023, accepted: 2023-05-25T19:32:45Z Publisher: Univerisdad de Cuenca. [En línea]. Disponible: http://dspace.ucuenca.edu.ec/handle/123456789/41963

[3] L. Guerrero y E. G. De, “La importancia de las pruebas de penetración (Parte I),” Ene. 2012, accepted: 2017-08-03T16:02:30Z Publisher: Universidad Nacional Autónoma de México, Dirección General de Cómputo y de Tecnologías de Información y Comunicación. [En línea]. Disponible: https://www.ru.tic.unam.mx/xmlui/handle/123456789/1762

[4] IBM, “¿qué son las pruebas de penetración? | ibm,” https://www.ibm.com/es-es/topics/penetration-testing, [En línea]. Disponible en:.

[5] J. Luswata, P. Zavarsky, B. Swar, y D. Zvabva, “Analysis of scada security using penetration testing: A case study on modbus tcp protocol,” in *29th Biennial Symposium on Communications (BSC)*, Toronto, ON, Canada, 2018, pp. 1–5.

[6] K. Petrosyan, “¿qué son las pruebas de penetración y por qué son importantes?” https://easydmarc.com/blog/es/que-son-las-pruebas-de-penetracion-y-por-que-son-importantes/, EasyDMARC, [En línea]. Disponible en:.

[7] H. Hilal y A. Nangim, “Network security analysis scada system automation on industrial process,” 11 2017, pp. 1–6.

[8] B. D. Quinde Saltos y C. L. Campoverde Andrade, “Diseño e implementación de un framework para la evaluación periódica de la seguridad usando pruebas de penetración en la red interna de la Universidad de Cuenca,” bachelorThesis, Universidad de Cuenca, Oct. 2021, accepted: 2021-10-26T20:02:06Z. [En línea]. Disponible: http://dspace.ucuenca.edu.ec/handle/123456789/37143

[9] “CIS Compliance | Achieving CIS Benchmark Compliance - ManageEngine Vulnerability Manager Plus.” [En línea]. Disponible: https://www.manageengine.

com/vulnerability-management/cis-compliance.html?network=g&device=c&keyword=cis%20benchmark&campaignid=18895994081&creative=634442455436&matchtype=e&adposition=&placement=&adgroup=144201432675&targetid=kwd-327189531819&location=1005373&target=&gclid=Cj0KCQjwhL6pBhDjARIsAGx8D5_PGfGkqc_1dAC6xrUlV2vvhaw3uKkHK5nh58jeHcStllt6buLYnY8aAnXzEALw_wcB

[10] A. Echeverría, C. Cevallos, I. Ortiz-Garces, y R. O. Andrade, "Cybersecurity Model Based on Hardening for Secure Internet of Things Implementation," *Applied Sciences*, vol. 11, num. 7, p. 3260, Ene. 2021, number: 7 Publisher: Multidisciplinary Digital Publishing Institute. [En línea]. Disponible: https://www.mdpi.com/2076-3417/11/7/3260

[11] J. A. Guacaneme, D. Velasco, y C. L. Trujillo, "REVISIÓN DE LAS CARACTERÍSTICAS DE SISTEMAS DE ALMACENAMIENTO DE ENERGÍA PARA APLICACIONES EN MICRO REDES," *Información tecnológica*, vol. 25, pp. 175 – 188, 00 2014. [En línea]. Disponible: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642014000200020&nrm=iso

[12] F. J. Ramón Ducoy, "Implantación de energías renovables en un planta de producción de amoniaco." [En línea]. Disponible: https://biblus.us.es/bibing/proyectos/abreproy/5145/fichero/3.+%C2%BFQu%C3%A9+es+una+microrred.pdf

[13] S. Chowdhury, S. P. Chowdhury, y P. Crossley, *Microgrids and Active Distribution Networks*. IET Digital Library. [En línea]. Disponible: https://digital-library.theiet.org/content/books/po/pbrn006e

[14] G. Konstantinou, "Chapter 8: SCADA and smart energy grid control automation," p. 67. [En línea]. Disponible: https://www.jaunty.eu/wp-content/uploads/2022/08/8.-SCADA-and-Smart-Energy-Grid-Control-Automation-Smart-Energy-Grid-Engineering.pdf

[15] A. M. Y. Koay, R. K. L. Ko, H. Hettema, y K. Radke, "Machine learning in industrial control system (ICS) security: current landscape, opportunities and challenges," vol. 60, num. 2, pp. 377–405. [En línea]. Disponible: https://doi.org/10.1007/s10844-022-00753-1

[16] X. Etxezarreta, I. Garitano, M. Iturbe, y U. Zurutuza, "Software-defined networking approaches for intrusion response in industrial control systems: A survey," *International Journal of Critical Infrastructure Protection*, vol. 42, p. 100615, 2023. [En línea]. Disponible: https://www.sciencedirect.com/science/article/pii/S1874548223000288

[17] K. KOBARA, “Cyber physical security for industrial control systems and iot,” *IEICE Transactions on Information and Systems*, vol. E99.D, num. 4, pp. 787–795, 2016.

[18] C. A. MACEDO y J. Menting, “Building a cybersecurity culture in the industrial control system environment,” *International Journal of Information Security and Cybercrime*, 2019.

[19] G. M. Makrakis, C. Kolias, G. Kambourakis, C. Rieger, y J. Benjamin, “Industrial and critical infrastructure security: Technical analysis of real-life security incidents,” *IEEE Access*, vol. 9, p. 165295–165325, 2021. [En línea]. Disponible: http://dx.doi.org/10.1109/ACCESS.2021.3133348

[20] M. Nankya, R. Chataut, y R. Akl, “Securing industrial control systems: Components, cyber threats, and machine learning-driven defense strategies,” vol. 23, num. 21, p. 8840. [En línea]. Disponible: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10649322/

[21] A. Kamur. Network security policy | infosec. [En línea]. Disponible: https://www.infosecinstitute.com/resources/network-security-101/network-security-policy-part-3/

[22] F. Djebbar y K. Nordström, “A comparative analysis of industrial cybersecurity standards,” vol. 11, pp. 85 315–85 332, conference Name: IEEE Access. [En línea]. Disponible: https://ieeexplore.ieee.org/document/10210561

[23] H. Gall, “Functional safety IEC 61508 / IEC 61511 the impact to certification and the user,” in *2008 IEEE/ACS International Conference on Computer Systems and Applications*, pp. 1027–1031, ISSN: 2161-5330. [En línea]. Disponible: https://ieeexplore.ieee.org/abstract/document/4493673

[24] K. Stouffer, S. Lightman, V. Pillitteri, M. Abrams, y A. Hahn, “Guide to industrial control systems (ICS) security.” [En línea]. Disponible: https://csrc.nist.gov/pubs/sp/800/82/r2/final

[25] S. Untiveros. Explorando el estándar IEEE 1686-2019: Ciberseguridad en tecnologías operativas (OT) – AprendaRedes. [En línea]. Disponible: https://www.aprendaredes.com/explorando-el-estandar-ieee-1686-2019-ciberseguridad-en-tecnologias-operativas-ot/

[26] N. Lee y D. Manners, “Gamification of penetration testing,” in *Counterterrorism and Cybersecurity: Total Information Awareness*, N. Lee, Ed. Springer International Publishing, pp. 343–347. [En línea]. Disponible: https://doi.org/10.1007/978-3-319-17244-6_14

[27] R. Johari, I. Kaur, R. Tripathi, y K. Gupta, “Penetration testing in IoT network,” in *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pp. 1–7. [En línea]. Disponible: https://ieeexplore.ieee.org/document/9276853

[28] breachlock_dev. The basics of penetration testing. [En línea]. Disponible: https://www.breachlock.com/resources/blog/the-basics-of-penetration-testing/

[29] A. Tyas Tunggal. What is penetration testing? | UpGuard. [En línea]. Disponible: https://www.upguard.com/blog/penetration-testing

[30] K. McConkey. Red teaming 101: An introduction to red teaming and how it improves your cyber security. [En línea]. Disponible: https://www.pwc.co.uk/issues/cyber-security-services/insights/what-is-red-teaming.html

[31] S. Sunaringtyas y D. Prayoga, “Implementasi penetration testing execution standard untuk uji penetrasi pada layanan single sign-on,” *Edu Komputika Journal*, vol. 8, num. 1, pp. 48–56, Jun. 2021. [En línea]. Disponible: https://journal.unnes.ac.id/sju/edukom/article/view/47179

[32] B. Dinis y C. Serrão, “Using ptes and open-source tools as a way to conduct external footprinting security assessments for intelligence gathering,” vol. 3, pp. 271–279, 2014.

[33] DARKRELAY. Deep dive into penetration testing methodologies. [En línea]. Disponible: https://www.darkrelay.com/post/a-deep-dive-into-penetration-testing-methodologies

[34] A. E. Ibor y J. N. Obidinnu, “System hardening architecture for safer access to critical business data,” vol. 34, num. 4, pp. 788–792, number: 4. [En línea]. Disponible: https://www.ajol.info/index.php/njt/article/view/124044

[35] R. Tatam. System hardening checklist, meaning, types, examples + more. [En línea]. Disponible: https://www.puppet.com/blog/system-hardening

[36] R. Sasidharan, “A case study to implement windows system hardening using CIS controls,” vol. 70, pp. 1–7.

[37] briasmitatms. Center for internet security (CIS) benchmarks - microsoft compliance. [En línea]. Disponible: https://learn.microsoft.com/en-us/compliance/regulatory/offering-cis-benchmark

[38] anjalisingh. Linux OS hardening: CIS benchmarks. [En línea]. Disponible: https://opstree.com/blog/2020/04/29/linux-os-hardening-cis-benchmarks/

[39] H. Attila, P. M. Erdősi, y F. Kiss, “The common vulnerability scoring system (CVSS) generations – usefulness and deficiencies,” pp. 137–153.

[40] E. Fleischmann. CVSS (common vulnerability scoring system) □ redlings. [En línea]. Disponible: https://www.redlings.com/en/guide/cvss-score

[41] I. Lee. What is common vulnerability scoring system (CVSS)? [En línea]. Disponible: https://www.wallarm.com/what/what-is-cvss

[42] “Common Vulnerability Scoring System SIG.” [En línea]. Disponible: https://www.first.org/cvss/

[43] C. Kolf, “Full CVSSv3x Coverage in the Greenbone Feeds,” Oct. 2021. [En línea]. Disponible: https://www.greenbone.net/en/blog/full-cvssv3x-coverage-in-the-greenbone-feeds/

[44] D. J. Marin, “Manual de uso general software glpi,” May 2023, accepted: 2023-06-22T02:39:09Z Publisher: Tecnológico de Antioquia, Institución Universitaria. [En línea]. Disponible: https://dspace.tdea.edu.co/handle/tdea/3603

[45] “Documentación GLPI.” [En línea]. Disponible: https://glpi-project.org/es/documentacion/

[46] KitPloit, “OpenCVE - CVE Alerting Platform - vulnerability database | Vulners.com,” Ene. 2021, section: kitploit. [En línea]. Disponible: https://vulners.com/kitploit/KITPLOIT:8252141968149880216

[47] “How it Works - OpenCVE Documentation (Category fork).” [En línea]. Disponible: https://gael-lejeune.github.io/opencve-docs/how-it-works/

[48] “Guía de referencia de Nmap (Página de manual).” [En línea]. Disponible: https://nmap.org/man/es/index.html

[49] C. Martorella, “laramies/theHarvester,” Jun. 2024, original-date: 2011-01-01T20:40:15Z. [En línea]. Disponible: https://github.com/laramies/theHarvester

[50] “projectdiscovery/nuclei,” Jun. 2024, original-date: 2020-04-03T18:47:11Z. [En línea]. Disponible: https://github.com/projectdiscovery/nuclei

[51] “OpenVAS - Escáner abierto de evaluación de vulnerabilidades.” [En línea]. Disponible: https://openvas.org/

[52] “Qué es OpenVAS, para qué sirve y características,” Nov. 2020. [En línea]. Disponible: https://openwebinars.net/blog/que-es-openvas/

[53] “Evaluación de vulnerabilidades usando OpenVAS.” [En línea]. Disponible: https://www.welivesecurity.com/es/recursos-herramientas/evaluacion-vulnerabilidades-openvas/

[54] D. Crawford. Keepass review | completely free - very secure - hard to use. [En línea]. Disponible: https://proprivacy.com/password-manager/review/keepass

[55] K. Jin, Z. Niu, J. Liu, J. Bai, y L. Zhang, “Research on network security technology of industrial control system,” vol. 355, p. 03067.

[56] H. Holm, M. Karresand, A. Vidström, y E. Westring, “A survey of industrial control system testbeds,” in *Secure IT Systems*, S. Buchegger y M. Dam, Eds. Springer International Publishing, pp. 11–26.

[57] T. Sommestad, G. N. Ericsson, y J. Nordlander, “SCADA system cyber security — a comparison of standards,” in *IEEE PES General Meeting*, pp. 1–8, ISSN: 1944-9925. [En línea]. Disponible: https://ieeexplore.ieee.org/document/5590215

[58] M. Slunjski, D. Sumina, S. Groš, y I. Erceg, “Off-the-shelf solutions as potential cyber threats to industrial environments and simple-to-implement protection methodology,” vol. 10, pp. 114 735–114 748, conference Name: IEEE Access. [En línea]. Disponible: https://ieeexplore.ieee.org/document/9931720

[59] R. Chapas, A. Hristova, T. Locher, y S. Obermeier, “Securing industrial control systems through autonomous hardening.”

[60] B. A. Guachichullca Guamán, “Diseño e implementación de una Arquitectura de Ciberseguridad para la Micro-red de la Universidad de Cuenca.” Feb. 2024.

[61] W. Stallings, “Snmpv3: A security enhancement for snmp,” *IEEE Communications Surveys*, vol. 1, num. 1, pp. 2–17, 1998.

Anexos

Anexo A: Instalación de GLPI *Inventory*

GLPI permite la instalación de extensiones que facilitan a los administradores la gestión de activos de una organización. En este caso, se procederá a instalar GLPI *Inventory*, que permite llevar un registro detallado y actualizado de los activos, se puede acceder directamente a través de la tienda de extensiones de GLPI. El enlace para descargar la extensión es: https://plugins.glpi-project.org/#/plugin/glpiinventory. Dado que el GLPI corre sobre Ubuntu Server 22.04, el primer paso es descargar el paquete utilizando el comando del Extracto de código 1.

```
wget https://github.com/glpi-project/glpi-inventory-plugin/releases/
    download/1.3.5/glpi-glpiinventory-1.3.5.tar.bz2
```

Extracto de código 1: Descarga de paquete GLPI *Inventory*

Una vez descargado el archivo en el servidor, es necesario descomprimirlo con el comando que se muestra en el Extracto de código 2.

```
tar -xjf glpi-glpiinventory-1.3.5.tar.bz2
```

Extracto de código 2: Descomprimir paquete GLPI *Inventory*

El siguiente paso es mover el directorio glpiinventory al directorio de plugins de GLPI. Este proceso también incluye otorgar los permisos necesarios para que GLPI pueda ejecutar la extensión, los comandos se pueden observar en el Extracto de código 3.

```
sudo mv glpiinventory /pathGLPI/var/www/html/glpi/plugins/
sudo chmod +x /pathGLPI/var/www/html/glpi/plugins/glpiinventory
```

Extracto de código 3: Configuración de paquete GLPI *Inventory*

Finalmente, para instalar el plugin GLPI Inventory, ingrese a la página web de GLPI y diríjase a la pestaña de Configuración en el menú principal, identificado como (1). A continuación, seleccione la pestaña Plugins (2), donde se mostrará la lista de extensiones disponibles. Localice *Inventory* en esta lista y proceda a instalarlo haciendo clic en la pestaña correspondiente, ilustrado con el indicador (3) en la Figura 1.

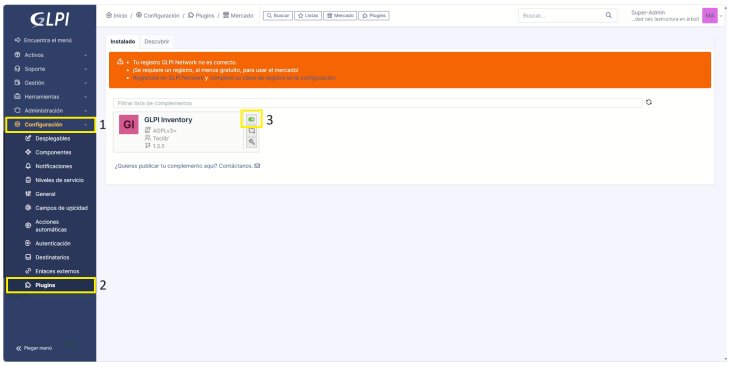

Figura 1: Instalación de GLPI *Inventory* desde interfaz web.

Anexo B: Instalación de Agentes GLPI

Los agentes GLPI tienen como objetivo recopilar información detallada de cada uno de los dispositivos que se encuentran dentro de la red, tanto de hardware como de software, para posteriormente enviarla al servidor GLPI. Estos agentes son compatibles con varios sistemas operativos y pueden descargarse desde su repositorio en GitHub https://github.com/glpi-project/glpi-agent/releases/tag/1.7.1.

Windows

Para el sistema operativo Windows, disponemos de dos versiones, cada una dedicada a computadoras con arquitectura de 64 y 32 bits. El archivo tiene una extensión .msi y, al ejecutarlo, se desplegará una ventana como se muestra en la Figura 2.

Figura 2: Interfaz de instalación, agente GLPI en Windows.

Lo siguiente será presionar el botón *Next* para que se muestren los acuerdos de licencia y la ubicación donde se instalará el agente. En la interfaz se mostrará el modo de instalación; en este paso específico, se seleccionará el modo *Custom*. Esto nos dará como resultado las utilidades del agente que estarán implementadas, por lo que se recomienda habilitar todas las herramientas, como se muestra en la Figura 3.

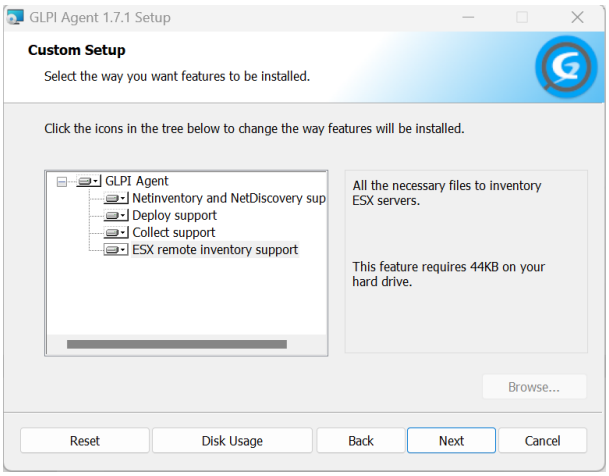

Figura 3: Interfaz con características a instalar, agente GLPI en Windows.

Ahora se debe configurar el servidor GLPI al que llegará la información de los dispositivos. Para esto, ingresaremos la URL del servidor en el recuadro que se muestra en la Figura 4.

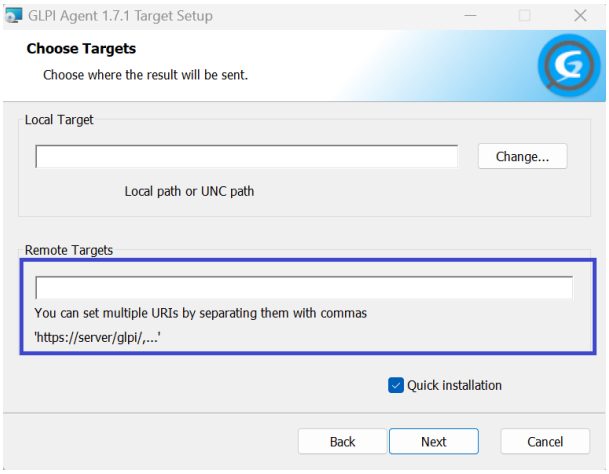

Figura 4: Definir URL del Servidor, agente GLPI en Windows.

Finalmente, haremos clic en *Next*, y se desplegará la interfaz que se muestra en la Figura 5, donde se debe hacer clic en *Install*. Luego, concederemos los permisos de administrador y esperaremos a que se instale el agente. Una vez instalado, el agente trabajará en segundo plano, esperará un tiempo para obtener la información del dispositivo y la enviará al servidor.

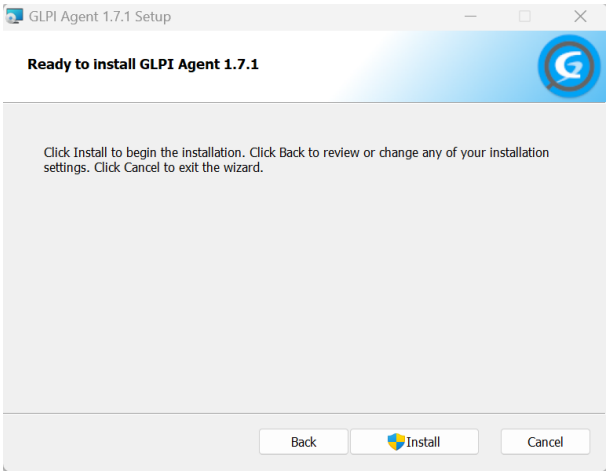

Figura 5: Instalar agente GLPI en Windows.

Linux

Para instalar en sistemas Linux, debemos hacerlo mediante la línea de comando. El primer paso es descargar los paquetes desde su repositorio en GitHub, por lo que se usa los comandos que se encuentra en Extracto de código 4.

```
1 wget https://github.com/glpi-project/glpi-agent/releases/download
    /1.7.1/glpi-agent-1.7.1-linux-installer.pl
```

Extracto de código 4: Descarga de paquetes agente GLPI.

Una vez obtenidos los paquetes del agente, se pueden instalar mediante Perl con el comando que se muestra en el Extracto de código 5.

```
1 sudo perl glpi-agent-1.7.1-linux-installer.pl
```

Extracto de código 5: Instalación de paquetes agente GLPI.

Durante el proceso de instalación de Perl, para configurar la URL del servidor de GLPI, se mostrará el mensaje que aparece en el Extracto de código 6.

```
1 Provide an url to configure GLPI server:
2 > http://192.xx.xx.xx:80
```

Extracto de código 6: Definir URL del Servidor GLPI.

Una vez que los agentes hayan recolectado la información de los dispositivos, enviarán al servidor GLPI los datos de hardware y software. Para verificar los agentes, primero hay que

dirigirse a la pestaña Administración en el menú de GLPI (1), y posteriormente ir a Inventario (2). En el icono Número de agentes, se mostrará cuántos dispositivos tienen instalado el agente, como se puede ver en la Figura 6. Al ingresar, observaremos el listado de los dispositivos con cada una de sus especificaciones.

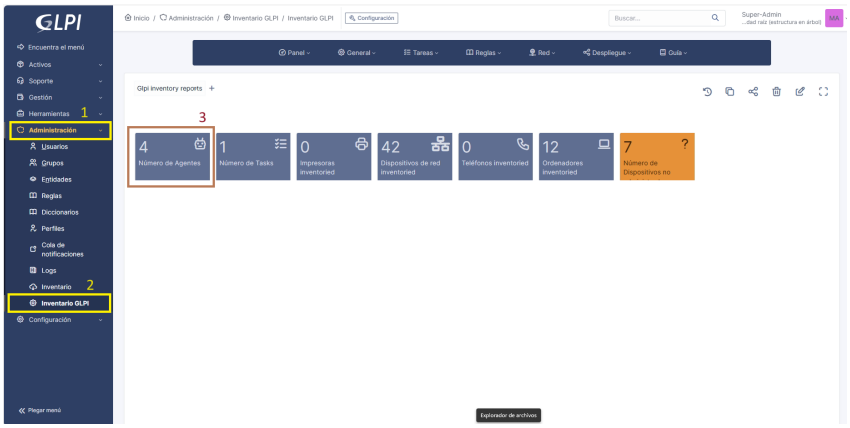

Figura 6: Dispositivos conectados al servidor por medio de agente GLPI.

Anexo C: Instalación de Daniz-Red *Framework*

El primer paso para instalar el framework Daniz-Red es instalar las herramientas Git, Docker y docker-compose. Esto se realiza ejecutando el comando del Extracto de código 7 .

```
1 sudo apt-get install git docker.io docker-compose
```

Extracto de código 7: Instalación de Docker y Docker-compose en Ubuntu

Con esta configuración, Docker correrá bajo los privilegios de superusuario. Para cubrir esto y evitar posibles fallos en la administración de los contenedores, es necesario agregar al usuario del sistema operativo al grupo Docker. Esto se realiza mediante el comando mostrado en el Extracto de código 8.

```
1 sudo usermod -aG docker $(whoami)
```

Extracto de código 8: Agregar permisos de grupo Docker a usuario del sistema.

Para el siguiente paso necesitamos descargar los repositorios del *framework* Daniz-Red desde los repositorio de Github, esto lo podremos realizar mediante el enlace https://github.com/fabianastudillo/daniz-red.git, pero para nuestra implementación se necesita descargar desde la rama *Supervisory Control and Data Acquisition* (SCADA), estos comandos se pueden ver en el Extracto de código 9.

```
git clone https://github.com/fabianastudillo/daniz-red.git
cd daniz-red
git checkout SCADA
```

Extracto de código 9: Descarga de Daniz-Red versión SCADA

Una vez descargado el repositorio y dentro del directorio del *framework*, es necesario ejecutar el comando del Extracto de código 10 para montar los contenedores de Daniz y OpenVAS, una vez levantado el servicio nos mostrara la Figura 7.

```
docker-compose -p danizred up -d
```

Extracto de código 10: Levantar servicios Daniz-Red y OpenVAS

Figura 7: Contenedores Daniz-Red y OpenVAS levantados.

Una vez levantados los servicios de Kali Linux y OpenVAS, podemos ingresar al contenedor de kali para ejecutar el *framework* Daniz-Red. Para esto, ejecutamos los comandos del extracto de código 11, lo que nos dará como resultado la figura 8

```
docker exec -it kali bash
python3 daniz-run.py -c config.json
```

Extracto de código 11: Levantar servicios Daniz-Red y OpenVAS

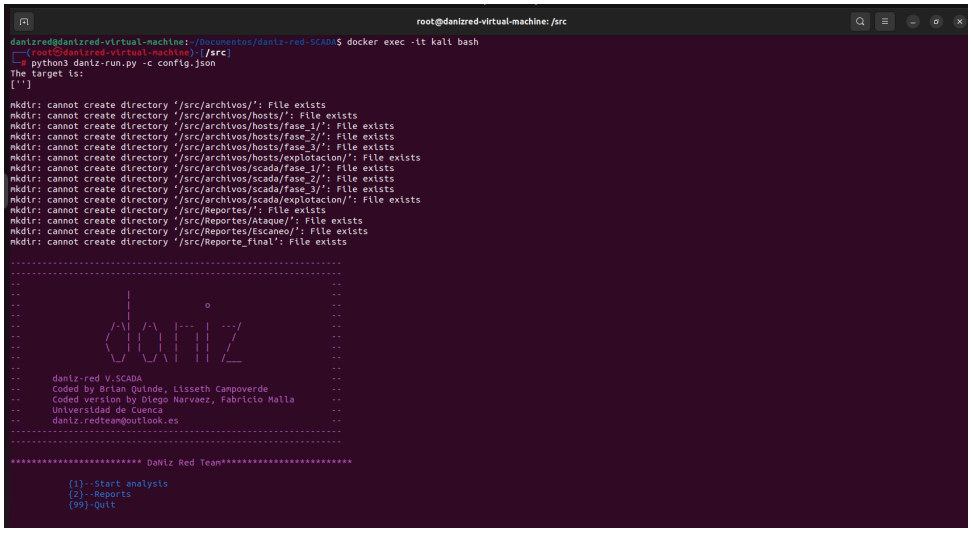

Figura 8: Ejecución de Daniz-Red desde contenedor

Además, para acceder al servicio OpenVAS, es necesario ingresar desde el navegador web a la dirección, lo que dará como resultado una interfaz como la que se muestra en la Figura 9. Si los *dashboards* no se cargan, se debe esperar un momento hasta que se actualicen los formularios CVE, NVT, escáneres de vulnerabilidades, entre otros.

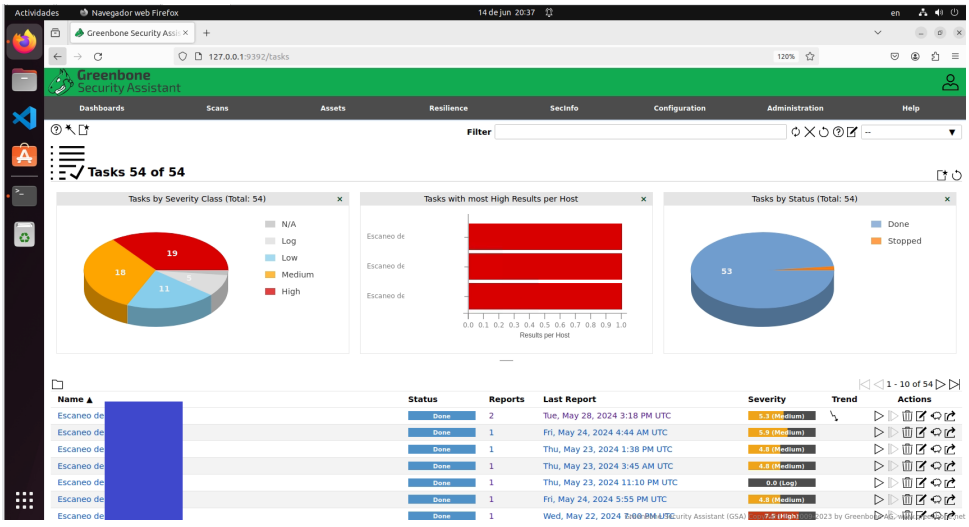

Figura 9: Dashboard de OpenVas

Anexo D: Explotación de vulnerabilidades en Apache Guacamole

Para realizar la explotación en el servicio Apache Guacamole, mediante un navegador web en la barra de direcciones se introdujo la dirección IP**192.168.XXX.XXX** con el puerto **8443**, para así ingresar a la interfaz del servicio, esto se puede observar en la Figura 10.

Figura 10: Ingreso a la interfaz web de Guacamole.

Una vez dentro de la interfaz se ingresaron los credenciales por defecto, (admin: guacadmin y password: guacadmin), tal y como se observa en las figuras 11 y 12.

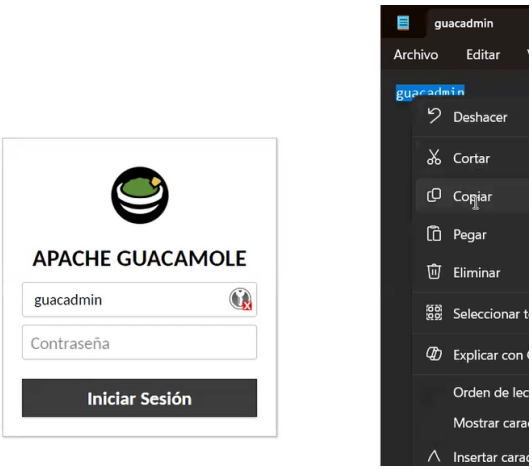

Figura 11: Se copia la contraseña por defecto.

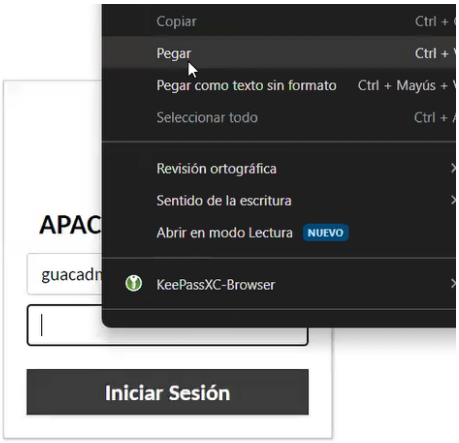

Figura 12: Ingreso de la contraseña por defecto.

Ya con las credenciales ingresadas se puede observar en la figura 13 como se accedió al servicio Apache Guacamole el cual permite tener el control remoto del sistema SCADA, tal y como se aprecia en la figura 3.9 de la subsección 3.3.2.1.

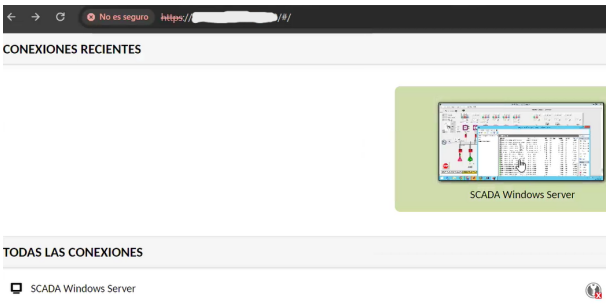

Figura 13: Acceso a la interfaz de Apache Guacamole.

Anexo E: Generación y Cambio de Contraseña en Apache Guacamole

Como primer paso para la actualización de la contraseña es generar una nueva que cumpla con los estándares de seguridad, Para esto se hizo uso del programa *KeePass 2*, previamente configurado. Como se muestra en la Figura 14, para ingresar a *KeePass 2* es necesario utilizar credenciales personalizadas por el usuario.

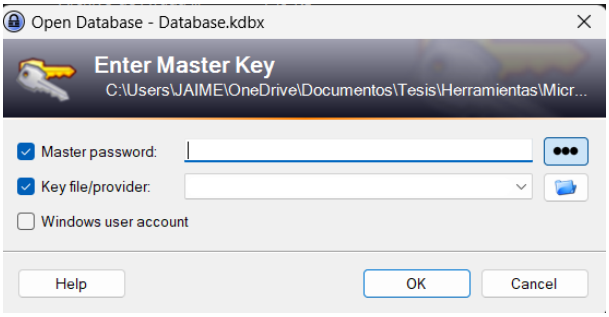

Figura 14: Acceso al programa *KeePass 2*.

Una vez dentro del programa, el siguiente paso es crear una nueva entrada, como se observa en la Figura 15. Antes de crear la entrada, es crucial configurarla adecuadamente. El aspecto más importante es la selección de la opción “Password Generator”, donde se configura el nivel de seguridad de la contraseña. Para este caso específico, se eligió una configuración que incluye una combinación de letras mayúsculas, minúsculas y números, como se muestra en la Figura 16.

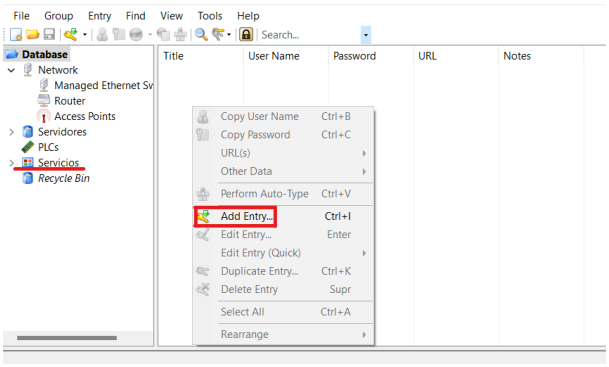

Figura 15: Creación de una nueva entrada en *KeePass 2*.

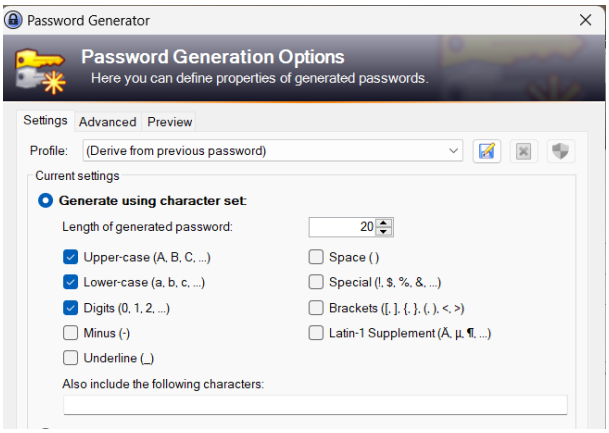

Figura 16: Configuración del nivel de seguridad de la contraseña a generar.

El siguiente paso consiste en actualizar la contraseña por defecto del servicio Apache Guacamole con la nueva contraseña generada. Dentro de la interfaz del servicio, se accede a la opción de configuración en la sección de usuarios, como se ilustra en la Figura 17.

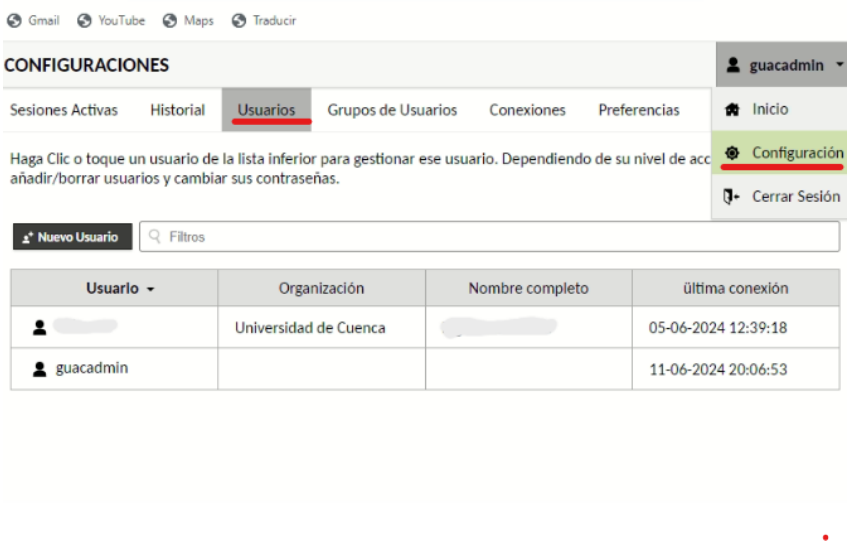

Figura 17: Ingreso a las configuraciones de Apache Guacamole, sección usuarios.

Finalmente, se selecciona el usuario a modificar, en este caso 'guacadmin'. Dentro de la configuración del usuario, se introduce y valida la nueva contraseña, se guardan las configuraciones y con esto se completa el cambio de contraseña.

Anexo F: Proceso de actualización del *Firmware* al Dispositivo MikroTik RouterOS

Para actualizar el *firmware* del dispositivo MikroTik RouterOS, el primer paso es acceder a su interfaz a través de un navegador web. La Figura 18 muestra la pantalla de inicio de sesión, donde se deben introducir las credenciales para acceder a las configuraciones del equipo.

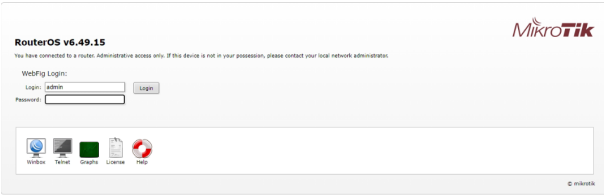

Figura 18: Interfaz de acceso de MikroTik RouterOS.

Una vez dentro de la interfaz, navegue hasta la pestaña "Quick Set". En la parte inferior de esta pantalla, haga clic en el botón "Check For Updates", como se ilustra en las Figuras 19 y

20.

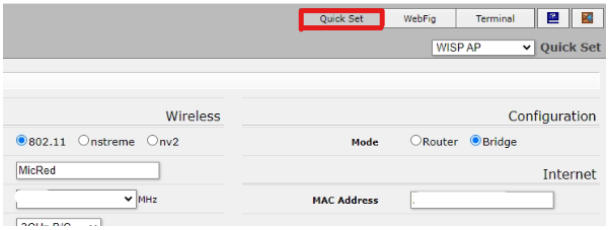

Figura 19: Pestaña “Quick Set” en MikroTik RouterOS.

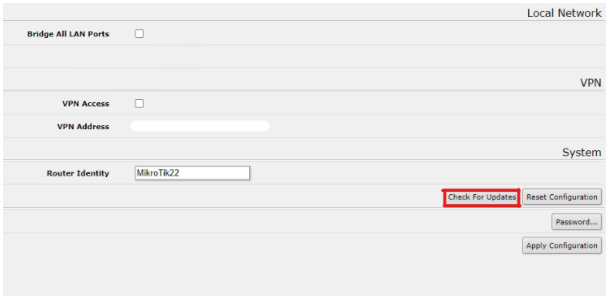

Figura 20: Botón “Check For Updates” en MikroTik RouterOS.

En la ventana que aparece tras seleccionar “Check For Updates”, presione el botón “Download&Upgrade”, como se aprecia en la Figura 21. El equipo iniciará automáticamente el proceso de actualización del firmware. Este proceso puede tardar varios minutos durante los cuales el dispositivo puede reiniciarse.

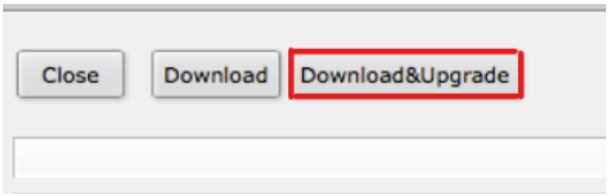

Figura 21: Botón “Download&Upgrade” en MikroTik RouterOS.

Asegúrese de no interrumpir la alimentación del dispositivo durante la actualización para evitar daños al dispositivo.

Anexo G: CVE mediante de implementación de GLPI y OpenCVE

Al implementar el sistema descrito en la Sección 3.1, basado en la información almacenada en GLPI, se han podido obtener CVEs de ocho modelos de activos del laboratorio, los cuales

se presentan a continuación:

- **Lenovo System X:**
  - **CVE (severidad):**

    CVE-2020-8340 (6.1, Media), CVE-2019-6157 (7.5, Alta), CVE-2018-9068 (7.5, Alta), CVE-2017-3775 (6.4, Media), CVE-2017-3768 (7.5, Alta), CVE-2017-3744 (6.5, Media), CVE-2016-8226 (6.8, Media)

  - **Activos afectados:**

    Servidor 1 IMM, Servidor 2 IMM

- **Windows 7:**
  - **CVE (severidad):**

    CVE-2012-5380 (6.0, Media), CVE-2012-5383 (6.2, Media), CVE-2012-5382 (6.0, Media), CVE-2012-5381 (6.0, Media), CVE-2012-5379 (6.0, Media), CVE-2011-1652 (5.0, Media), CVE-2024-23594 (6.4, Media), CVE-2024-23593 (6.7, Media), CVE-2010-0249 (8.8, Alta), CVE-2018-0749 (7.8, Alta), CVE-2018-0748 (7.8, Alta), CVE-2023-41840 (7.8, Alta), CVE-2023-33304 (5.5, Media), CVE-2023-37939 (3.3, Baja), CVE-2023-34367 (6.5, Media), CVE-2023-21899 (5.5, Media), CVE-2022-42970 (9.8, Critico), CVE-2022-42973 (7.8, Alta) CVE-2023-21898 (5.5, Media), CVE-2022-42971 (9.8, Critico)

  - **Activos afectados:**

    Torre Micro-Red

- **Raspberry Pi:**
  - **CVE (severidad):**

    CVE-2024-35932, CVE-2021-38759 (9.8, Critico), CVE-2021-38545 (5.9, Media), CVE-2019-20354 (4.3, Media), CVE-2018-18068, CVE-2018-18068 (9.8, Critico), CVE-2018-19860 (8.8, Alta), CVE-2017-15638 (6.5, Media)

  - **Activos afectados:**

    RP.1

- **Lynksys E900:**
  - **CVE (severidad):**

CVE-2021-46951 (5.5, Media), CVE-2021-47146, CVE-2021-47015, CVE-2023-38433 (7.5, Alta), CVE-2022-42455 (7.8, Alta), CVE-2019-5169 (7.8, Alta), CVE-2018-20404 (7.5, Alta), CVE-2014-9696 (8.8, Alta), CVE-2014-9695 (8.8, Alta), CVE-2016-6898 (6.6, Media)

- **Activos afectados:**

  AP1 MicroredAccess

- **SIMATIC S7-1500:**
  - **CVE (severidad):**

    CVE-2022-25622 (7.5, Alta), CVE-2019-19300 (7.5, Alta), CVE-2022-38465 (7.8, Alta), CVE-2021-37204 (7.5, Alta), CVE-2021-37185 (7.5, Alta), CVE-2021-37205 (7.5, Alta), CVE-2020-28397 (5.3, Media), CVE-2020-15782 (9.8, Critico), CVE-2020-15796 (7.5, Alta), CVE-2020-7580 (6.7, Media), CVE-2019-10936 (7.5, Alta), CVE-2019-19281 (7.5, Alta), CVE-2018-16559 (7.5, Alta), CVE-2018-16558 (7.5, Alta), CVE-2019-10929 (5.9, Media), CVE-2019-6575 (7.5, Alta), CVE-2019-10943 (7.5, Alta), CVE-2018-13805 (7.5, Alta), CVE-2018-13815 (7.5, Alta), CVE-2018-4843 (6.5, Media)

  - **Activos afectados:**

    PPC

- **SIMATIC S7-1500:**
  - **CVE (severidad):**

    CVE-2022-25622 (7.5, Alta), CVE-2019-19300 (7.5, Alta), CVE-2022-38465 (7.8, Alta), CVE-2021-37204 (7.5, Alta), CVE-2021-37185 (7.5, Alta), CVE-2021-37205 (7.5, Alta), CVE-2020-28397 (5.3, Media), CVE-2020-15782 (9.8, Critico), CVE-2020-15796 (7.5, Alta), CVE-2020-7580 (6.7, Media), CVE-2019-10936 (7.5, Alta), CVE-2019-19281 (7.5, Alta), CVE-2018-16559 (7.5, Alta), CVE-2018-16558 (7.5, Alta), CVE-2019-10929 (5.9, Media), CVE-2019-6575 (7.5, Alta), CVE-2019-10943 (7.5, Alta), CVE-2018-13805 (7.5, Alta), CVE-2018-13815 (7.5, Alta), CVE-2018-4843 (6.5, Media)

  - **Activos afectados:**

    PPC

- **Magelis HMI:**
  - **CVE (severidad):**

    CVE-2019-6833 (6.5, Media), CVE-2016-8374 (7.5, Alta), CVE-2016-8367 (5.3, Media)

  - **Activos afectados:**

    APIS5_SH_Magelis, APIS1_FV_Magelis, APIS2_ALM_Magelis, APIS3_GE_Magelis

- **Modicom M251:**
  - **CVE (severidad):**

    CVE-2019-6820 (8.2, Alta), CVE-2017-6026 (9.1, Critico), CVE-2017-6028 (9.8, Critico), CVE-2017-6030 (6.5, Media)

  - **Activos afectados:**

    PLC Apis1, PLC1 Apis2, PLC2 Apis2, PLC3 Apis2, PLC4 Apis2, PLC Apis3