E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Ciberseguridad del PLC Siemens Simatic S7-300

Grado en Ingeniería
en Tecnologías Industriales

Trabajo Fin de Grado

Eduardo Juan Sáenz Idoate

Pedro Julián Becerril Rodrigo

Pamplona, Junio

upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Contenido

TABLA DE ILUSTRACIONES

RESUMEN

Con la cuarta revolución industrial que se está produciendo en la actualidad y todas las tecnologías que la han habilitado, todo dispositivo presente en la industria está empezando a conectarse a la red. Esto implica que cada dispositivo miembro de un proceso industrial estará conectado entre sí y con la red en pocos años, lo que supone un avance tremendo en el control de procesos industriales, pero también crea unas amenazas que hasta ahora no existían en la industria.

Estas amenazas normalmente no han sido tenidas en cuenta durante el desarrollo de los dispositivos, ya sea por la reciente aparición de estas o por la errónea creencia de que con la protección y aislamiento proporcionados por la red sería suficiente, por tanto, la funcionalidad y robustez de los dispositivos priman en la actualidad sobre su ciberseguridad.

En el presente proyecto mediremos la ciberseguridad de un autómata marca Siemens modelo “Simatic S7-300”, en el escenario de una fábrica que adquirió este autómata en 1994 y no ha realizado su actualización con las últimas versiones proporcionadas por Siemens.

Los ataques se llevarán a cabo a través de la distribución Kali 2019.4 del sistema operativo Linux.

Palabras clave:

ICS

Autómata

PLC

Exploit

ABSTRACT

With the fourth industrial revolution that is taking place today and all the technologies that have enabled it, every device in the industry is starting to connect to the network. This implies that each member device of an industrial process will be connected to each other and to the network in a few years, which represents a tremendous advance in the control of industrial processes, but also creates threats that until now did not exist in the industry.

These threats have not normally been considered during the development of the devices, either due to their recent appearance or due to the mistaken belief that with the protection and isolation provided by the network were enough, therefore, the functionality and robustness of devices currently prevail over their cybersecurity.

In this project we will measure the cybersecurity of a Siemens Simatic S7-300 automaton, in the setting of a factory that acquired this automaton in 1994 and has not carried out its update with the latest versions provided by Siemens.

The exploitation will be carried out through the Kali 2019.4 distribution of the Linux operating system.

Key words:

ICS

Automaton

PLC

Exploit

1. Ciberseguridad Industrial

La ciberseguridad Industrial es un campo de la seguridad informática que ha surgido recientemente. Según el Instituto Nacional de Ciberseguridad se podría considerar la ciberseguridad como “el conjunto de prácticas, procesos y tecnologías, diseñadas para gestionar el riesgo del ciberespacio derivado del uso, procesamiento, almacenamiento y transmisión de información utilizada en las organizaciones e infraestructuras, utilizando las perspectivas de personas, procesos y tecnologías”. En los últimos años ha conseguido una gran relevancia que aumentará cada vez más conforme las industrias adopten la automatización. Esta automatización busca los siguientes objetivos:

· **Competitividad:** La globalización ha aumentado en gran medida la competitividad entre las empresas, que ya no compiten con sus homologas de la misma región sino con todas las del mundo. Por tanto, se requiere un aumento de la eficacia de la empresa mediante la reducción del coste de tareas, que se consigue mediante su automatización.

· **Seguridad:** La sustitución de personas por autómatas en la realización de tareas con riesgo permite una gran reducción de los accidentes.

· **Calidad:** Los autómatas son capaces de realizar tareas con una precisión y repetitividad inalcanzables para el hombre.

· **Disponibilidad de producto:** Se busca también aumentar el control sobre la tasa de producción, reduciendo las perdidas.

· **Flexibilidad:** Con un pequeño cambio de la programación se logra modificar tareas de forma extremadamente sencilla y rápida.

De aquí podría surgir la pregunta de si los autómatas lograrán en un futuro sustituir totalmente en la industria a los seres humanos. Hay tareas en las que obviamente las personas son insustituibles, pero cada día estos se perfeccionan y mejoran a límites inimaginables años antes, por tanto, es imposible saber dónde llegarán y podría ser que sustituyeran a los seres humanos incluso en las tareas más creativas en un futuro.

1.1. Historia

La seguridad informática tradicionalmente se ha centrado en la tecnología de la información (IT), que son los equipos informáticos encargados de manipular y transmitir datos, esto hace que manejen una gran cantidad de información y por tanto que tengan que ser extremadamente seguros.

En cambio, la tecnología de la operación (OT) se ha visto relegada a un segundo plano, esta tecnología se encarga de la monitorización y control de dispositivos y suele encontrarse en industrias. Debido a su lógico aislamiento del internet se ha creído que no era necesario desarrollar la ciberseguridad de estos dispositivos, que la red de la empresa debía cumplir esa función, solución que ha demostrado ser del todo ineficaz. Los dispositivos OT han sido desarrollados para trabajar en entornos duros y sin descanso, por tanto, su manipulación y actualización suele ser compleja y costosa ya que supone un parón de la producción. Por ello es común en las industrias encontrarse estos dispositivos desactualizados y vulnerables a un ataque.

1.2. Stuxnet

En 2010 se descubrió el gusano Stuxnet, que se ha convertido en el malware más famoso en lo que a sistemas industriales se refiere. Llego a infectar a una gran cantidad de ordenadores especialmente en Irán, en lo que llego a considerarse un ataque dentro de una guerra cibernética real. Stuxnet funcionaba infectando los programas WinCC y Step 7, software encargado de programar algunos PLCs, para después tomar el control de las librerías encargadas de comunicarse con el autómata. Así conseguía controlar la programación con la que se cargaba e introducir código malicioso sin que el ingeniero pudiera detectarlo. El comportamiento de Stuxnet se basaba en mantenerse indetectable mientras saboteaba el equipo controlado por el autómata.

Tras el descubrimiento de este ataque las vulnerabilidades en los dispositivos OT empezaron a tomarse muy en serio y expertos en ciberseguridad empezaron a buscarlas en los equipos y a exigir a las empresas fabricantes un mayor enfoque en la ciberseguridad durante el desarrollo de sus productos.

1.3. Actualidad

Hoy en día las grandes empresas ya otorgan a la ciberseguridad industrial una gran importancia. La integración de las tecnologías IT y OT ya es una realidad en los sistemas industriales de control (ICS). Pero que los expertos aumenten la seguridad de los dispositivos y redes no es suficiente, la seguridad informática es tarea de todos los miembros de una empresa, los atacantes intentarán acceder a la red a través de los eslabones más débiles y es por tanto indispensable la formación de todo el personal en esta materia. El gusano Stuxnet accedió a los ICS iraníes por un pendrive que conecto un empleado.

El sabotaje industrial informático, como ya se ha visto, será uno de los pilares de los futuros conflictos que sucedan en el planeta, es por tanto crucial contar una industria capaz de resistir y rechazar este tipo de ataques. El país que no cuente con una ciberseguridad adecuada en sus industrias será vulnerable a ataques de sus enemigos, lo que puede causar no solo una gran pérdida económica sino también en vidas, ya que un comportamiento inesperado de un dispositivo de control es muy fácil que cause un accidente, sin hablar de posibles desastres que podría causar el sabotaje a un ICS controlando algún sistema de una presa o de una central eléctrica, en especial de una nuclear.

1.4. Objetivos

Realizar hacking ético sobre el PLC Simatic S7-300 en un escenario en el que se encuentra controlando dos balizas señalizadoras del tráfico de un cruce, este se encontraría con la versión de firmware que tenía cuando se adquirió y no ha sido actualizado con las últimas versiones publicadas por Siemens. El atacante se encontrará en una situación en la que ya ha accedido a la red y debe buscar información sobre los dispositivos que se encuentran activos en ella, estudiar sus vulnerabilidades y atacarlos.

2. Equipo

En el presente proyecto se ha diseñado un laboratorio de ciberseguridad que cuenta con los siguientes equipos.

· Switch.

· Cableado Ethernet.

· Máquina virtual con el sistema operativo Windows XP.

· Máquina virtual con la distribución Kali del sistema operativo Linux.

· PLC Siemens Simatic S7-300.

· Fuente de alimentación

Con estos equipos se ha creado una red LAN que simula la que se podría encontrar en cualquier industria con ICS, obviamente en cualquiera de estas aparecerían estos elementos en mucha mayor cantidad, pero los sistemas de ataque que utilizaría un hacker serían muy similares.

La red diseñada tiene una topología en estrella, que es la más utilizada en las industrias actualmente, con los autómatas y las estaciones de programación conectadas entre sí a través de un switch y mediante cable Ethernet.

En la siguiente imagen se muestra una red con topología en estrella de una empresa cualquiera, la que se ha utilizado en este proyecto es exactamente igual reduciendo tan solo el número de PLC y estaciones de programación.

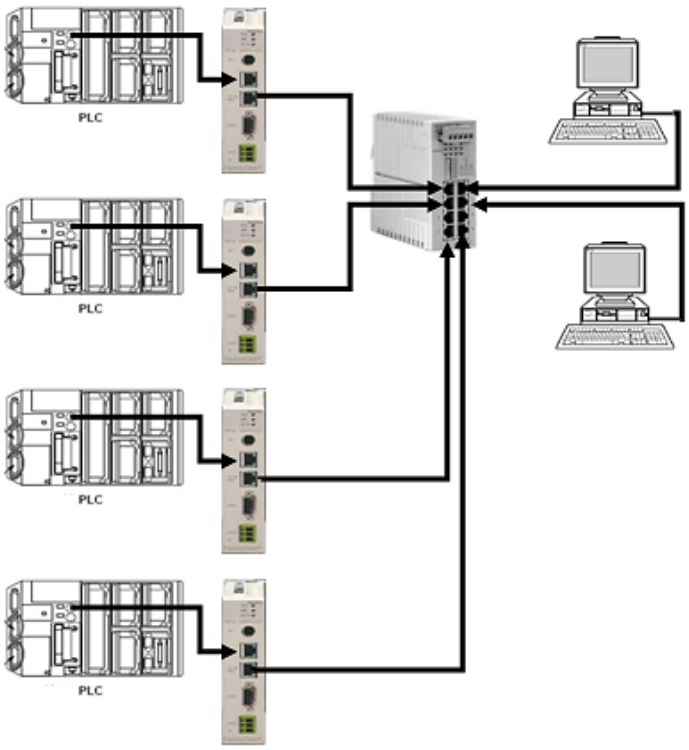

Ilustración 1: Red con topología en estrella.

A continuación, se estudiarán los equipos utilizados individualmente.

2.1. Switch

Es un dispositivo de interconexión de redes que se utiliza para la conexión de equipos formando una red LAN. Funcionan almacenando las direcciones MAC de los dispositivos, asignándoles un puerto, y reenviando los paquetes que reciben por el puerto correspondiente a la MAC de destino, de esta forma se evitan las colisiones entre paquetes que se producirían creando la red con un hub. Por esto actualmente ya no se encuentran hubs en empresas, ya que los switch tienen prestaciones muy superiores. Estos también permiten una muy fácil expansión de la red, debido a que se pueden conectar switch entre ellos y funcionar como uno solo.

A continuación, se muestra un esquema del funcionamiento de un Switch.

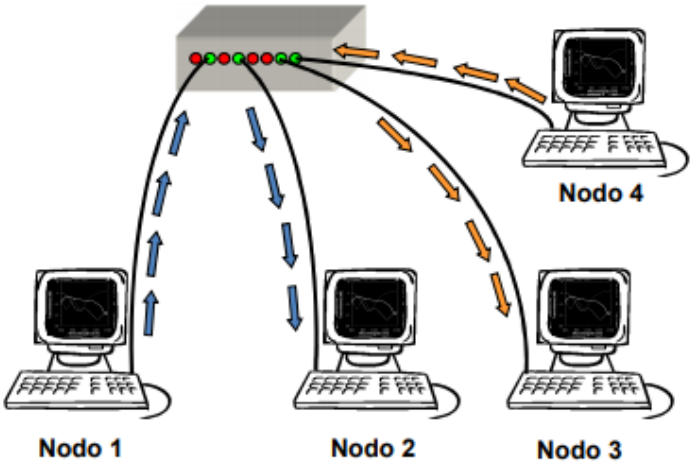

Ilustración 2: Esquema del funcionamiento de un Switch.

En el caso de este proyecto se ha utilizado un Switch de la marca Edimax de 8 puertos. De los que solo se han utilizado dos, uno para conectar el autómata y otro para el ordenador que contiene las máquinas virtuales. Se muestra en la siguiente ilustración.

Ilustración 3: Switch Edimax de 8 puertos.

2.2. Cableado Ethernet

Para la conexión entre los diferentes dispositivos y el switch se ha utilizado cableado tipo Ethernet categoría 6 marca OWLOTECH.

2.3. Máquina virtual como estación de programación

Esta máquina virtual ha sido creada mediante el software VMware v.15 y consiste en un Windows XP Proffesional. Se le han configurado las siguientes características.

Parámetro	Valor
Memoria RAM	2048 MB
Procesadores	2
Disco Duro	40 GB

Red	Adaptador puente

Tabla 1: Características de la estación de programación.

Se le han dado unas capacidades muy elevadas para tratarse de un sistema operativo tan antiguo debido a que se debía asegurar su buen funcionamiento y se cuenta con un ordenador con altas prestaciones.

La configuración de red que se ha escogido es adaptador puente, de esta forma la máquina actuará como si fuera una máquina física más en la red y simulará una estación de programación de una industria.

Para realizar la programación y monitorización del autómata se cuenta con el software Administrador Simatic Step 7 v.5.5. A continuación, se muestra una captura de la máquina.

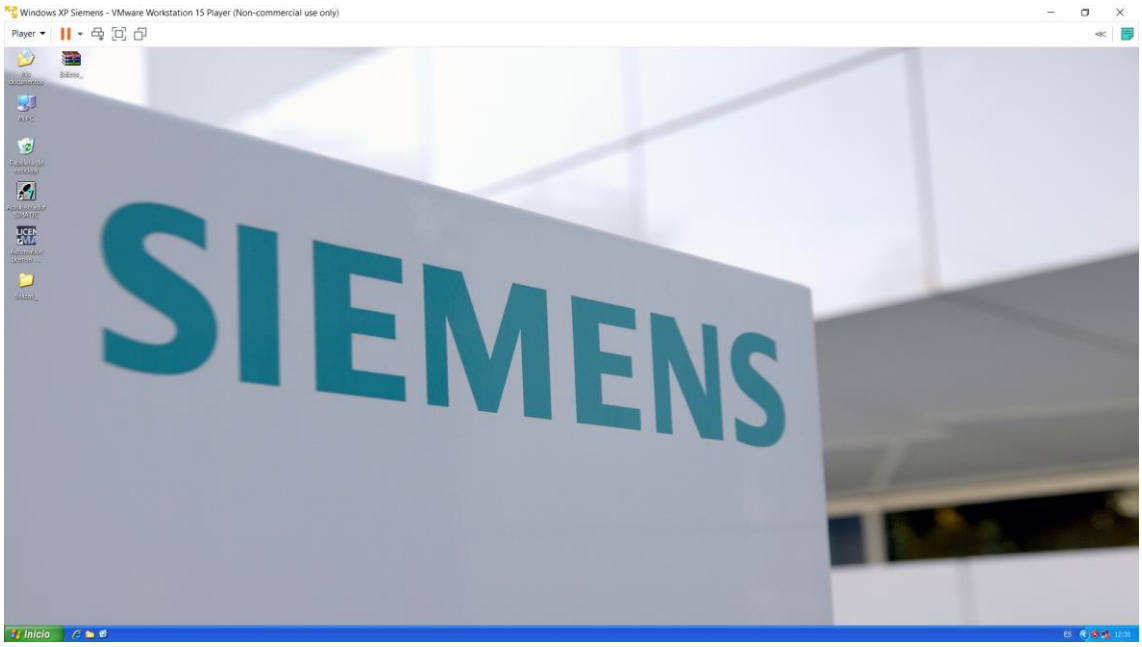

Ilustración 4: Máquina virtual Windows XP.

2.4. Máquina virtual con el sistema operativo Kali Linux

La virtualización de esta máquina se ha realizado mediante el software VirtualBox v.6.1 y consiste en la distribución Kali Linux 2019.4 y cuenta con la siguiente configuración.

Parámetro	**Valor**
Memoria RAM	8192 MB
Procesadores	4
Disco Duro	120 GB

Red	Adaptador puente

Tabla 2: Características de la máquina atacante.

En el caso de esta máquina se le han dado unas capacidades mucho mayores que las de la estación de programación debido a que en esta se tendrá que ejecutar software con muchos más requerimientos.

La configuración de la red que se ha escogido de nuevo es adaptador puente, para hacer el escenario de que está máquina es un ordenador físico en la red controlado por un usuario malintencionado con el objetivo de sabotear el sistema industrial. A continuación, se muestra una captura de la máquina.

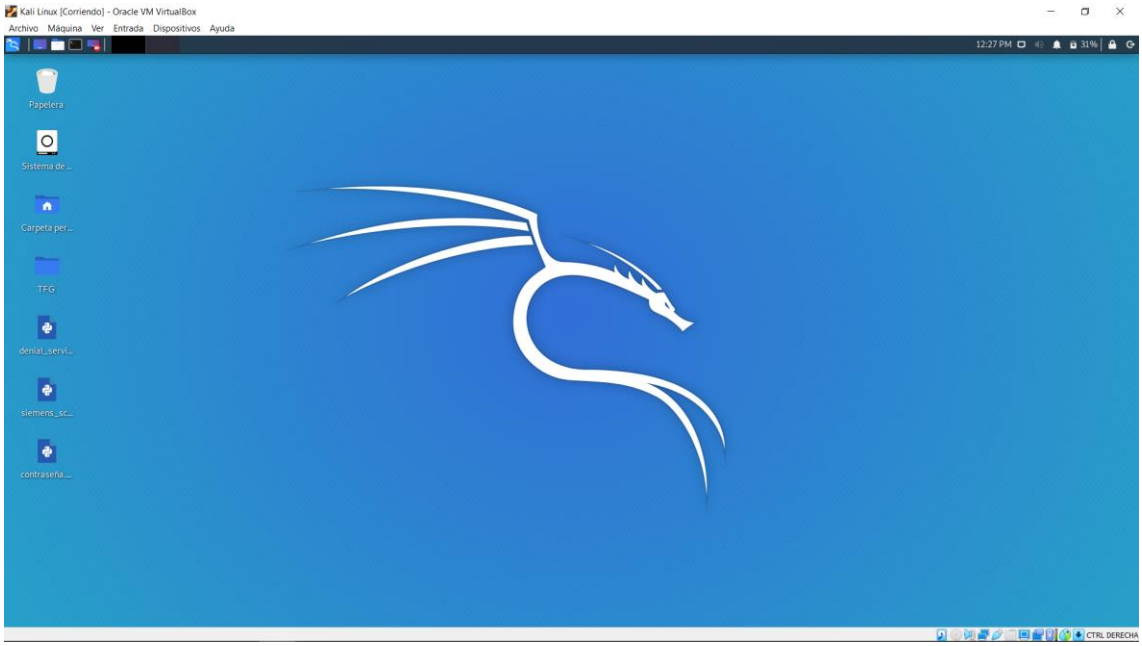

Ilustración 5: Máquina virtual Kali Linux.

Se utilizarán los siguientes programas que vienen por defecto en esta distribución:

· **Zenmap:** Este programa es la interfaz gráfica del software nmap, este programa es capaz de realizar escaneos de puertos de cualquier equipo conectado a la red. Proporciona información muy útil como la cantidad de dispositivos conectados y sus direcciones IP, en algunos casos es capaz también de proporcionar el sistema operativo que se está utilizando en un dispositivo. Su uso es muy sencillo e intuitivo para usuarios poco experimentados, tan solo requiere introducir el rango de direcciones IP en el que se realizará el escaneo y el tipo de escaneo que desea el usuario. Se han utilizado en este proyecto varios tipos de escaneos que se explican a continuación:

· Ping scan: Este escaneo tiene como objetivo el descubrimiento de sistemas en la red mediante el envío de sondas solicitando respuestas que demuestren que una dirección IP está activa. No aporta una gran cantidad de información al usuario, pero tiene la ventaja de no ser excesivamente intrusivo.

· Quick scan: Este escaneo es más rápido de lo normal debido a que utiliza técnicas de detección agresivas y escanea una cantidad limitada de puertos en los dispositivos.

· Intense scan: Incluye identificación del sistema operativo y de la versión instalada entre otras funciones. Es un escaneo más profundo que los anteriores y por tanto entrega una mayor cantidad de información al usuario, pero tiene la desventaja de ser mucho más intrusivo y por tanto mucho más fácil de detectar.

· **Ettercap:** Es un programa que permite realizar ataques Man In “The Middle” sobre redes LAN con switch. Mediante esta herramienta el usuario es capaz de conseguir que los paquetes que se están intercambiando dos dispositivos se redirijan a través de su ordenador, y este sea capaz de capturarlos.

· **Metasploit:** Esta herramienta esta diseña para ejecutar exploits contra dispositivos remotos, dispone de una gran librería de exploits y permite realizar ataques de una forma sencilla e intuitiva. Se ha utilizado la versión 5.0.71 que cuenta con 1962 exploits y 558 payloads.

· **Wireshark:** Se trata de un analizador de protocolos capaz de mostrar al usuario el tráfico que pasa a través de una red. Cuenta con una gran capacidad de filtrado que permite al usuario una rápida identificación de los paquetes deseados. En este proyecto se utilizará para el análisis de las comunicaciones del PLC con la estación de programación.

2.5. PLC Siemens Simatic S7-300

La empresa Siemens es muy conocida por la fabricación de autómatas programables, “programmable logic controllers” en inglés (PLC), estos son dispositivos capaces del control de procesos o de la automatización en la fabricación de productos, entre otras muchas aplicaciones. Su estructura es similar a la de un ordenador doméstico ya que básicamente son ordenadores digitales cuya única función es la automatización de procesos. Han sido diseñados para recoger datos a través de sus entradas analógicas y digitales y dar una respuesta mediante sus salidas a los actuadores. Pueden realizar las siguientes funciones:

· Recoger información a través de sus entradas.

· Tomar decisiones mediante la programación.

· Almacenar datos en la memoria.

· Generar ciclos de tiempo.

· Realizar cálculos matemáticos.

· Actuar sobre dispositivos externos a través de sus salidas.

· Comunicarse con otros sistemas externos.

2.5.1. Estructura externa

La estructura externa de un autómata puede ser de tres tipos:

· **Compacta:** Todos los elementos se encuentran en un mismo bloque, la fuente de alimentación, la CPU, las entradas y salidas, etc. Esta estructura la presentan los PLCs de gama baja, sus capacidades suelen ser limitadas.

·**Semimodular:** Se caracterizan por tener los módulos de entradas/salidas independientes de la CPU y de la memoria del programa. Se utiliza en PLCs de gama media.

· **Modular:** Los elementos se estructuran en módulos, esto permite añadir o quitar módulos para expandir o reducir el sistema en función de las necesidades. El PLC del presente proyecto se estructura de esta forma y tiene un módulo de entradas y salidas, uno para la CPU y otro de comunicaciones. Además, se alimenta con una fuente de alimentación externa conectada a la red mediante una protección magnetotérmica. La comunicación entre módulos se realiza mediante un bus de datos. Esta estructura se utiliza en autómatas de gama alta. A continuación, se incluye una instantánea del ICS creado para este proyecto, que contiene el autómata con los módulos mencionados.

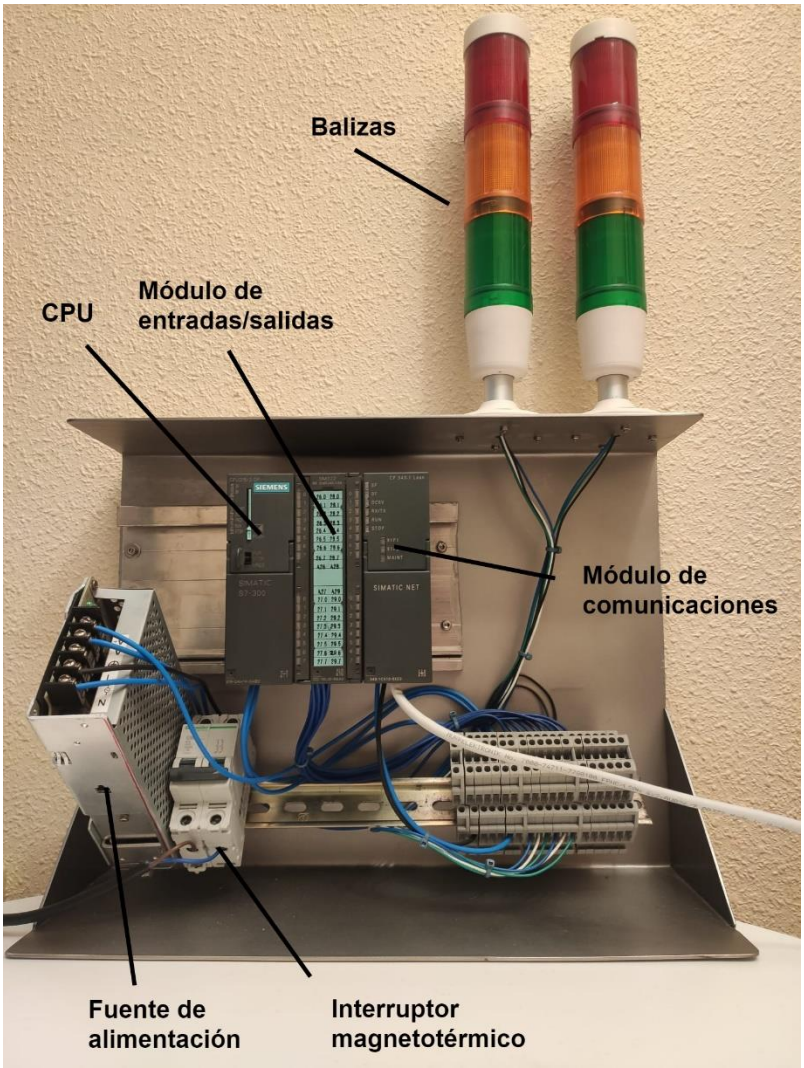

Ilustración 6: Alzado del ICS.

2.5.2. Estructura interna

· **Memoria:** Consta de varias regiones con datos insertados por el fabricante y otras libres para ser rellenadas por el usuario. Se distinguen tres áreas:

·Área de datos: Contiene datos de configuración de software.

·Área de programa: En esta área se guardarán las instrucciones de la programación del usuario.

· Área de configuración del sistema: Datos de la configuración y estructura de la programación aportada por el usuario.

· **CPU:** Gestiona el funcionamiento del resto de unidades y procesa los datos. Dispone de los siguientes elementos:

· Registros: Contienen los datos que hay que procesar, pueden estar compuestos por uno, dos o más bytes.

· Unidad aritmeticológica (ALU): Gestiona las operaciones aritméticas de suma, resta y las funciones lógicas AND, OR, NOT, etc.

· Contador de programa: Registro que almacena la dirección de la siguiente instrucción a ejecutar por la CPU.

· Decodificador de instrucciones: Interpreta las instrucciones.

· **Unidades de entrada y salida:** Comunican la CPU con el exterior.

· **Buses:** Conectan los diferentes dispositivos, existen varios tipos:

· Bus de datos: Su anchura determina la velocidad de procesado.

· Bus de direcciones: Su anchura determina la cantidad de direcciones que se pueden mapear.

· Bus de control: Gobierna el uso y acceso a los buses de datos y direcciones, evita las colisiones de información en el sistema.

· **Módulo de entradas digitales:** Permiten conectar al autómata con sensores digitales ON/OFF. Trabajan con señales de 24 V.

·**Módulo de salidas digitales:** Sirven para conectar el autómata con accionadores tipo ON/OFF que pueden ser de dos tipos:

· Salidas a relé: La tensión proviene de una fuente externa y se libera sobre la carga mediante la activación del relé.

· Salida estática: El conmutador en este caso es un transistor. Este tipo es más rápida y resistente ya que un relé es un accionamiento mecánico, y por tanto está mucho más limitado que un transistor.

· **Marcas:** Bit interno que permite guardar información temporalmente y donde se pueden realizar operaciones de escritura y lectura.

· **Temporizadores:** Dispositivos mecánicos, electrónicos o neumáticos que se activan pasado un determinado intervalo de tiempo.

· **Contadores:** Dispositivos mecánicos o eléctricos que van incrementándose hasta alcanzar un valor predeterminado, donde se activan.

· **Entradas analógicas:** Almacenan una magnitud analógica con una resolución que dependerá del número de bits que se disponga.

· **Salidas analógicas:** Permiten traducir un valor numérico en una señal de tensión o de corriente para controlar dispositivos.

2.5.3. Programación

En la industria existen muchos lenguajes de programación de automatismos, unos propios de cada autómata y otros comerciales, como el S7 siemens. Además, existen diferentes lenguajes de programación que son:

· **Lista de instrucciones (AWL):** El programa es representado mediante líneas de código que se traducen a lenguaje máquina mediante un compilador.

· **Esquema de funciones (FUP):** Las operaciones lógicas del programa se representan de forma gráfica.

· **Esquema de contactos (KOP):** El programa es representado mediante símbolos eléctricos.

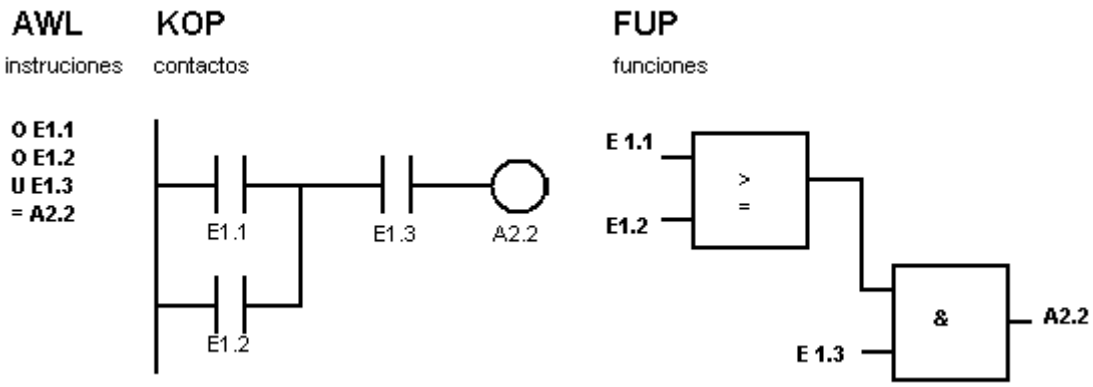

Ilustración 7: Lenguajes de programación de los autómatas.

Un proyecto puede tener distintas formas de programación coexistiendo, pero si se tiene KOP y FUP habrá que tener cuidado ya que no son totalmente compatibles.

Los PLC S7 disponen además de diferentes módulos software para simplificar la programación, en total son 1024 y los principales son:

· **Módulos de organización (OB):** Comunican la CPU con el programa del usuario, algunos tipos son:

  · Módulo principal (OB1): Se ejecuta cíclicamente y de él salen las llamadas al resto de módulos.

  · Módulo de arranque (OB100): En este módulo se pueden introducir valores que se desea que tenga el autómata por defecto al arrancarlo.

  · Módulo de error y alarma: Contienen la secuencia de acciones a realizar en caso de que se produzca un error o alarma programado.

· **Módulos de código (FC):** Sirven para incluir parte del programa del usuario de forma que se obtenga una programación más estructurada.

· **Módulos de función (FB):** En ellos se definen funciones complejas usadas habitualmente.

· **Módulos de datos (DB):** Contienen datos del usuario.

El autómata ejecutará el programa del usuario en un tiempo determinado que dependerá mayormente de la longitud de este, primero leerá las entradas reales y las almacenará en la memoria, tras esto ejecutará el programa usando los valores de la memoria y finalmente actualizará los valores de las salidas reales, en función del resultado de la ejecución del programa. El ciclo de trabajo del autómata será la suma de estos tres procesos y tras su finalización se repetirá de nuevo.

La variación de las entradas durante la ejecución del programa no modificará las salidas ya que se trabaja con una imagen en memoria de estas, para ver los resultados será necesario que se inicie un nuevo ciclo.

Durante la programación será necesario trabajar con diferentes tipos de datos que hay que saber clasificar, los principales son:

· E: Entradas.

· A: Salidas.

· M: Marcas.

Los tres tipos de datos cuentan con 2048 bytes de espacio en memoria cada uno en el PLC S7-300. Además, se pueden direccionar en 4 posibles modos:

· X: Para ocupar 1 bit, esta opción está por defecto si no se añade el direccionamiento.

· B: En el caso de querer ocupar 1 byte.

· W: Para palabras, 2 bytes.

· D: Para palabras dobles, 4 bytes.

Para realizar la programación se utilizar el software Step 7 TIA (Totally Integrated Portal), este programa permite el diseño de los sistemas industriales y su posterior volcado en los autómatas. Entre otras utilidades es posible realizar a través de este software la configuración de parámetros hardware como la dirección IP, hora, firmware, tipo de comunicación y realizar diagnósticos en el PLC. Se pueden distinguir las siguientes funciones en el programa:

· Programación.

· Comunicación.

· Diagnostico.

· Prueba.

2.5.4. Comunicaciones

Para poder comunicarse con el exterior el autómata cuenta con un módulo de comunicaciones Simatic Net CP 343-1 Lean. Este dispositivo permite la comunicación del autómata a Industrial Ethernet mediante el protocolo PROFINET IO además de

integrar un switch de dos puertos, de los que tan solo se utilizará uno ya que se cuenta con un switch externo. Este módulo soporta los siguientes protocolos de comunicación:

· **PROFINET IO**: El protocolo PROFINET es una evolución del protocolo PROFIBUS y de Industrial Ethernet IEC 61784, el tráfico utiliza el conjunto de protocolos TCP/IP comunicándose a través del puerto 102, el cual se utilizará más adelante para realizar los ataques. Fue concebido para facilitar el control de cientos o miles de PLCs de un sistema industrial de forma centralizada por una sola persona incluso, lo cual lo hace extremadamente eficiente. Es capaz de establecer prioridades en la red evitando su saturación y mejorando la seguridad. Profinet es uno de los estándares más acogidos a nivel industrial, a finales de 2018 había instalados más de 20 millones de dispositivos, cifra que aumenta cada día.

· **Protocolo S7**: Es un protocolo de comunicación propiedad de Siemens utilizado por la familia de PLCs Simatic S7 para la comunicación con la estación de programación, es capaz de realizar las siguientes funciones:

· Programar los PLC desde la estación de programación.

· Intercambiar datos con el PLC.

· Acceder a datos del PLC mediante sistemas SCADA (supervisory control and data acquisition).

La comunicación se produce mediante el envío de tramas de datos a través del Ethernet, estas tramas realizan un encapsulado de datos de la siguiente forma:

Tabla 3: Encapsulado de las tramas del protocolo S7.

· **Servidor Web HTTP:** El módulo CP incluye el acceso mediante un navegador a un servidor Web desde el cuál es posible comprobar el estado del autómata y recibir información sobre él. Para acceder hay que introducir la siguiente dirección en el navegador:

http:\\<Dirección IP del módulo CP>

2.6. Fuente de alimentación

Es necesario alimentar el autómata con tensión continua de 24V, por ello se requiere de una fuente de alimentación que sea capaz de proporcionar ese valor de tensión a partir de la red, de la que se obtiene una señal de 230 V en alterna. Para esta tarea se ha elegido una fuente marca OMRON modelo S82J-05024DD, con las siguientes características.

OMRON S82J-05024DD	
Voltaje de entrada	100-240 V en alterna

Potencia nominal	50 W
Voltaje de salida	24 V
Corriente de salida	2.1 A

Tabla 4: Características de la fuente de alimentación.

Por la entrada se introducirá la señal obtenida de cualquier enchufe conectado a la red dentro del rango, y mediante la salida se alimentará el autómata. Esta fuente cuenta además con las siguientes dimensiones.

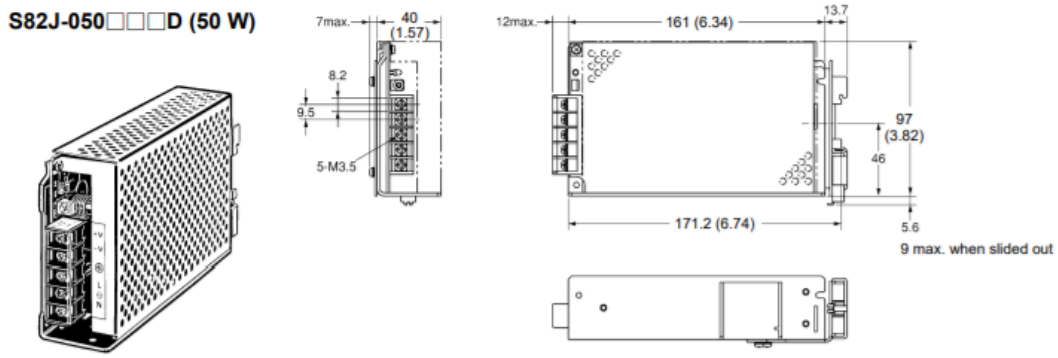

Ilustración 8: Dimensiones de la fuente de alimentación.

2.7. Protección magnetotérmica

Para la protección del circuito se ha colocado un interruptor magnetotérmico capaz de proteger el autómata y la fuente de alimentación frente a sobreintensidades. Este funciona cortando la corriente cuando esta sobrepasa unos límites a partir de los cuales se pueden llegar a dañar los equipos, en el caso de este proyecto se ha escogido un interruptor marca Schneider modelo C60N 24332, con las siguientes características.

Schneider C60N 24332	
Número de polos protegidos	2
Corriente nominal	2 A
Código de curva	C
Capacidad de corte	50 kA Icu en 220-240 V CA 50Hz

Tabla 5: Características del interruptor magnetotérmico.

Esta protección se ha colocado entre la red y la fuente de alimentación de forma que cuando el interruptor esté en OFF todo el sistema estará totalmente aislado de la red. A continuación se añade una ilustración del interruptor escogido.

Ilustración 9: Interruptor magnetotérmico.

3. Sistema Industrial

Para simular el trabajo de un autómata en un sistema industrial se ha diseñado un programa en lenguaje KOP que consiste en el control de los semáforos de un cruce. Cuando uno permite el paso encendiendo la luz verde el otro lo impide encendiendo la roja, en las transiciones de verde a roja se encenderá la luz amarilla indicando que la prohibición del paso va a comenzar. El funcionamiento durante el primer ciclo se muestra a continuación en forma de tabla, así como las direcciones de memoria asociadas a los colores de cada baliza.

	Baliza 1		**Baliza 2**	
Tiempo (s)	**Luz**	**Dirección**	**Luz**	**Dirección**
0	Verde	A0.0	Roja	A0.5
5	Amarilla	A0.1	Roja	A0.5
7	Roja	A0.2	Verde	A0.3
12	Roja	A0.2	Amarilla	A0.4
14	Verde	A0.0	Roja	A0.5

Tabla 6: Funcionamiento del primer ciclo del sistema industrial.

La programación se ha dividido en seis segmentos que se explicarán individualmente:

· Segmento 1:

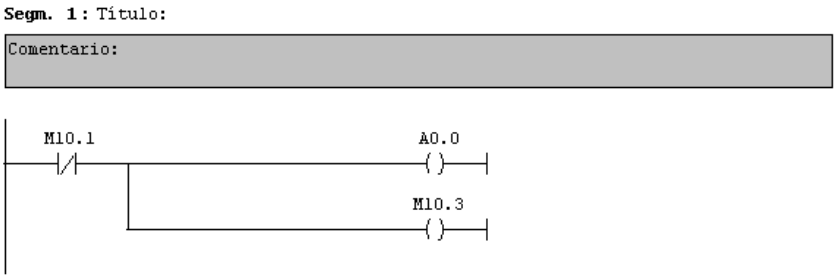

Ilustración 10: Primer segmento de la programación.

La primera parte de la programación tiene como objetivo el encendido de la luz verde de la primera baliza, como las marcas de la memoria se resetean a cero en cada apagado, el contacto M10.1 se ha escogido como normalmente cerrado. De esta forma este encenderá al iniciar el primer ciclo la luz verde con A0.0 y activará el primer temporizador con M10.3.

· Segmento 2:

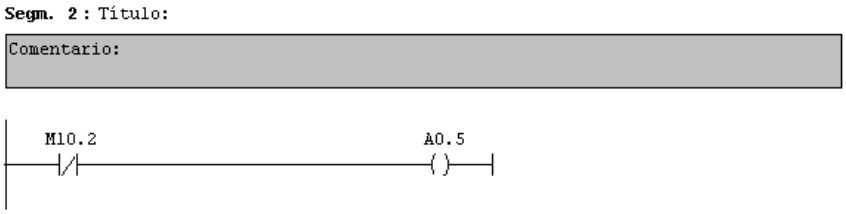

Ilustración 11: Segundo segmento de la programación.

Este segmento funciona de manera prácticamente igual al anterior, pero encendiendo la luz roja de la segunda baliza con A0.5, ha sido necesario crear un segmento especial para esta función debido a que el apagado se produce en instantes diferentes a la luz anterior.

· Segmento 3:

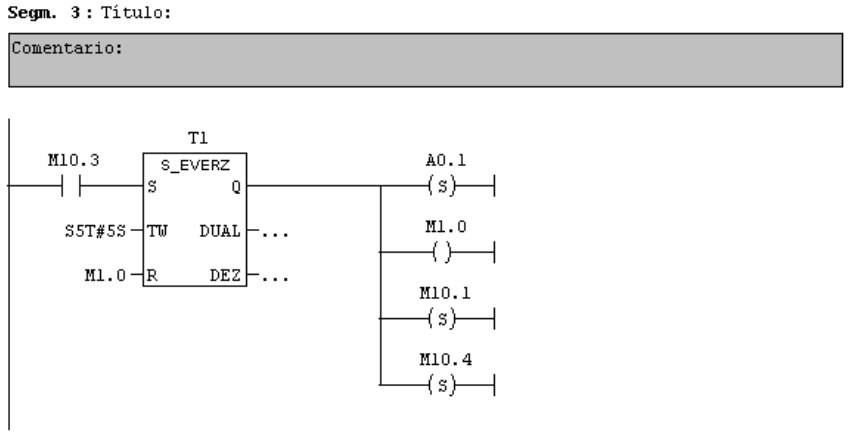

Ilustración 12: Tercer segmento de la programación.

La función de este segmento es apagar la luz verde A0.0 tras haber pasado los primeros 5 segundos, encender la luz amarilla de la primera baliza y el segundo temporizador. Se han utilizado los temporizadores de retardo a la conexión S_EVERZ, estos funcionan transmitiendo la señal, tan solo si la entrada se mantiene encendida el tiempo marcado por los parámetros que se le han configurado. A continuación, se añade una tabla explicativa de la función de cada una de sus patillas.

Parámetro	Tipo de dato	Descripción
S	BOOL	Entrada de arranque
TW	S5TIME	Valor de temporización
R	BOOL	Entrada de reseteo
Q	BOOL	Estado del temporizador
DUAL	WORD	Valor de temporización actual en binario
DEZ	WORD	Tiempo restante en decimal codificado en binario

Tabla 7: Parámetros de temporizador S_EVERZ.

Una vez que la luz verde se haya mantenido encendida 5 segundos, el temporizador activará la salida Q dando paso a cuatro bobinas, la primera A0.1 encenderá la luz amarilla de la primera baliza permanentemente, M1.0 reseteará el temporizador, M10.1 apagará la luz verde de la primera baliza y M10.4 activará el segundo temporizador.

· Segmento 4:

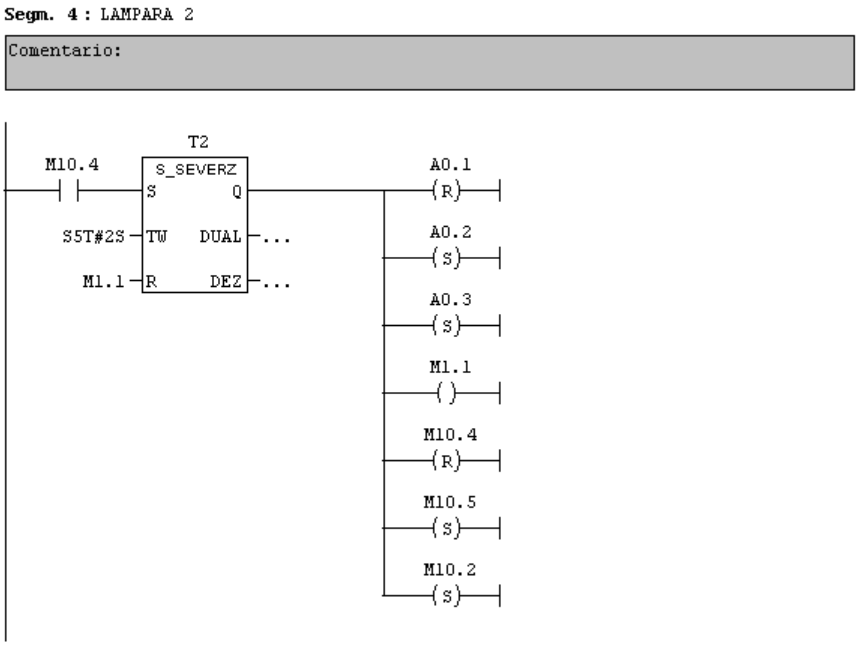

Ilustración 13: Cuarto segmento de la programación.

En este cuarto segmento se ha utilizado un temporizador de dos segundos durante los cuales se mantendrá encendida la luz amarilla, tras esto se activan las bobinas A0.1 para apagar la luz amarilla de la primera baliza, A0.2 para encender la luz roja de la primera baliza, A0.3 para encender la luz verde de la segunda baliza, M1.1 para resetear el temporizador T2, M10.4 para apagar el temporizador T2, 10.5 para activar el temporizador T3 y M10.2 para apagar la luz roja de la segunda baliza.

· **Segmento 5:**

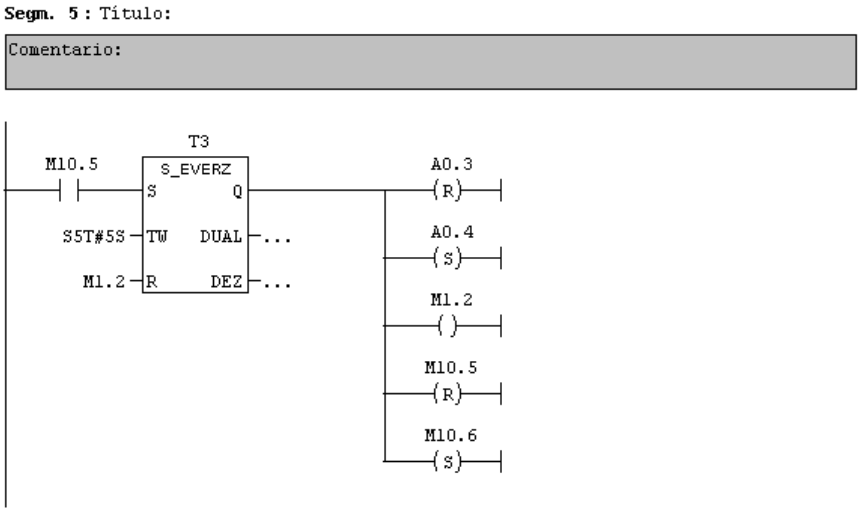

Ilustración 14: Quinto segmento de la programación.

Este segmento tiene la misma estructura que los anteriores y tras haber pasado 5 segundos activará las bobinas A0.3 para apagar la luz roja de la primera baliza, A0.4 para encender la luz amarilla de la segunda baliza, M1.2 para resetear el temporizador T3, M10.5 para apagar el temporizador T3 y M10.6 para activar el temporizador T4.

· **Segmento 6:**

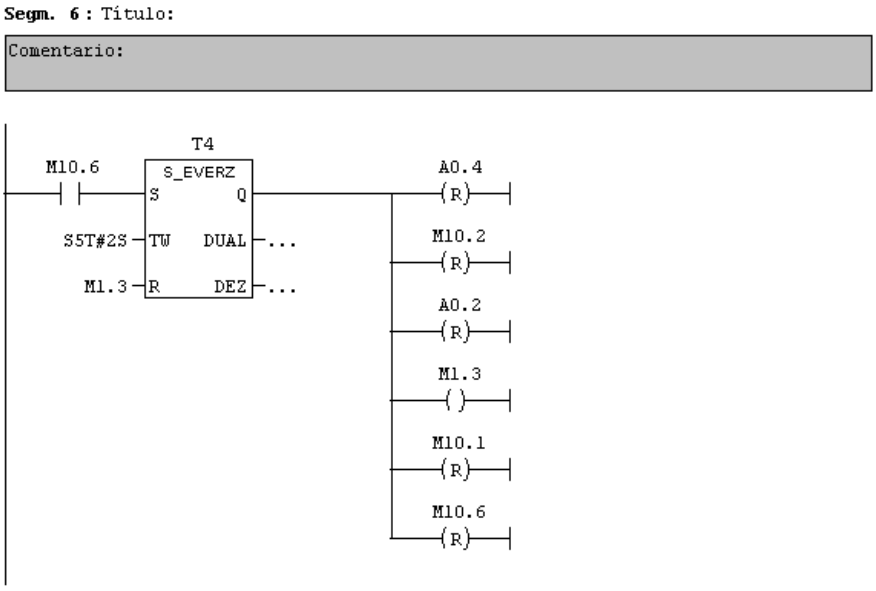

Ilustración 15: Sexto segmento de la programación.

Segmento final con la misma estructura que los anteriores, con un temporizador de dos segundos que activará las bobinas A0.4 para apagar la luz amarilla de la segunda baliza, M10.2 para encender la luz roja de la segunda baliza, A0.2 para apagar la luz roja de la primera baliza, M1.3 para resetear el temporizador T4, M10.1 para encender la luz verde de la primera baliza y M10.6 para apagar el temporizador T4.

Tras terminar este segmento las luces del sistema se encontrarán en la misma posición que al comenzarlo y el ciclo habrá terminado volviendo a los segmentos 1 y 2.

Se podría pensar en utilizar las mismas salidas A0.X para el control de los temporizadores, pero en ese caso el programa dejaría de funcionar al pasar el autómata a modo stop. Para que continúe funcionando se han utilizado marcas individuales, ya que estas se resetean en las transiciones a stop y el ciclo puede volver a empezar al pasar otra vez al modo run.

4. Vulnerabilidades

Los autómatas hoy en día se pueden encontrar en la mayoría de las industrias y centrales eléctricas cumpliendo funciones vitales para ellas, siendo capaces de trabajar en ambientes nocivos para el ser humano sin apenas mantenimiento, por ello rara vez se mantienen actualizados o se les realizan diagnósticos mientras cumplan su función correctamente. Esto puede incluso llegar a ser muy complicado según donde se encuentren, es necesario detener la tarea que estén realizando lo que por ejemplo en centrales nucleares, que deben mantenerse en funcionamiento la mayor parte de su vida útil para resultar rentables, llega a ser muy costoso. Además, como se comunican solo con dispositivos de su misma red se diseñaron con esta como su única línea de defensa, por tanto, un usuario malintencionado que se encuentre dentro puede causar daños catastróficos tantos humanos como económicos.

El aislamiento de los autómatas dentro de la red es importante pero no suficiente, a continuación, se demostrará cómo es posible localizar, identificar y atacar autómatas con suma facilidad una vez que se ha accedido a la red. Para ello se estudiarán en primer lugar las vulnerabilidades más importantes del autómata Simatic S7-300.

4.1. Protocolos abiertos

Como se ha visto la comunicación entre la estación de programación y el autómata utiliza la estructura tcp (protocolo de control de transmisión), este es el protocolo mediante el cual se transporta el tráfico de internet y, por lo tanto, carece de la encriptación de datos necesaria, ya que está orientado tan solo a la transmisión de datos entre dos puntos, pero no a su protección.

El hecho de que los paquetes de datos no estén encriptados permite captarlos, realizar ingeniería inversa y hacer modificaciones sobre los mismos, pudiendo derivar en un usuario malintencionado capaz de replicar funciones de la estación de programación como el apagado del autómata, desactivar la seguridad existente e incluso subir nuevos proyectos, entre otras. Esto tiene un daño potencial enorme, un atacante puede llegar a cambiar parámetros del programa haciendo que los dispositivos controlados por el autómata funcionen de manera errática y no esperada, lo que puede causar una catástrofe tanto económica como humana.

A continuación, se mostrará cómo se ha realizado un ataque de este tipo sobre el Simatic S7-300, bajo el escenario de un usuario conectado a la red donde se encuentran la estación de programación y el autómata. El primer paso es comprender que los datos en un comienzo no van a pasar por el ordenador atacante, de la estación de programación pasarán al switch y de ahí directamente al autómata, por tanto, habrá que conseguir que la estación envíe los datos a al atacante y que de ahí se envíen al autómata. En el caso de que la conexión se realice mediante un hub esto no sería necesario ya que se recibirían todos los paquetes, al conectarse mediante un switch se reciben tan solo los paquetes dirigidos al propio ordenador. Por tanto, para recibir todos los paquetes de la red será necesario realizar un ataque “Man In The Middle” (MITM), que consiste básicamente en hacer creer al emisor que el ordenador atacante es el receptor y al receptor que el atacante es el emisor, de esta forma se podrá interceptar toda la conversación sin que ninguna de las dos víctimas este al tanto. A continuación, se muestra una ilustración que muestra un esquema del ataque.

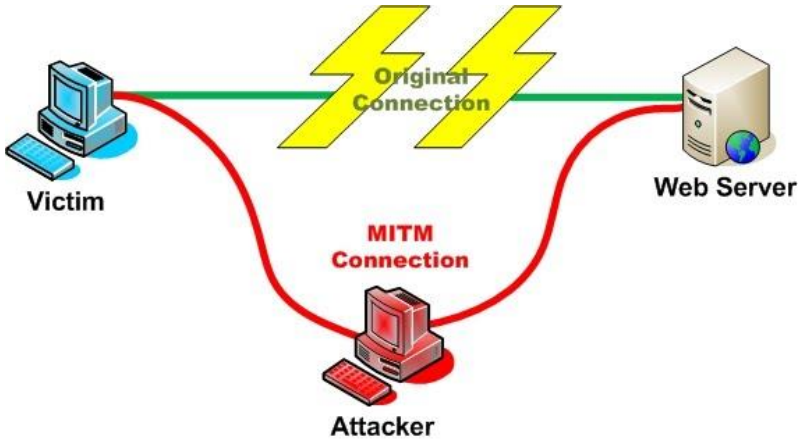

Ilustración 16: Ataque MITM.

Para realizar un ataque MITM la distribución Kali dispone de una herramienta eficaz y sencilla llamada Ettercap. Una vez abierta la aplicación solo se deben elegir las direcciones IP de las víctimas y el tipo de ataque que se desea hacer, en este caso se realizó un ARP poisoning, donde el ordenador atacante envía mensajes falsos a través del protocolo de resolución de direcciones (ARP) para que todo el tráfico se redirija a través de él. Como en el caso de este proyecto ya se conocen las direcciones IP ha sido muy sencillo realizar el ataque, en el caso de no conocerse Ettercap ofrece otras posibilidades como el MAC spoofing, en este caso se inunda la red con paquetes a direcciones aleatorias de forma que el switch se queda sin espacio para apuntar donde están y se ve obligado a enviar los paquetes por todos los puertos, de esta forma el switch se comporta como un hub y el atacante recibe todos los paquetes. A continuación, se muestra la ventana del programa mientras se está realizando el ataque.

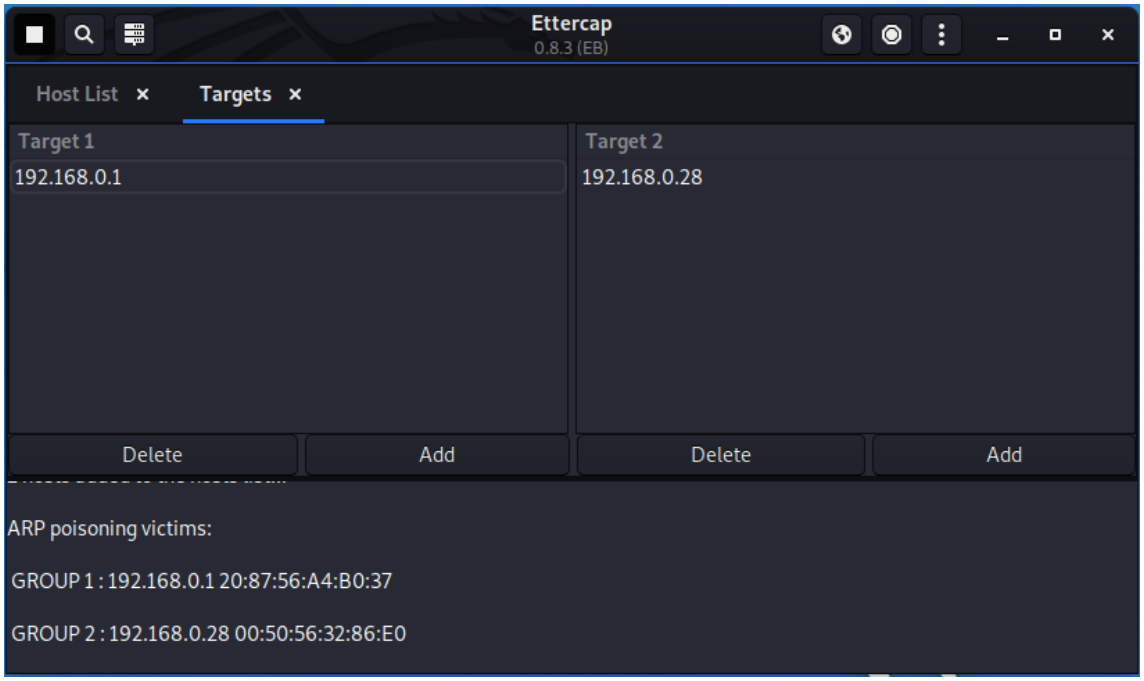

Ilustración 17: Software Ettercap mientras se realiza el ataque MITM.

Una vez iniciado el ataque todos los paquetes de datos que intercambien la estación y el autómata pasará por el atacante, por lo que este tan solo tendrá que

capturarlos mediante cualquier analizador de protocolos, en este caso se utilizó Wireshark que viene por defecto en la distribución Kali Linux.

Wireshark es un analizador de paquetes utilizado para el estudio de redes de comunicación que permite ver los paquetes de datos que intercambia un ordenador y es capaz además de filtrar dichos paquetes según los requerimientos del usuario. En este caso como se quería observar las conversaciones del autómata para obtener información sobre él, se aplicó el filtro “ip.addr==192.168.0.1”. De esta forma el programa muestra los paquetes enviados y recibidos por esa dirección IP.

Las últimas versiones de Wireshark, a partir de V2.1.1, admiten el protocolo S7 de Siemens lo que va a permitir el análisis de los paquetes de datos intercambiados por el PLC.

En el caso de este proyecto se realizaron las tres interacciones más habituales en un autómata de estas características, para observar que datos se podrían averiguar desde la perspectiva del atacante.

· La primera es la conexión inicial que se produce entre la estación y el autómata, y se obtuvo la siguiente lectura:

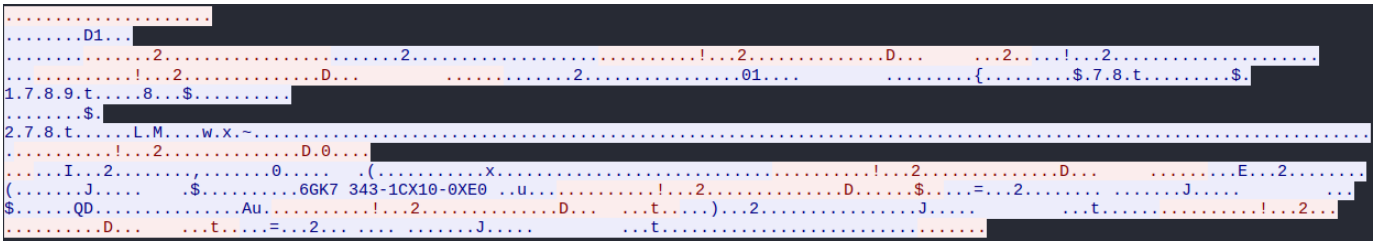

Ilustración 18: Conversación entre el PLC y la estación de programación inicial.

Ya en el inicio de la conexión se puede observar un dato muy valioso, el número de articulo del módulo de comunicaciones a partir del cual el atacante ya puede saber qué tipo de PLC es y el módulo de comunicaciones que utiliza.

· El siguiente fue el escenario en el que la estación da la orden de parada al autómata, dando la siguiente lectura:

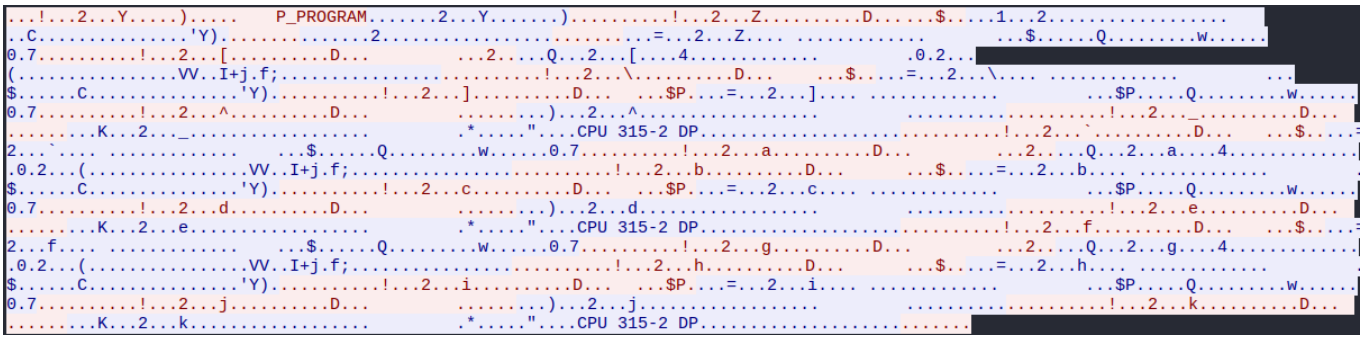

Ilustración 19: Orden de parada de la estación de programación al PLC.

En este caso se observa claramente el tipo de CPU que se está utilizando.

· A continuación, se subió una nueva programación al autómata obteniendo:

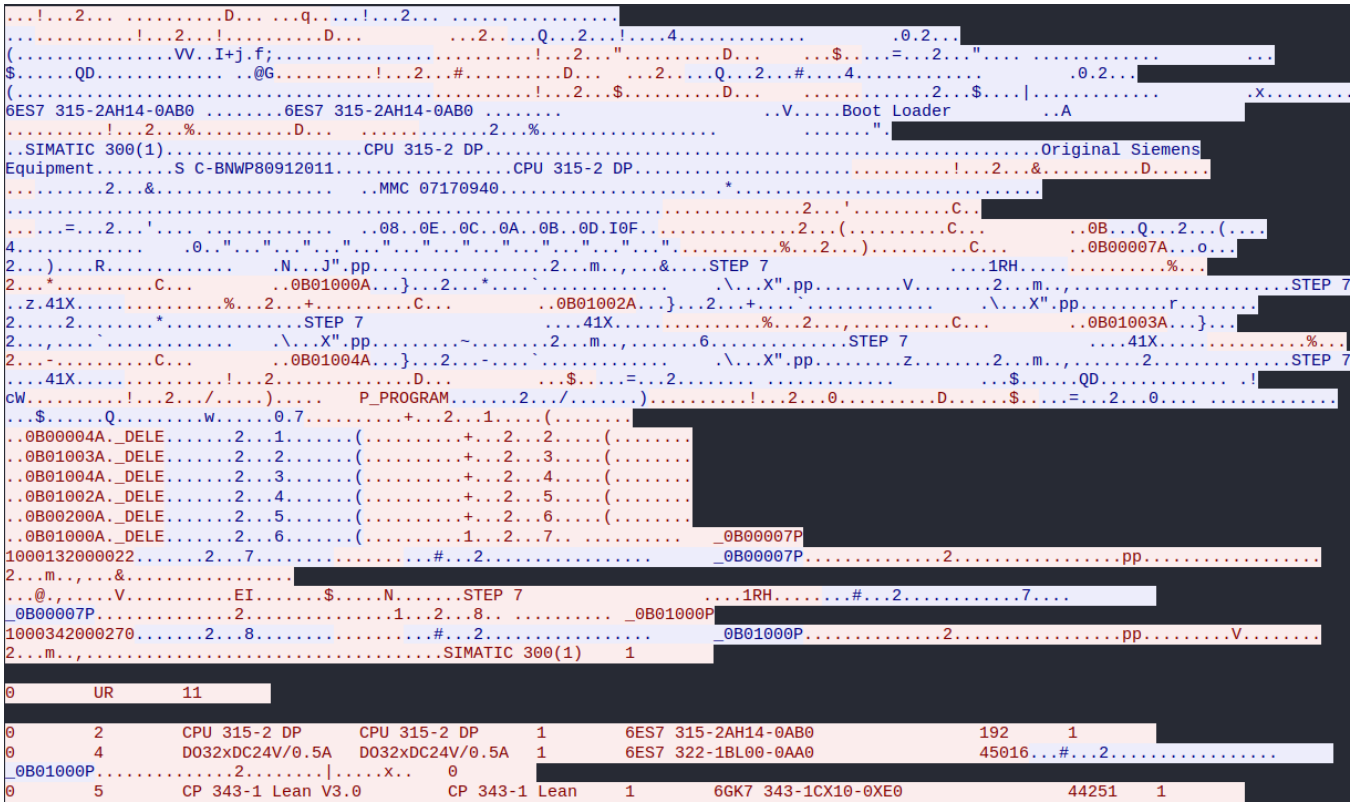

Ilustración 20: Subida de programación de la estación al PLC.

En este último escenario ya se observa toda la información obtenible sobre el autómata, es posible leer directamente Simatic 300, el tipo de CPU y módulo de comunicaciones, la versión a la que estan actualizados y sus números de articulo. Esta información es extremadamente sensible y proporciona al atacante información suficiente para realizar el ataque.

Como se ha visto es extremadamente sencillo obtener información del autómata pero no es lo más peligroso, es posible reenviar los paquetes obtenidos de la parte de la conversación del cliente, ocupar de esta forma el lugar de la estación de programación y con ello todas las posibilidades que ofrece esta, incluyendo el apagado del PLC o la subida de nuevos proyectos. Estos paquetes funcionarán además con cualquier otro autómata simatic 300 desactualizado. Puede incluso subirse codigo malicioso a la memoria del PLC que utilice su comunicación con la estación de programación para infectarla, desde donde podría atacarse el resto de la red resultando en un control absoluto del atacante sobre esta, con los consiguientes peligros economicos y humanos ya comentados.

4.2. Puerto 102 abierto

El puerto 102 de los autómatas se encuentra abierto y utiliza el protocolo ISO-TSAP, en este no se realiza encriptación alguna de los paquetes se envían, por ello es posible efectuar lo que se llama un Replay Attack. Este consiste en el ya comentado reenvío de paquetes capturados y modificados por el atacante. Una de las grandes ventajas de este ataque es que, en el caso de encontrarse un dispositivo de inspección

en la red, los paquetes normales y los modificados no tendrían ninguna diferencia entre sí, salvo su origen.

El protocolo ISO-TSAP que se utiliza no fue diseñado para ser un protocolo seguro y está abierto a análisis si sus paquetes son capturados. En el caso de no haber configurado ninguna contraseña en el dispositivo cualquier comando enviado por la estación de programación puede ser capturado, modificado y enviado de nuevo al PLC. En el caso de si estar configurada una contraseña en el PLC este sigue siendo vulnerable, la contraseña puede ser capturada igualmente por un atacante con acceso a la red o al PLC, ya que el protocolo que se utiliza sigue siendo totalmente abierto al análisis.

La realización de Replay Attacks sobre el protocolo ISO-TSAP no es visto por el ICS-CERT como una vulnerabilidad del protocolo, ya que este no fue nunca diseñado para proporcionar seguridad, sin embargo, los dispositivos que utilizan este protocolo son vulnerables a este tipo de ataques.

4.3. Contraseña por defecto

Las versiones más antiguas del firmware de los PLC Simatic tienen una contraseña por defecto que permite acceder al atacante a una ventana de comandos de diagnósticos internos del autómata. Esta ventana permite la realización de algunos diagnósticos internos del PLC y la extracción de contenido de la memoria. Fue rápidamente corregido por Siemens, pero aún sigue presente en los PLCs desactualizados de la tabla mostrada a continuación.

PLC	Versiones vulnerables	Arreglado en
CPU315(incluyendo F)-2PN/DP	V2.6 y anteriores	V3.1
CPU317(incluyendo F)-2PN/DP	V2.6 y anteriores	V3.1
CPU319(incluyendo F)-3PN/DP	V2.7 y anteriores	V2.8
IM151-8(incluyendo F) PN/DP CPU	V2.7	V3.2
M154-8 PN/DP CPU	V2.5	V3.2
S7-400 - Todos los modelos	No son vulnerables	

Tabla 8: CPUs vulnerables a la contraseña por defecto.

En este proyecto se cuenta con un CPU 315-2 DP V3.3 y por tanto este error ya ha sido corregido y no ha sido posible comprobar su funcionamiento. El usuario y contraseña por defecto en los PLCs vulnerables es basisk/basisk y existe un exploit, creado por D. Beresford, que viene por defecto en el software Metasploit capaz de automatizar este ataque.

5. Explotación

En este apartado se verá como una vez se ha accedido a la red es extremadamente sencillo localizar y atacar autómatas desactualizados. El primer paso consistirá en realizar un estudio de la red para saber que dispositivos están conectados a ella, tras esto se localizará el autómata averiguando sus características y finalmente se realizará un ataque de negación de servicio.

5.1. Estudio de la red

Para realizar el estudio de la red Kali Linux dispone de una herramienta llamada Zenmap que es capaz de detectar equipos, servicios, sistemas operativos y rastrear sus puertos. En la red local creada para este proyecto se dispone del autómata, de la estación de programación y del ordenador atacante, como se está ocupando el lugar del atacante es sencillo averiguar su dirección IP mediante el código ifconfig que se muestra a continuación.

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.24  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe8c:74f  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:8c:07:4f  txqueuelen 1000  (Ethernet)
        RX packets 709  bytes 148197 (144.7 KiB)
        RX errors 0  dropped 2  overruns 0  frame 0
        TX packets 484  bytes 61914 (60.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 204640  bytes 32095048 (30.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 204640  bytes 32095048 (30.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Ilustración 21: Respuesta al comando ifconfig.

Una vez se ha conocido la dirección IP del atacante, que es 192.168.0.24, se ha realizado un escaneo mediante Zenmap. Los parámetros del escaneo se han configurado de forma que el objetivo sean las direcciones IP entre la 192.168.0.1 y 192.168.0.255, y el perfil sea un ping scan, cuyo objetivo es la identificación de dispositivos activos en la red. Este escaneo no es demasiado intrusivo y se suele utilizar para obtener un conocimiento liviano de la red, como cuantos dispositivos se encuentran activos, sus direcciones IP, sus direcciones MAC y el fabricante de los mismos. A continuación, se muestra una captura de el escaneo una vez realizado con los resultados obtenidos.

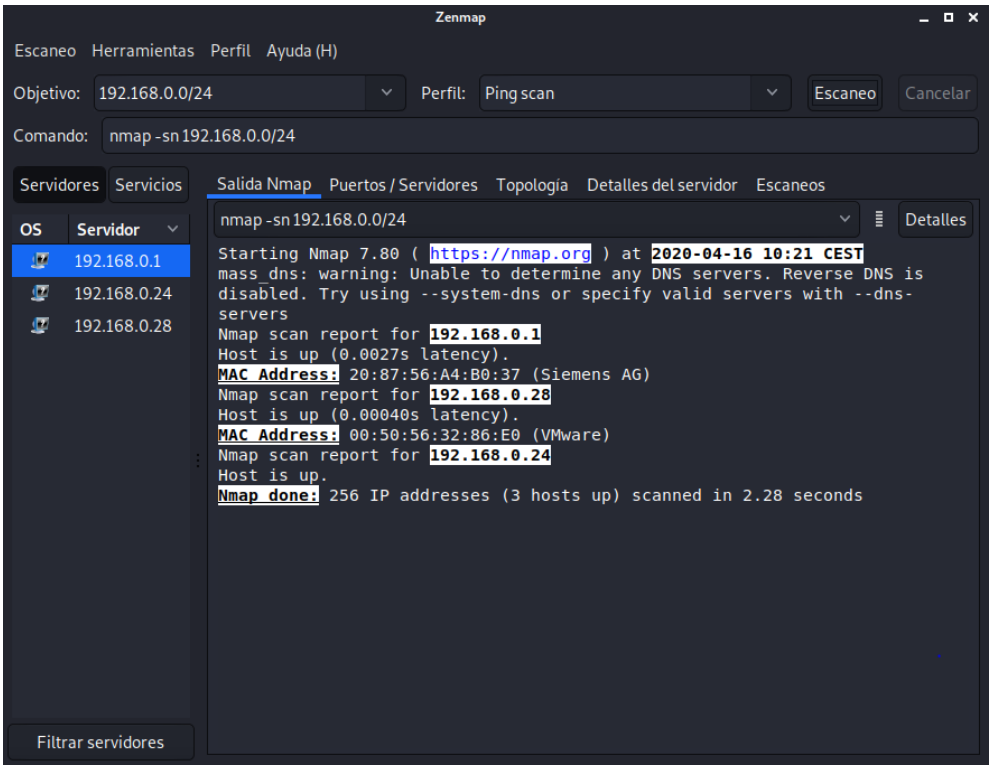

Ilustración 22: Escáner de la red mediante Zenmap.

En el caso de este proyecto ya se conocían los dispositivos conectados y por tanto el resultado ha sido el esperado, un dispositivo conectado en la dirección 192.168.0.1 cuyo fabricante es Siemens AG y otro en a la dirección 192.168.0.24 con el fabricante como VMware, que es el programa que se ha utilizado para el uso de la máquina virtual que actúa como estación de programación.

Una vez descubierto un dispositivo de Siemens en la red se podría iniciar el ataque MITM de la manera explicada ya anteriormente, o para no depender de que un ingeniero se comunique con el autómata, que puede llegar a pasar mucho tiempo, existe una herramienta de estudio y explotación de redes industriales que es capaz de detectar dispositivos del fabricante Siemens, aportar al atacante información crítica sobre ellos y realizar ataques de negación de servicio sobre los autómatas S7-300.

Esta herramienta está escrita en Python y tiene una interfaz similar a Metasploit, como se puede observar a continuación.

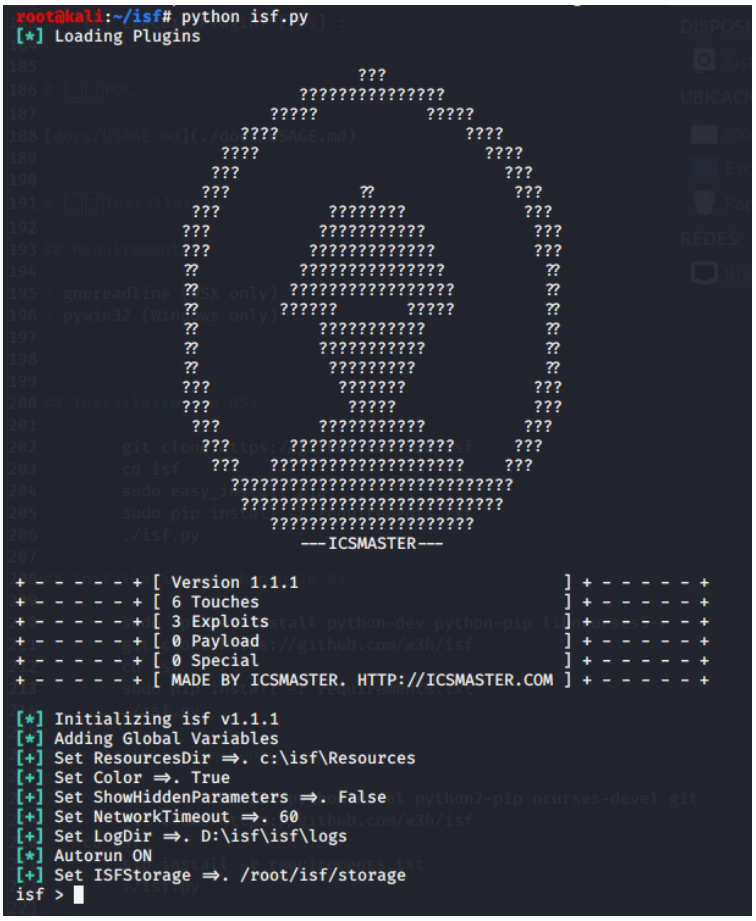

Ilustración 23: Portada de la aplicación de explotación de PLCs.

Una vez instalada la aplicación ya es posible acceder y activar los exploits que en ella se ofrecen, en el caso de este proyecto es especialmente interesante el escáner de PLCs. Para utilizarlo simplemente hay que utilizar el comando “use plcscan” y configurar la dirección IP del autómata a estudiar, 192.168.0.1 en el caso de este proyecto, y el tipo de PLC, en este caso S7. Una vez configurado quedan las siguientes opciones, las que no se han mencionado se han dejado en sus valores por defecto.

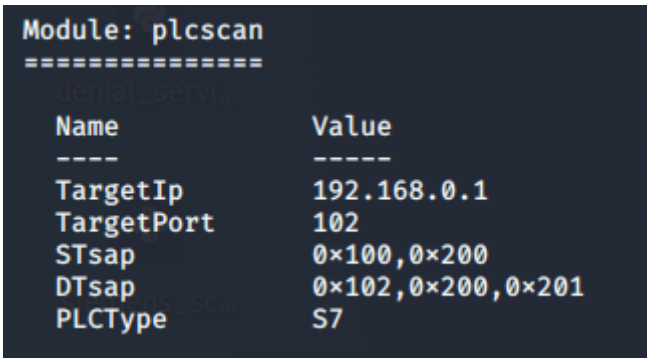

Ilustración 24: Configuración del exploit de escaneo de PLCs.

Tras ejecutarlo se obtiene la siguiente información.

```
192.168.0.1:102 S7comm (src_tsap=0×100, dst_tsap=0×102)
  Module                 : 6ES7 315-2AH14-0AB0  v.0.4        (364553372033331352d32414831342d304142302000c000040001)
  Basic Hardware         : 6ES7 315-2AH14-0AB0  v.0.4        (364553372033331352d32414831342d304142302000c000040001)
  Basic Firmware         :                      v.3.3.2      (202020202020202020202020202020202020202000c056030302)
  Unknown (129)          : Boot Loader           A           (426f6f74204c6f6164657220202020202020202000041200909)
  Name of the PLC        : SIMATIC 300(1)                    (53494d4154494320333030283129000000000000000000000000000000000000)
  Name of the module     : CPU 315-2 DP                      (435055203331352d322044500000000000000000000000000000000000000000)
  Plant identification   :                                   (000000000000000000000000000000000000000000000000000000000000000000)
  Copyright              : Original Siemens Equipment        (4f726967696e616c205369656d656e73204571756970d656e74000000000000)
  Serial number of module : S C-BNWP80912011                 (5320432d424e575038303931323031310000000000000000000000000000000000)
  Module type name       : CPU 315-2 DP                      (435055203331352d322044500000000000000000000000000000000000000000)
Scan complete
[+] plcscan Succeeded
```

Ilustración 25: Respuesta del exploit de escaneo de PLCs.

Como se puede observar mediante este exploit se obtiene información extremadamente sensible y el atacante ya conocería toda la información necesaria para realizar su ataque. Con la versión de firmware instalada ya es posible averiguar qué vulnerabilidades se encuentran activas en el autómata y que ataques serán efectivos.

5.2. Ataque de negación de servicio

La herramienta isf además de escanear la red en busca de PLCs permite la realización de un ataque de negación de servicio sobre los autómatas Simatic S7-300 y se ha comprobado su efectividad sobre la versión 3.3.2, instalada en el PLC de este proyecto.

Para acceder a este exploit se introduce el comando "use Siemens_300_400_CPU_Control" y se introduce la configuración deseada según la siguiente tabla.

Opción	Valor	Descripción
TargetIp	192.168.0.1	Dirección IP de la víctima
TargetPort	102	Puerto por el que se realizará el ataque
Slot	0	Ataque sobre el módulo de comunicaciones
	2	Ataque sobre la CPU
Command	stop	Apagado
	cold	Encendido en frío
	hot	Encendido en caliente

Ilustración 26: Posibles configuraciones del exploit de apagado del S7-300.

Se muestra ahora la configuración elegida para realizar un ataque de negación de servicio sobre la CPU.

```
Module: Siemens_300_400_CPU_Control
===================================

  Name         Value
  ----         -----
  TargetIp     192.168.0.1
  TargetPort   102
  Slot         2
  Command      stop
```

Ilustración 27: Configuración elegida para apagar el S7-300.

Una vez realizado el ataque la CPU es puesta en modo stop y deja de controlar el sistema industrial que tiene encomendado, para que vuelva a funcionar será

necesario un rearranque manual o desde el software de programación. Esto puede parecer una tarea trivial en el laboratorio, pero en un ambiente industrial donde puede haber cientos de autómatas es una tarea muy compleja, pueden llegar a ser muy difíciles de identificar, encontrar y en ocasiones imposibles de acceder manualmente, ya que se pueden encontrar dentro de otros dispositivos o en ambiente poco adecuados para el ser humano.

Como se ha visto este ataque puede llegar a ser devastador si se aplica al objetivo adecuado, pero se puede aumentar su efecto si se cambiaran las direcciones IP de los autómatas y su nombre, de esta forma los ingenieros serían incapaces de identificar los PLCs en la red y tendrían que comprobar manualmente la identidad de cada uno de ellos. Incluso podría darse la situación de un volcado de programación sobre un PLC erróneo, lo que podría causar un accidente grave.

Esta opción nos la ofrece la siguiente herramienta que se estudiará en este proyecto, se trata de un script escrito en Python y creado por el experto en ciberseguridad industrial Tijl Deneut capaz de realizar escáneres de la red en busca de PLCs, averiguar información crítica sobre ellos y en el caso de encontrar un S7-1200 es capaz de leer sus salidas e incluso cambiarlas. Esto último no ha sido posible comprobarlo, pero lo interesante de este programa es que es capaz de cambiar el nombre y dirección IP del autómata de este proyecto. A continuación, se muestra la portada del script y la selección de la red por la que se realizará el escaneo.

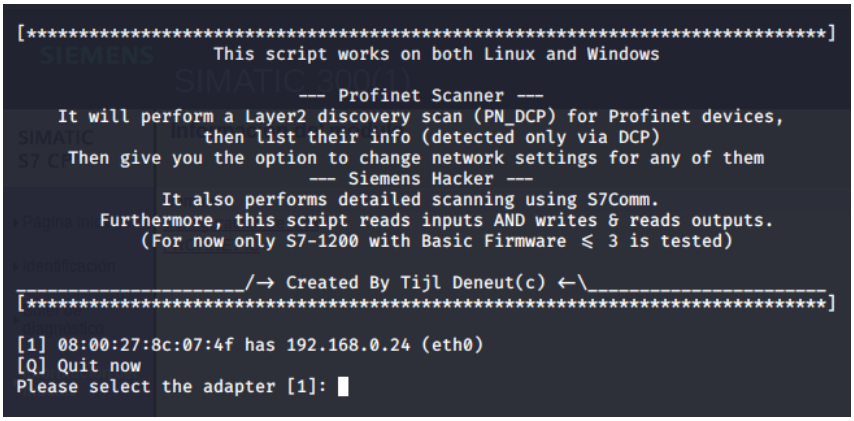

Ilustración 28: Portada del segundo script de explotación de PLCs.

Una vez realizado el escaneo se muestran los PLCs encontrados y se elige el que se desea atacar, como se muestra ahora.

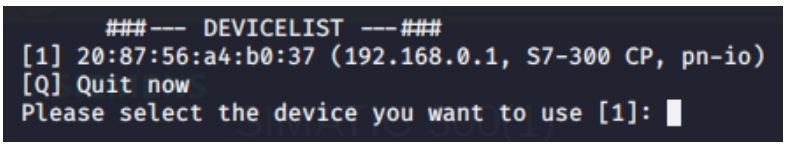

Ilustración 29: Elección del PLC que se desea atacar.

Una vez elegida la víctima el script muestra los posibles ataques que es capaz de realizar.

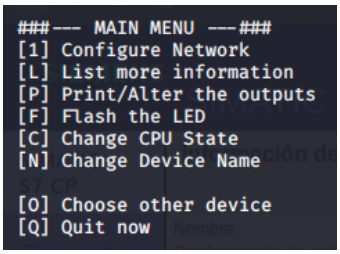

Ilustración 30: Posibles ataques sobre el PLC.

Se ha comprobado el funcionamiento de todos ellos sobre el PLC de este proyecto y se han obtenido los siguientes resultados:

· **Configure Network:** Este ataque cambia la dirección IP del autómata y se ha realizado con éxito.

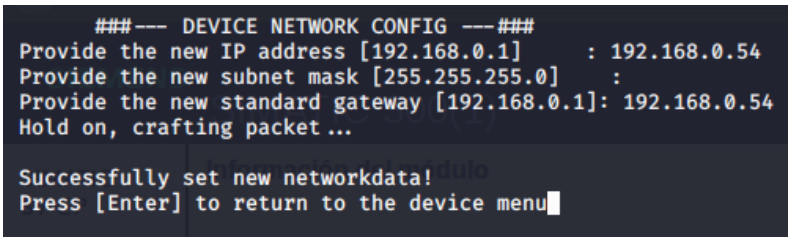

Ilustración 31: Cambio de la dirección IP del PLC.

· **List more information:** En este caso el script mostrará la información que es capaz de obtener sobre la víctima. Se ha realizado correctamente mostrando la siguiente información.

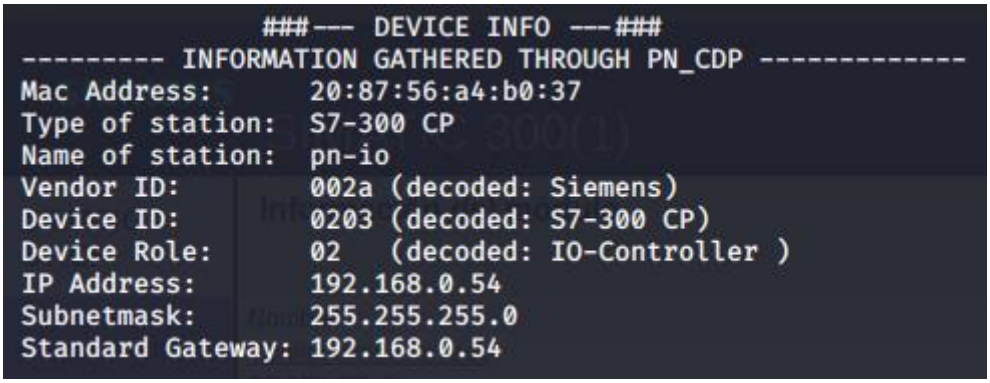

Ilustración 32: Información obtenible sobre el PLC.

Se puede observar que aporta información muy crítica sobre el PLC objetivo, pero no tanta como la herramienta vista anteriormente, no se obtiene la versión del firmware, muy importante en la planificación del ataque.

· **Print/Alter the outputs:** Este ataque muestra las salidas del autómata y permite cambiarlas. Según el autor funciona en los autómatas S7-1200, en el S7-300 de este proyecto no lo ha hecho.

· **Flash the LED:** Se realiza un parpadeo de los leds del autómata durante un tiempo determinado, esta función es muy útil para identificar a los autómatas, pero no tiene utilidad para los objetivos de este proyecto. Se ha realizado con éxito.

· **Change CPU State:** Ataque de negación de servicio sobre la víctima, no ha funcionado sobre el S7-300.

· **Change Device Name:** Sirve para cambiar el nombre del dispositivo, se ha realizado con éxito.

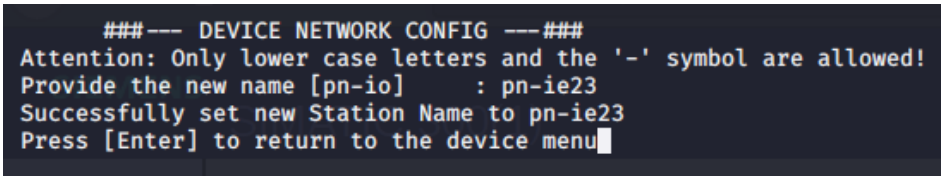

Ilustración 33: Cambio del nombre del PLC.

Tras la realización de los ataques es posible comprobar su funcionamiento desde la estación de programación, los resultados se muestran a continuación.

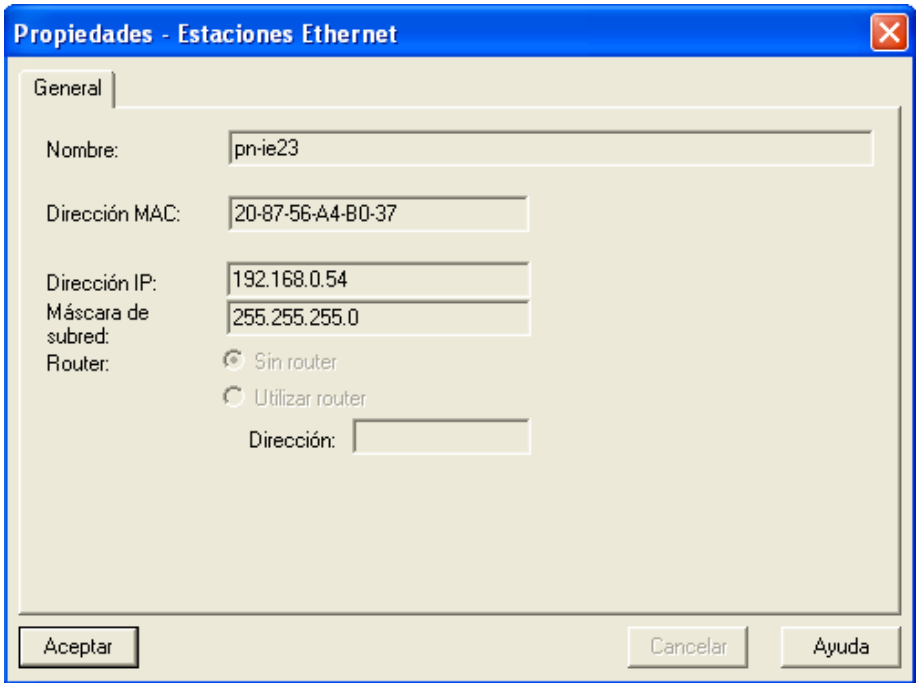

Ilustración 34: Cambios efectuados vistos desde la estación de programación.

Como se observa en la ilustración tanto el nombre como la dirección IP del dispositivo ha cambiado sin el conocimiento de los ingenieros encargados, en el caso de contar con tan solo un PLC es sencillo de averiguar y arreglar, pero en una red de cientos de PLCs donde se intercambien sus nombres y direcciones IP es fácil darse cuenta de que recuperar el control de la red se torna en una tarea titánica, y más cuando tras ello se realiza otro ataque de negación de servicio apagando todos los dispositivos. Es posible además automatizar este ataque de forma que se realice masivamente en cuestión de segundos. El tiempo que se tardaría en restablecer el funcionamiento puede llegar a ser extremadamente costoso para muchas industrias.

Los PLCs S7-300 es habitual encontrarlos controlando sistemas en centrales nucleares, estas centrales para llegar a ser rentables deben permanecen en funcionamiento casi la totalidad de su vida útil, un ataque de estas características a parte del accidente que podría llegar a causar sería devastador económicamente.

6. Autómatas sin protección de red

Como ya se ha comentado la mayor protección de los PLCs en la actualidad es la propia red, sin esta los PLCs son extremadamente vulnerables a ataques y pueden ser incluso el punto de partida de una infección de la estación de programación. Ya sea por error humano o por desconocimiento es posible encontrar PLCs conectados libremente a internet sin protección alguna, que pueden ser encontrados y posteriormente explotados con suma facilidad, con los peligros y riesgos que eso supone.

Para búsquedas de dispositivos conectador a internet es muy útil la herramienta Shodan, esta es un motor de búsqueda que rastrea la red en busca de equipos de acceso público conectados a internet, obteniendo además información sensible sobre ellos como el software que utilizan, los puertos abiertos de los que disponen y los servicios que estos ofrecen.

Entre los muchos dispositivos que se pueden localizar mediante Shodan destacan los sistemas industriales, ya en la pagina principal ofrece la realización de una búsqueda sobre estos. En el caso de este proyecto es interesante la búsqueda de autómatas Siemens conectados libremente a internet, a continuación se muestran los resultados obtenidos.

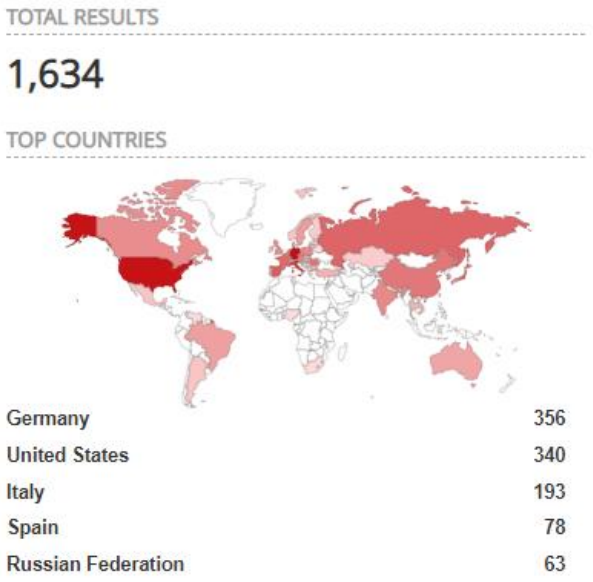

Ilustración 35: Autómatas Siemens conectados a internet en el mundo.

Para obtener estos resultados se ha introducido en el buscador “port: 102 Siemens”, para que encontrará los autómatas Siemens con el puerto 102 abierto, que es por donde se realizan los ataques. Se han obtenido por tanto 1634 autómatas Siemens potencialmente explotables alrededor del mundo, encabezando la lista Alemania y Estados Unidos y con presencia la gran mayoría de países desarrollados. Posteriormente se muestra un ejemplo de un autómata conectado a internet en Francia sin protección de red y que información es capaz de revelar Shodan acerca de él.

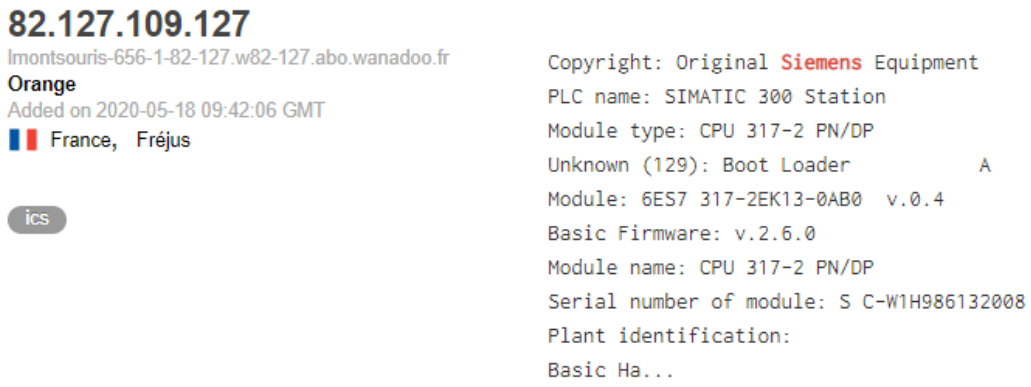

Ilustración 36: Información obtenible sobre un autómata mediante Shodan.

Shodan muestra directamente información muy sensible y suficiente para planear un ataque efectivo, se obtiene directamente el tipo de CPU con la versión de firmware que utiliza, dirección IP, el tipo de módulo, la empresa a la que pertenece y la dirección física donde se encuentra. Es posible incluso encontrar autómatas con el servidor Web sin protección y abierto a cualquier usuario que quiere acceder.

Si se reduce el parámetro de búsqueda a los autómatas que se encuentren en España se obtiene el siguiente resultado.

Ilustración 37: Autómatas Siemens conectados a internet en España.

Hay 78 sistemas industriales potencialmente explotables en España, abiertos a cualquier usuario malintencionado del mundo.

7. Estrategias defensivas

Ninguna estrategia defensiva es capaz de proteger un sistema industrial con un 100% de efectividad, pero no por ello hay que desestimar la inversión en seguridad informática. Aplicando las estrategias defensivas adecuadas es posible reducir drásticamente el riesgo a un ataque o si este se produce reducir su efecto o gravedad. A continuación, se proponen una serie de estrategias defensivas que toda empresa con un ICS debería emplear.

· La principal estrategia es actualizar siempre los dispositivos con las últimas versiones proporcionadas por Siemens, en el caso de no ser posible por la naturaleza del proceso que están controlando una solución temporal podría ser la adopción de una estrategia de defensa en profundidad. Esta consiste en la segmentación de la red mediante la colocación de barreras que aíslen unas partes de otras, de esta manera para acceder al ICS el atacante deberá traspasar un número determinado de líneas defensivas, dando tiempo para la detección y respuesta. Básicamente se trataría de aislar lo máximo posible el ICS dentro de la red.

· Establecer en la empresa una política estricta de administración de cuentas, que asegure el uso de contraseñas robustas y elimine las cuentas no utilizadas o que vengan por defecto en los dispositivos.

· Implantar un sistema de detección de intrusiones capaz de monitorizar todo el tráfico de la red detectando aquel que no esté autorizado, en especial el que circule por el puerto 102/TCP de los dispositivos S7, el que se mande fuera de la red industrial y el que circule entre estaciones de control.

· Es importante también el uso de firewalls que administren el tráfico que entre y salga de la red industrial, permitiendo el aislamiento de la red industrial respecto a la del resto de la empresa. Es recomendable que tan solo puedan comunicar con estas direcciones MAC conocidas, para lo que habrá que confeccionar una lista blanca de direcciones que incluya tan solo los dispositivos necesarios. Además, se puede mejorar la seguridad mediante el bloqueo del tráfico de protocolos no deseados, como el telnet o el http dentro de la red industrial.

·Prevenir los ataques de ingeniería social mediante la formación de todo el personal en seguridad informática, aunque su función no esté directamente relacionada con el sistema industrial, evitando posibles ataques de usurpación de identidad (phishing) o estafas.

8. Conclusión

En este proyecto se ha demostrado como es extremadamente sencillo estudiar redes y atacar autómatas S7-300 desactualizados. Las consecuencias de un ataque de estas características a un sistema industrial pueden ser enormes para una empresa, por ello es importante la inversión en seguridad informática de las redes industriales.

La ciberseguridad es un factor esencial de las industrias del futuro y del presente, actualmente es extremadamente sencillo atacar redes industriales y por tanto hay que prestar especial atención a la protección y actualización de estas.

Los autómatas van a estar cada día más presentes en nuestra vida tomando el control de sistemas más y más críticos e importantes en la industria, y en un futuro también de nuestra sociedad, por ello el conjunto de esta debe ser instruida en esta materia. La mayor fuente de vulnerabilidades de un sistema es el desconocimiento de sus usuarios.

Es común la comparación de la ciberseguridad con una cadena, esta es tan fuerte como lo es el más débil de sus eslabones, cuando es forzada romperá por el más débil por ello es importante la fortaleza de cada uno de ellos. Volviendo a la ciberseguridad el eslabón más débil es aquella persona que tiene conocimiento nulo sobre informática y por tanto es por donde el atacante intentará acceder a la red. Es tan importante la creación de sistemas robustos y seguros como la formación de cada una de las personas.

9. Bibliografía

[1] S. Ag, “CPs S7 para Industrial Ethernet Configurar y poner en servicio,” pp. 1–320, 2008.

[2] N. Ben Aloui, “Industrial Control Systems Dynamic Code Injection,” *Cybersecurity Labs – DCNS Toulon*, 2016.

[3] D. Beresford, “Exploiting Siemens Simatic S7 PLCs,” *Black Hat USA*, pp. 1–26, 2011, doi: 10.1016/j.braindev.2013.07.010.

[4] L. Cheng, L. Donghong, and M. Liang, “The spear to break the security wall of S7CommPlus,” *Defcon 25*, 2017, [Online]. Available: https://media.defcon.org/DEF CON 25/DEF CON 25 presentations/Cheng Lei/DEFCON-25-Cheng-Lei-The-Spear-to-Break-the-Security-Wall-of-S7CommPlus-WP.pdf.

[5] C. P. Lean, “S7-300 - Industrial Ethernet / PROFINET,” pp. 1–70, 2018.

[6] Siemens, “técnicos S7-300 CPU 31xC y CPU 31x : Datos técnicos,” p. 41, 2011, [Online]. Available: http://w3.siemens.com/mcms/programmable-logic-controller/en/advanced-controller/s7-300/pages/default.aspx.

[7] “Siemens SIMATIC PLCs Reported Issues Summary (Update A) | CISA.” https://www.us-cert.gov/ics/advisories/ICSA-11-223-01A (accessed May 06, 2020).

[8] L. V. Vite Constante, “Hacking ético en dispositivos plc de control industrial conectados a red,” 2017, [Online]. Available: http://repositorio.uta.edu.ec/handle/123456789/26670.