Universitat Oberta de Catalunya

uoc.edu

Estudio y desarrollo de un enfoque de Pentesting para Sistemas de Control Industrial (ICS)

Eder Pérez Ignacio
Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones
Seguridad en la Internet de las Cosas

Consultor: Carlos Hernández Gañán
Consultor colaborador: Xabier Larrucea Uriarte (Tecnalia)
Profesora responsable de la asignatura: Helena Rifà Pous

06/2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Estudio y desarrollo de un enfoque de Pentesting para Sistemas de Control Industrial (ICS)*
Nombre del autor:	*Eder Pérez Ignacio*
Nombre del consultor/a:	*Carlos Hernández Gañán (UOC)* *Xabier Larrucea Uriarte (Tecnalia)*
Nombre del PRA:	*Helena Rifà Pous*
Fecha de entrega (mm/aaaa):	06/2019
Titulación:	*Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones*
Área del Trabajo Final:	*Seguridad en la Internet de las Cosas*
Idioma del trabajo:	*Castellano*
Palabras clave	*pentesting, ICS, metodología*

Resumen del Trabajo (máximo 250 palabras):

Los Sistemas de Control Industrial forman parte de las infraestructuras críticas de un país, las cuáles rigen y sostienen el funcionamiento del mismo: agua, sanidad, industria farmacéutica, energía... son algunos ejemplos.

Estas infraestructuras críticas necesitan de un control de mandos, y es ahí cuando entran en juego los Sistemas de Control Industrial (ICS), que permiten administrar y regular el comportamiento de los Sistemas Industriales que las conforman. Estos Sistemas Industriales son el objetivo de ataques cibernéticos continuos, como ha sido el caso de Ucrania en 2015 que afectó a 225.000 usuarios de la red eléctrica; o el tan conocido mediáticamente caso de Stuxnet, bautizado como el primer arma de ciberguerra que afectó a la planta nuclear de Natanz en Irán.

Las consecuencias sobre este tipo de sistemas pueden ser desde pérdidas económicas desmesuradas hasta lo que es peor: poner en riesgo la vida de los ciudadanos de un país.

Por este motivo, resulta de vital importancia someter a auditorías de seguridad a los Sistemas de Control Industrial. Con este Trabajo de Fin de Máster se pretende realizar un estudio y elaborar una metodología o enfoque que sirva para realizar auditorías de seguridad o tests de penetración (pentest) sobre este tipo de sistemas.

A su vez, se pondrá en práctica el enfoque desarrollado mediante la realización de una PoC (*Proof of Concept)* sobre el honeypot de baja interacción Conpot.

Abstract (in English, 250 words or less):

Industrial Control Systems are part of the critical infrastructures of a country, which govern and sustain the proper operation of it: wáter, sanitation, pharmaceutical industry, energy… are some examples.

These critical infrastructures need a main controller, and that is when the Industrial Control Systems (ICS) come into play: they allow to manage and regulate the behaviour of the Industrial Systems that define them. These Industrial Systems are the target of continuous cyber attacks, as has been the case in Ukraine in 2015 that affected 225,000 users of the power grid; or the well-known media case of Stuxnet, also known as the first cyberwar weapon that targeted the Natanz's nuclear plant in Iran.

The consequences on this type of systems can range from excessive economic losses to what is worse: they can put the lifes of the citizens of a country at risk.

For this reason, it is vitally important to perform security audits on Industrial Control Systems. The objective of this Master's Thesis is to conduct a study and develop a methodology to perform security audits or penetration tests (pentest) on this type of systems.

At the same time, the approach developed through the implementation of a PoC (*Proof of Concept*) on the low interaction honeypot Conpot will be put into practice.

Tabla de contenido

Lista de figuras

Lista de tablas

1. Introducción

El siguiente epígrafe corresponde al planteamiento inicial del proyecto. En él se describen la justificación del proyecto, los objetivos que se pretenden alcanzar, el enfoque y método a seguir, el conjunto de tareas a desarrollar, la planificación temporal de las mismas, las herramientas de las que se hará uso, etc.

1.1 Contexto y justificación del Trabajo

Los Sistemas de Control Industrial forman parte de las infraestructuras críticas de un país. Una infraestructura crítica es aquella que desempeña una tarea esencial para el correcto funcionamiento de un país. La Ley PIC (Protección de Infraestructuras Críticas) 8/2011 [1] las define como “aquellas cuyo funcionamiento es indispensable y no permite soluciones alternativas, por lo que su perturbación o destrucción tendría un grave impacto sobre los servicios esenciales”; siendo estos servicios esenciales aquellos necesarios para el mantenimiento de las funciones sociales básicas, la salud, la seguridad, el bienestar social y económico de los ciudadanos, o el eficaz funcionamiento de las Instituciones del Estado y las Administraciones Públicas.

Actualmente, en nuestra sociedad, no existe actividad humana que no sea dependiente de los servicios esenciales prestados por las infraestructuras críticas. La energía, el agua, la industria nuclear, etc. son algunos ejemplos de estos servicios. Así, resumidamente y a modo introductorio, un Sistema de Control Industrial englobará a todos los dispositivos que permitan administrar y regular el comportamiento de los sistemas industriales que conforman estas infraestructuras críticas.

Estos Sistemas de Control Industrial son objetivo de ataques cibernéticos como ha sido el caso de Ucrania en 2015, que afectó a 225.000 usuarios de la red eléctrica, y el de 2016, en el que se vieron afectadas en torno a una quinta parte de la población de Kiev [2]; o el tan conocido mediáticamente caso de Stuxnet, bautizado como el primer arma de ciberguerra que afectó a la planta nuclear de Natanz en Irán.

Los ataques a estas infraestructuras críticas suponen un riesgo devastador para las empresas y ciudadanos de un país y son, por ello, un objetivo atractivo para los ciberatacantes. No sólo las pérdidas económicas pueden ser desmesuradas; está en riesgo la seguridad física de los ciudadanos.

Un caso posible en el que se puede ver la gravedad de las consecuencias sería un ataque terrorista coordinado en el que se compromete una planta eléctrica dejando sin luz a la ciudad realizando un ataque simultáneo por las calles provocando el caos. Sin hablar de un corte eléctrico prolongado mediante, por ejemplo, el cifrado de toda la información de estos sistemas mediante un ransomware: apagón masivo, semáforos sin funcionamiento, sin acceso a la red telefónica ni a Internet serían algunas de las fatales consecuencias que desatarían el caos.

Por ello, resulta de vital importancia investigar y desarrollar mecanismos que ayuden a la mitigación de los riesgos asociados a los Sistemas de Control Industrial. Esto se llevará a cabo a través del desarrollo de una metodología de ataque mediante un test de penetración o *pentest* que permita evaluar las vulnerabilidades u otros fallos de seguridad y que posibilite posteriormente prevenir futuros ataques por medio de la corrección de los mismos en estos entornos.

Para realizar el estudio y el desarrollo del enfoque de pentesting para ICS se construirá un laboratorio ya que no se dispone de un entorno real donde poder realizar las pruebas. Dicho laboratorio estará formado por una red aislada de dos máquinas: una máquina virtual, que constituirá el ordenador atacante desde donde será realizada la prueba de penetración; y una segunda máquina en un contenedor que simulará un componente de un Sistema de Control Industrial. Para ello se hará uso del proyecto Conpot [3], un honeypot de baja interacción y de código abierto que puede simular diferentes tipos de componentes de un ICS de forma realista. Por lo tanto, se pretende también poner en práctica el enfoque de pentesting en este entorno simulado.

1.2 Objetivos del Trabajo

Los objetivos de este proyecto se pueden dividir en tres bloques, filtrando el nivel de granularidad de los mismos en dos de ellos. Por un lado, los objetivos generales del proyecto o de más alto nivel; por otro lado, los objetivos específicos del proyecto a más bajo nivel, y que persiguen, en conjunto, el cumplimiento de los objetivos generales. Por último, se expondrán los objetivos personales que persigue.

1.2.1 Objetivos generales

El objetivo principal de este TFM es único y claro: estudiar y desarrollar un enfoque de pentesting para Sistemas de Control Industrial (ICS).

A través del honeypot de baja interacción y de código abierto Conpot, la meta final de este proyecto es estudiar y desarrollar un enfoque de pentesting en un entorno simulado y controlado que sirva como metodología base para enfrentar futuras auditorías de seguridad contra este tipo de sistemas. Esto implicará indirectamente la posterior mitigación de las vulnerabilidades, disminución de las amenazas y la consiguiente mejora de la seguridad global.

1.2.2 Objetivos específicos

Los objetivos específicos son listados a continuación:

- Poner en marcha en una máquina virtual Conpot.
- Definir un ejemplo con Conpot.
- Clasificar los elementos que pueden ser simulados de un ICS en el honeypot.
- Definir los datos que pueden ser recogidos en un honeypot.
- Realizar pruebas sobre el entorno.
- Realizar un enfoque de pentesting para Sistemas de Control Industrial.

1.2.3 Objetivos personales

Cualquier proyecto de este calado, como lo es un Trabajo de Fin de Máster, presenta unos objetivos personales. El objetivo más grande que persigue con su realización es concluir el Máster de Seguridad de las TIC de forma satisfactoria, aplicando los

conocimientos adquiridos durante el transcurso del mismo y, además, yendo un poco más allá.

Otro de los objetivos personales que se pretende conseguir es mejorar los conocimientos de pentesting del autor; ahondando en un área desconocida, como son los Sistemas de Control Industrial y SCADA, lo que permitirá desarrollar habilidades que serán útiles en un futuro para enfrentar proyectos de auditorías de seguridad en las que están implicadas tecnologías de las cuales se desconoce.

Por último, el desarrollo de este Trabajo de Fin de Máster pretende dar un empujón al autor del mismo para finalizar su etapa universitaria y adentrarse profesionalmente y de manera óptima en el mundo laboral de la ciberseguridad. En concreto en el área del pentesting o rama derivada de la misma sobre seguridad ofensiva.

1.3 Enfoque y método seguido

Para que los objetivos del proyecto sean cumplimentados, se van a plasmar los mismos en una serie de fases, sobre las cuales, más adelante, se definirán una serie de tareas más específicas en el apartado de “Planificación del trabajo”.

Las fases son las siguientes:

1) ***Planificación del proyecto.***

   Esta primera fase es clave en el desarrollo posterior del proyecto. En ella se definen el contexto y justificación del trabajo, se fijan los objetivos que se pretende cumplir y, lo más importante, las tareas a desarrollar y su planificación temporal dentro del proyecto. Además, junto a la planificación temporal, se fijan unas metas temporales en forma de hitos.

2) ***Construcción del laboratorio.***

   En esta segunda fase se realizará todo lo necesario para poner en marcha el laboratorio sobre el que trabajar. Para ello será necesaria la instalación de un software de virtualización, la instalación de una máquina virtual atacante y la instalación de Conpot en un entorno ‘dockerizado’. Una vez instalado, se deberá configurar para que funcione correctamente. Adicionalmente, se deberá configurar el laboratorio para que la conexión únicamente sea posible entre estas dos “máquinas”.

3) ***Aprendizaje de Conpot.***

   En todo proyecto es necesario conocer en profundidad la tecnología a emplear. Esta fase pretende aumentar los conocimientos del investigador acerca de los Sistemas de Control Industrial. Una vez adquirida una base teórica, se procederá con Conpot: cómo funciona, qué datos se pueden simular, qué datos se pueden recoger de un honeypot, etc. son algunas de las tareas a desarrollar en esta fase.

4) ***Estudio del enfoque de pentesting en ICS.***

Fase crucial en el desarrollo del proyecto y núcleo del mismo. Se elaborará el planteamiento o estudio de una auditoría de seguridad de caja negra. En este tipo de auditoría el pentester no cuenta con ningún tipo de información del sistema; simula a la perfección un ataque real desde fuera de la organización. En este apartado se expondrá el enfoque de pentesting para ICS acompañado de las pruebas realizadas durante el estudio.

5) Conclusiones y trabajo futuro.

Etapa culmen del proyecto en la cual serán expuestas las conclusiones obtenidas con el desarrollo del trabajo. Se valorarán el grado de cumplimiento de los objetivos marcados al inicio del mismo, los plazos estimados en la planificación temporal y las posibles líneas de trabajo futuro.

6) Finalización de la memoria y desarrollo de la defensa

En esta última etapa serán realizados los últimos retoques de la memoria para proceder a su entrega. También se ha incluido el desarrollo de la presentación para la defensa del TFM.

1.4 Planificación del Trabajo

En este apartado son definidas las tareas, la estimación temporal de cada una de ellas y la planificación temporal del proyecto. Todo ello quedará recogido en un diagrama de Gantt.

1.4.1 Tareas

Definidos los bloques de trabajo o fases en el apartado anterior, se procede a detallar las tareas asociadas a cada una de ellas:

1) Planificación del proyecto.

- **Definición del proyecto.** En esta tarea están recogidas todas las actividades relacionadas con la definición inicial del proyecto: reuniones con los directores del TFM en las cuales se dictamina cuál va a ser el enfoque que va a tomar el proyecto junto a su objetivo principal y las directrices a seguir.

- **Desarrollo de los objetivos.** Definición tanto de los objetivos generales como específicos que se deben cumplir con el proyecto.

- **Definición del alcance.** Desarrollo del enfoque y la metodología del proyecto. En esta tarea son fijadas las fases o etapas de las que constará el proyecto y de las cuales dependerán las entregas a realizar durante el mismo.

- **Planificación del trabajo.** Definición más específica de las tareas a realizar durante el proyecto. A su vez, se acompaña con una planificación temporal del mismo mediante un diagrama Gantt.

- **Definición de la arquitectura.** Exposición abstracta de la arquitectura del proyecto.

- **Evaluación de riesgos.** Identificación y evaluación de los posibles riesgos, impacto y medidas de mitigación.

- **Breve sumario de productos obtenidos.** Se definen las entregas parciales que serán realizadas durante el proyecto; estarán marcadas por los hitos parciales de cada PEC (Prueba de Evaluación Continua).

2) ***Construcción del laboratorio.***

- **Instalación Soft. Virtualización.** Descarga e instalación del software de virtualización.

- **Configuración de máquina atacante.** Descarga e instalación de Kali Linux, en su última versión, en el software de virtualización.

- **Configuración de máquina víctima.** Descarga e instalación del contenedor Docker donde se realizará el despliegue de Conpot.

- **Configuración de Conpot.** Configuración y despliegue de Conpot dentro del contenedor docker. En esta tarea se realizarán las configuraciones necesarias para que funcione correctamente.

- **Configuración del laboratorio.** Se deberá configurar el laboratorio para que la conexión únicamente sea posible entre las máquinas que conforman el laboratorio; en ningún caso podrán acceder a Internet ni ser accedidas ni desde Internet ni desde la propia LAN.

3) Aprendizaje de Sistemas de Control Industrial y Conpot.

- **Estudio y definición del funcionamiento de ICS.** Esta fase pretende aumentar los conocimientos del investigador acerca de los Sistemas de Control Industrial: componentes, funcionamiento, protocolos de comunicación, vulnerabilidades comunes, etc.

- **Adquisición de conocimientos del Conpot.** Mediante el uso del Conpot se pretende que el investigador se familiarice y aumente sus conocimientos acerca de esta tecnología.

- **Definición de datos simulables.** Descripción de los elementos que se pueden simular en el Conpot asociados a los elementos de un Sistema de Control Industrial.

- **Descripción de datos recolectables.** Datos que se pueden recoger en un honeypot de estas características.

- **Definición de un ejemplo con Conpot.** Explicación del entorno una vez montado y de las posibilidades del mismo.

4) Estudio del enfoque de pentesting en ICS.

- **Definición de fases de un pentest.** Definición y explicación de las fases de una auditoría de seguridad corriente.

- **Enfoque de pentesting para ICSs.** Desarrollo de un enfoque de pentesting para Sistemas de Control Industrial.

- **Realización del pentest sobre Conpot.** Sometimiento a Conpot a una auditoría de seguridad y captura de evidencias.

- **Exposición de pruebas realizadas.** Desarrollo de las pruebas realizadas sobre el entorno.

5) Conclusiones y trabajo futuro.

- **Exposición de conclusiones.** Desarrollo de las conclusiones obtenidas al finalizar el proyecto.

- **Valoración de cumplimiento.** Vista a la planificación temporal inicial, a las tareas y a los objetivos y valoración del grado de cumplimiento.

- **Definición de líneas de trabajo futuras.** Descripción de posibles líneas de trabajo futura o de mejora.

- **Desarrollo de conclusiones personales.** Exposición de las conclusiones personales tras la finalización del trabajo y la completitud del Máster de Seguridad de las TIC.

6) Finalización de la memoria y desarrollo de la defensa

- **Maquetación de la memoria.** Revisión final de la memoria del proyecto y realización de la edición de aspectos necesarios. Actividad previa a la entrega del último hito temporal.

- **Desarrollo de la presentación.** Desarrollo de la presentación que será utilizada para defender el Trabajo de Fin de Máster.

- **Edición del video con la presentación final.** Edición del video que será utilizado en la defensa del TFM acompañado de la voz en *off* del autor del trabajo y responsable de la defensa.

1.4.2 Planificación temporal

El proyecto, realizado durante 4 meses junto a otras 4 asignaturas, tiene una dedicación de 9 créditos ECTS, equivalentes a 225-250 horas de trabajo según la UNED [4].

En la siguiente tabla se muestra la estimación temporal de cada tarea. El tiempo empleado en la redacción de la memoria se ha incluido en las horas estimadas para las tareas para las cuales es necesario:

	Tareas	*Tiempo estimado (h.)*
1	***Definición del proyecto***	9
2	***Desarrollo de los objetivos***	4
3	***Definición del alcance***	4
4	***Planificación del trabajo***	4
5	***Definición de la arquitectura***	3
6	***Evaluación de riesgos***	2
7	***Breve sumario de productos obtenidos***	**3**
8	***Instalación Soft. Virtualización***	1
9	***Configuración de máquina atacante***	1
10	***Configuración de máquina víctima***	1
11	***Configuración de Conpot***	15
12	***Configuración del laboratorio***	1
13	***Estudio y definición del funcionamiento de ICS***	25
14	***Adquisición de conocimientos del Conpot***	10
15	***Definición de datos simulables***	10
16	***Descripción de datos recolectables***	10
17	***Definición de un ejemplo con Conpot***	10
18	***Definición de fases de un pentest***	9
19	***Enfoque de pentesting para ICSs***	40
20	***Realización del pentest sobre Conpot***	40
21	***Exposición de pruebas realizadas***	12
22	***Exposición de conclusiones***	3
23	***Valoración de cumplimiento***	1
24	***Definición de líneas de trabajo futuras***	2
25	***Desarrollo de conclusiones personales***	1
26	***Maquetación de la memoria***	5
27	***Desarrollo de la presentación***	10
28	***Edición del video con la presentación final***	5

Tabla 1 - Estimación temporal de tareas

	Fases	**Tiempo estimado (h.)**
1	*Planificación del proyecto.*	29
2	*Construcción del laboratorio.*	19
3	*Aprendizaje de Sistemas de Control Indutrial y Conpot.*	65
4	*Estudio del enfoque de pentesting en ICS.*	101
5	*Conclusiones y trabajo futuro.*	7
6	*Finalización de la memoria y desarrollo de la defensa.*	20
	Total	**241**

Tabla 2 - Estimación temporal de cada fase

Realizando la suma de dedicación de cada fase se obtiene un total de 241 horas.

A partir de la estimación temporal de cada tarea y fase se obtiene el siguiente diagrama de Gantt. Para exponer el tiempo de dedicación previsto se ha optado por este tipo de diagrama debido a lo gráfico y sencillo de entender que resulta. Debido a la carga de trabajo del resto de asignaturas, se ha establecido una jornada de 3 horas dedicadas al TFM, de Lunes a Sábado.

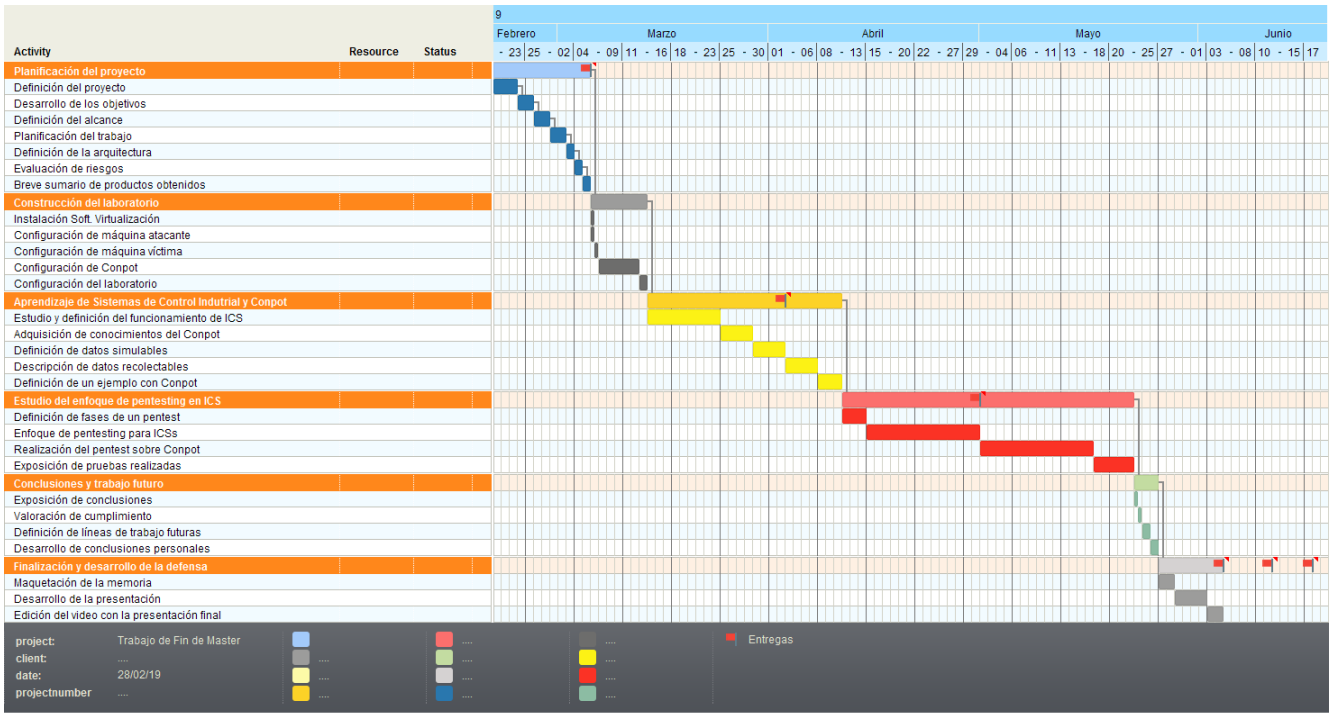


Ilustración 1 - Diagrama de Gantt inicial

1.5 Breve sumario de productos obtenidos

El proyecto se divide en diferentes productos parciales entregables. Dichos entregables estarán marcados por hitos temporales en forma de PECs, que en su conjunto conforman el Trabajo de Fin de Máster.

- **PEC 1 (20/02/2019 – 05/03/2019).** Esta primera entrega consiste en la elaboración del DOP (Documento de Objetivos del Proyecto). Deberá ser definido el plan de trabajo; clave en el funcionamiento del TFM: problema que se pretende resolver, trabajo que se llevará a cabo, descomposición de tareas, planificación temporal, metas temporales, arquitectura, herramientas, objetivos y metodología a seguir.

- **PEC 2 (06/03/2019 – 02/04/2019).** En esta segunda entrega parcial se redactarán los capítulos que sean necesarios para describir el funcionamiento básico de un Sistema de Control Industrial: definición, componentes, protocolos, funcionamiento, vulnerabilidades comunes, etc. También se agregará a la memoria los datos simulables dentro de Conpot.

- **PEC 3 (03/04/2019 – 30/04/2019).** Esta segunda parcial entrega se compone de dos etapas.

  - Por un lado se finalizará la fase 3 añadiendo a la memoria los datos que se pueden recoger en un honeypot de estas características y la definición de un ejemplo con Conpot.

  - Por otro lado, se iniciará la fase 4 relativa al estudio del enfoque de pentesting en un Sistema de Control Industrial: Se definirán las fases de un pentest convencional y se desarrollará el enfoque de pentesting para ICSs.

- **PEC 4 (01/05/2019 – 04/06/2019).** Se trata del último entregable relativo a la memoria.

  - Se aplicará el enfoque definido para llevar a cabo el pentest sobre Conpot exponiendo las pruebas realizadas en la memoria, a modo de anexo, finalizando así con la fase 4.

  - Las conclusiones, el grado de cumplimiento y el trabajo futuro serán redactados junto a las conclusiones personales dando por finalizada la memoria.

  - Se maquetará la memoria llevando a cabo las últimas revisiones.

- **PEC 5 (05/06/2019 – 11/06/2019).** Última entrega en la que se recoge la presentación de la defensa del TFM a través de un video en el que se muestran las diapositivas realizadas junto a la voz en *off* del autor del trabajo.

1.6 Evaluación de riesgos

En la siguiente sección serán identificadas las posibles amenazas que pueden aparecer durante el transcurso del proyecto. Aparecerán acompañadas de su impacto y riesgo asociado, además de los planes de contingencia a desplegar para que, en el caso de que cualquiera de ellas se presentara, tengan el menor impacto posible en el desarrollo del proyecto.

El impacto de cada una de ellas puede encuadrarse en cuatro intensidades diferentes, situadas en un intervalo de 0 a 100:

- Bajo (10): El proyecto se ve retrasado en 8 horas o menos.
- Medio (20-30): El proyecto se retrasado entre 8 y 24 horas.
- Alto (50-70): El proyecto se ve retrasado entre 24 y 48 horas.
- Crítico (90-100): El proyecto se ve retrasado más de 48 horas.

Para calcular la probabilidad se realiza una estimación de lo probable que sea que ocurra según las circunstancias del problema que describe la amenaza. Así, el riesgo de una amenaza será la probabilidad de aparición de la misma multiplicado por el impacto asociado, dando lugar a un valor entre 0 y 100.

Problema	**Alcance del proyecto inabarcable**
Descripción	Se ha realizado un alcance del proyecto demasiado optimista no siendo posible el cumplimiento de la planificación temporal.
Plan de contingencia	Reducir el número de tareas definidas o el grado de profundidad de las mismas.
Probabilidad	50%
Impacto	60/100
Riesgo	30/100

Tabla 3 - Riesgo 01

Problema	Problemas con Conpot
Descripción	Pueden existir problemas con librerías o incompatibilidades que no permitan el avance normal del proyecto.
Plan de contingencia	▪ Intentar solucionar el problema manualmente. ▪ Como última instancia se valoraría la sustitución de Conpot por otro proyecto de similares características, siendo condición *sine qua non* que sea un honeypot de tipo SCADA.
Probabilidad	20%
Impacto	60/100
Riesgo	12/100

Tabla 4 - Riesgo 02

1.7 Herramientas

A continuación se listan las herramientas que serán utilizadas para el desarrollo del proyecto:

- **VirtualBox[1].** Software de virtualización para arquitecturas de x86/amd64. Se ha elegido esta herramienta por ser de código abierto y gratuita.
- **Kali Linux[2].** Distribución de Linux especializada en seguridad informática. Se instalará en una máquina virtual desde donde se llevarán a cabo las pruebas necesarias y el test de penetración.
- **Docker[3].** Proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software. De esta manera, proporciona una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Es posible ejecutar diferentes contenedores independientes dentro de una sola instancia de Linux, evitando así la sobrecarga de iniciar y mantener máquinas virtuales.
- **Conpot[4].** Honeypot de baja iteración que simula un Sistema de Control Industrial diseñado para ser fácil de arrancar, modificar y extender. Provee de un número de protocolos industriales comunes, capaz de emular una compleja

[1] Página oficial de VirtualBox: https://www.virtualbox.org/
[2] Página oficial de Kali Linux: https://www.kali.org/
[3] Página oficial de Docker: https://www.docker.com/
[4] Página oficial de Conpot: http://conpot.org/

infraestructura industrial para convencer al enemigo de que ha encontrado un enorme complejo industrial. Conpot es desarrollado y mantenido por Honeypot Project.

- **Google Drive[5].** Servicio que ofrece el almacenamiento de archivos en la nube. Se permite compartir los archivos con determinados usuarios. El espacio de almacenamiento gratuito es de hasta 15 GB, suficientes para la realización del Trabajo de Fin de Máster. Se utilizará para almacenar todo lo referente al mismo: documentación, copias de seguridad, etc.

- **Microsoft Office[6].** Paquete de software de oficina de Microsoft. Se utilizará para la redacción de la memoria y para la elaboración de la presentación final.

1.8 Arquitectura

La configuración y puesta en marcha del laboratorio debe de hacerse en un entorno controlado. Uno de los usos tradicionales de un honeypot es permitir conexiones desde el exterior hacia el honeypot, generalmente situado en una DMZ, para recoger los datos de los ataques/atacantes o para que salten las alarmas dentro de la organización en el caso de que el honeypot sea atacado: un ataque al honeypot será señal inequívoca de que la empresa propietaria del honeypot está siendo atacada, por lo que podrá activar sus medidas de seguridad o planes de contingencia.

En cambio, en el caso que aborda este Trabajo de Fin de Máster no se deberá exponer el honeypot al exterior. Se deberá crear una red aislada en la que formarán parte la máquina atacante y el honeypot. El objetivo primordial del establecimiento de este honeypot será, por tanto, ser atacado por el investigador para así realizar el estudio de un enfoque de pentesting en un Sistema de Control Industrial.

Para conseguir este fin, se virtualizará la máquina atacante y se introducirá en una red interna la cual no tendrá acceso a Internet ni será accesible ni por el sistema anfitrión ni por el resto de dispositivos de la red local. En la máquina virtual atacante se albergará un contenedor docker en el que se desplegará Conpot. Este contenedor podrá comunicarse exclusivamente con la máquina virtual atacante a través de una interfaz de red 'puente' creada para ese fin.

Únicamente formarán parte de esta red estos dos sistemas, ya que añadir más sistemas y redes simulando una red de una corporación queda fuera del alcance de este TFM y no es el objetivo del mismo. Se supondrá que ya se han comprometido las redes y sistemas necesarios realizando movimientos laterales o *pivoting* entre sistemas para que el atacante tenga acceso directo al sistema que simula un componente de un Sistema de Control Industrial.

---

[5] Página oficial de Google Drive: https://www.google.com/intl/es_ALL/drive/
[6] Página oficial de Microsoft Office: https://products.office.com/es-es/home

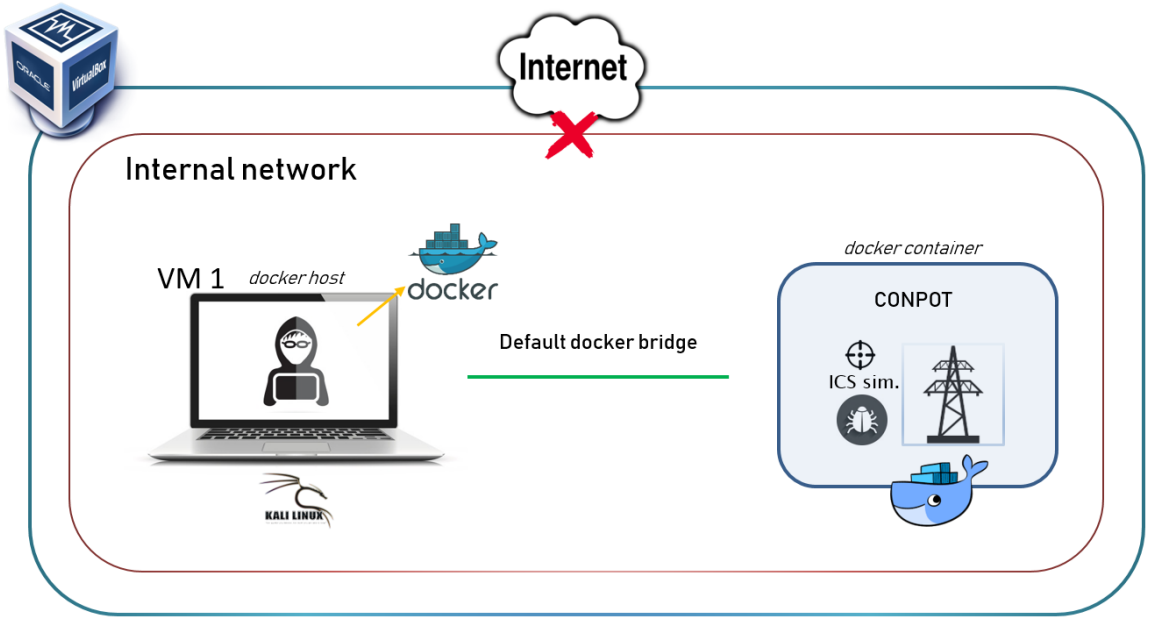


Ilustración 2 - Esquema abstracto de la arquitectura de red

1.9 Breve descripción de los otros capítulos de la memoria

En esta sección se realizará un resumen de cada capítulo de la memoria:

- **Capítulo 1: Introducción.** En este primer capítulo se presentará el plan de trabajo del proyecto. Se describirá la justificación del proyecto, los objetivos que se pretenden alcanzar con su realización, el enfoque y método a seguir, el conjunto de tareas a desarrollar, la planificación temporal, metas temporales, las herramientas de las que se hará uso, etc. En sí constituye un elemento clave en el desarrollo del proyecto.

- **Capítulo 2: Sistemas de Control Industrial.** En el segundo capítulo se definirá el concepto de Sistema de Control Industrial. Se hará un recorrido por sus componentes, protocolos más comunes y las vulnerabilidades que, por definición, existen en estos protocolos. Se listarán también los activos más importantes dentro de un ICS y las vulnerabilidades que tienen lugar con más frecuencia. Este capítulo será una pieza clave a la hora de entender y saber cómo enfocar un pentest sobre un ICS.

- **Capítulo 3: Conpot.** Se presentará el honeypot de baja interacción: Conpot. Dicho honeypot simula un componente de un ICS y será utilizado posteriormente para realizar una PoC (*Proof of Concept)*. Además se detallarán los elementos que se pueden simular y recolectar, junto a la definición de un ejemplo con Conpot.

- **Capítulo 4: Pentesting en Sistemas de Control Industrial (ICS).** Capítulo central de la memoria. Se recordará cómo es un pentest dirigido contra sistemas de las tecnologías de la información y se definirá y se detallará la *ICS Cyber Kill Chain* junto a sus etapas. Ambos conceptos serán importantes a la hora de elaborar el enfoque y metodología de pentesting para Sistemas de

Control Indutrial, que es lo siguiente que se realizará. Así, primero se presentarán las diferentes posibilidades en cuanto al punto de partida a nivel de red de un pentest contra un ICS y cómo llegar hasta la red OT, punto de partida del enfoque de pentesting definido. Se detallarán las diferentes fases a seguir en la metodología, explicando cada una de ellas detalladamente y mostrando las herramientas disponibles junto a las posibles estrategias a seguir para el correcto desempeño de cada fase.

- **Anexo I: Pentest sobre Conpot.** Por último, en forma de anexo, se expondrá la PoC donde se aplica y se pone en práctica el enfoque de pentesting definido sobre Conpot, que simulará, en concreto, un PLC de un ICS.

2. Sistemas de Control Industrial

Conocer la tecnología que se va a auditar es un paso intermedio necesario y decisivo a la hora de llevar a cabo una auditoría de seguridad; se podría resumir en: establecer una mentalidad de seguridad, adquirir conocimientos técnicos y aprender técnicas de ataque aplicadas a la tecnología que se desea auditar. Así, este capítulo llevará tanto al lector como al investigador a conocer lo necesario acerca de los componentes y protocolos más comunes utilizados en Sistemas de Control Industriales; acompañados de la arquitectura y las vulnerabilidades más comunes.

2.1 Introducción a los ICS

Una infraestructura crítica es aquella cuyo funcionamiento es indispensable y no permite soluciones alternativas, por lo que su perturbación o destrucción tendría un grave impacto sobre la sociedad y economía de un país. Según el CNPIC (Centro Nacional para la Protección de las Infraestructuras Críticas) ejemplos de servicios que utilizan este tipo de infraestructuras son [5]: la Alimentación, el Agua, la Energía, la Industria Nuclear, la Salud, etc.

Todas estas infraestructuras críticas están gobernadas por Sistemas de Control Industriales que permiten administrar y regular el correcto comportamiento de las mismas para lograr el fin que persiguen. Estos Sistemas de Control Industrial, a su vez, están formados por una serie de componentes y utilizan protocolos específicos para su comunicación.

2.1.1 Niveles jerárquicos

Los componentes que constituyen un Sistema de Control Industrial se pueden encuadrar en 5 niveles o categorías jerárquicas diferentes. El estándar ISA 95 [6] de la integración de los sistemas de control empresarial los describe de la siguiente manera:

- **Nivel 0.** En este nivel se encuentran todos los procesos físicos de producción: la maquinaria.
- **Nivel 1.** Define las actividades involucradas directamente en la medición y manipulación del proceso de producción situados en el nivel anterior.
- **Nivel 2.** Actividades de monitorización, supervisión y control de los procesos de producción. Los componentes de este nivel interactuarán con los del nivel 1 para lograr su cometido.
- **Nivel 3.** Actividades de control de flujo y órdenes para producir el producto final deseado. Estos componentes guardan históricos o registros que permiten optimizar el proceso de producción.
- **Nivel 4.** Define los procesos de negocio necesarias para gestionar una organización de fabricación. Alta gestión logística y empresarial de los procesos, gestión de inventarios, facturación, contabilidad, etc.

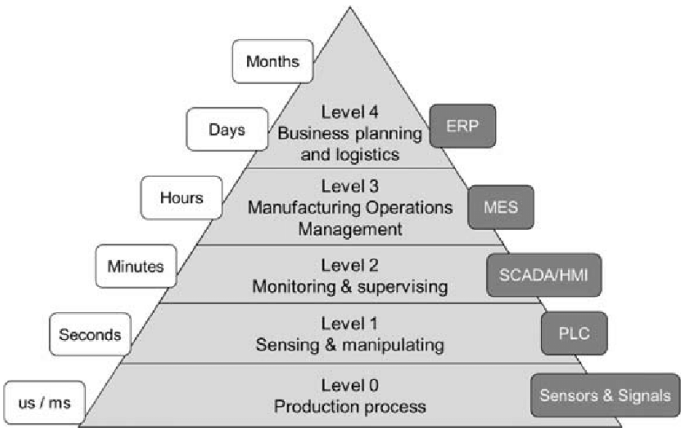


Ilustración 3 - Jerarquía definida en ISA-95 (Extraída de: [7])

2.2 Componentes

En este apartado se realizará una descripción de los componentes más comunes dentro de un Sistema de Control Industrial. Para ello, se utilizará la división jerárquica desarrollada por el modelo ISA 95 descrita anteriormente, agrupando cada componente dentro de los niveles definidos para una mejor comprensión. El objetivo de este apartado no es realizar una descripción exhaustiva de los mismos sino exponer los conocimientos necesarios para entender la relación entre los componentes y poder llevar así a cabo el enfoque de la auditoría de seguridad.

2.2.1 Nivel 0

En este nivel se encuentran los procesos físicos de producción: motores, válvulas, sensores, actuadores, etc., es decir, la maquinaria física. Reciben órdenes directamente de la capa superior y transmiten información. El objetivo último de la auditoría de seguridad es conseguir a través de elementos de capas superiores la manipulación, directa o indirecta, del funcionamiento de elementos de este nivel ya que afectan directamente al proceso de producción. La información meramente técnica de cada uno de los elementos englobados en este grupo se escapa del objetivo de este proyecto.

2.2.2 Nivel 1

Corresponden a este nivel elementos que se sitúan muy cerca de los procesos físicos de producción, estando directamente involucrados en la manipulación de los mismos; tanto en la manipulación como en la recogida de información. Transmiten datos hacia capas superiores del modelo jerárquico y pueden recibir órdenes de los mismos.

2.2.2.1 PLC

Un PLC (Programmable Logic Controller) es un autómata programable cuya función es realizar acciones a partir de los programas almacenados en su memoria. Partiendo de unas entradas el PLC emite unas salidas que se ven reflejadas en acciones sobre los elementos de la capa 0. Las entradas se pueden recibir a través de sensores de temperatura, de movimiento, de presión, etc. Estos dispositivos están diseñados para controlar procesos de manera secuencial y pueden ser programados para controlar cualquier tipo de máquina [7]. La comunicación con elementos de capas iguales o

superiores se realiza a través de protocolos como Modbus, DNP, Ethernet o Profinet [8].

2.2.2.2 RTU

RTU (Remote Terminal Units) es un dispositivo formado por microprocesadores capaz de obtener señales independientes de los procesos y enviarla a un sitio remoto para que sea procesada. Su función es la monitorización de datos y transmisión de los mismos. Recogen información a partir de, por ejemplo, cualquier tipo de sensor de la capa 0 y la envían a elementos de capas iguales o superiores, como un Sistema de Control Distribuido, o para tener una visión precisa del proceso de producción. Modbus, DNP y el estándar IEC 60870 son algunos de los protocolos de comunicación utilizados por este tipo de dispositivos [9].

2.2.2.3 IED

Siglas correspondientes a *Intelligent Electronic Devices*. Se trata de componentes usados principalmente en plantas eléctricas para llevar un control sobre los sistemas eléctricos como condensadores, transformadores, etc. Reciben la información a través de sensores y son capaces de tomar decisiones en función de los datos monitorizados [10]. Un ejemplo de actuación sería bajar o subir el voltaje tras percibir una anomalía del mismo en la monitorización.

2.2.2.6 DCS

Sistema de Control Distribuido. Su función final es similar a la de un PLC, con la diferencia de que está compuesto por sistemas distribuidos. Permite tanto el control como la monitorización. Un DCS está asociado a instalaciones grandes donde es provechoso distribuir los controladores y las E/S a lo largo de la instalación, incluyendo varias estaciones de operación ubicadas en la sala de control [11]. En cuanto a las diferencias sobre un PLC, según un artículo en Rockwell Automation [12], “las funciones específicas de alta velocidad y lógica discreta se pueden controlar con el uso de PLC exclusivos, mientras que los sistemas de control distribuido (DCS) se utilizan cuando algunos controladores y puntos de acceso deben conectarse y estar accesibles en toda la planta”. Además, gracias a el concepto de distribuido, un cambio en la programación del DCS se verá plasmada en el resto de componentes del mismo, sin necesidad de aplicar individualmente el cambio en cada uno de ellos; siendo este un punto a favor frente a los PLCs si es necesario que la instalación cuente con muchos componentes que desarrollen esta función.

2.2.3 Nivel 2

A esta capa corresponden componentes encargados tanto de la monitorización y supervisión como del control y dirección del proceso industrial. Para ello, reciben información de la capa inferior (PLCs, RTUs, DCS, etc) y pueden actuar directamente sobre ellos. También recolectan información y la transmiten a capas superiores para la elaboración de informes y seguimiento.

2.2.3.1 SCADA

Un SCADA (Supervisory Control And Data Adquisition) permite supervisar y controlar los procesos asociados a un área geográfica relativamente grande. Un ejemplo sería el control de las entradas y salidas de los diferentes procesos en las distintas fábricas

que se controla a lo largo de una zona geográfica extensa. Según Tripp Roybal [13] la palabra clave es “supervisión”: los sistemas supervisores no suelen controlar directamente los elementos de campo; proporcionan información a los ingenieros que toman decisiones sobre los procesos.

Comúnmente se confunde el término ICS con SCADA, siendo realmente este último un componente y estando englobado dentro del primero. Así, SCADA es una subcategoría de un ICS.

Ilustración 4 - Paneles SCADA (Extraída de: [14])

2.2.3.2 HMI

Un HMI (Human Machine Interface) o Interfaz Hombre Máquina es un componente que permite interactuar a una persona con los diferentes elementos de las capas inferiores, como PLCs o RTUs, que actúan sobre el sistema de producción. Proporciona un panel gráfico a partir del cual poder transmitir órdenes y visualizar el proceso y flujo de datos de manera gráfica.

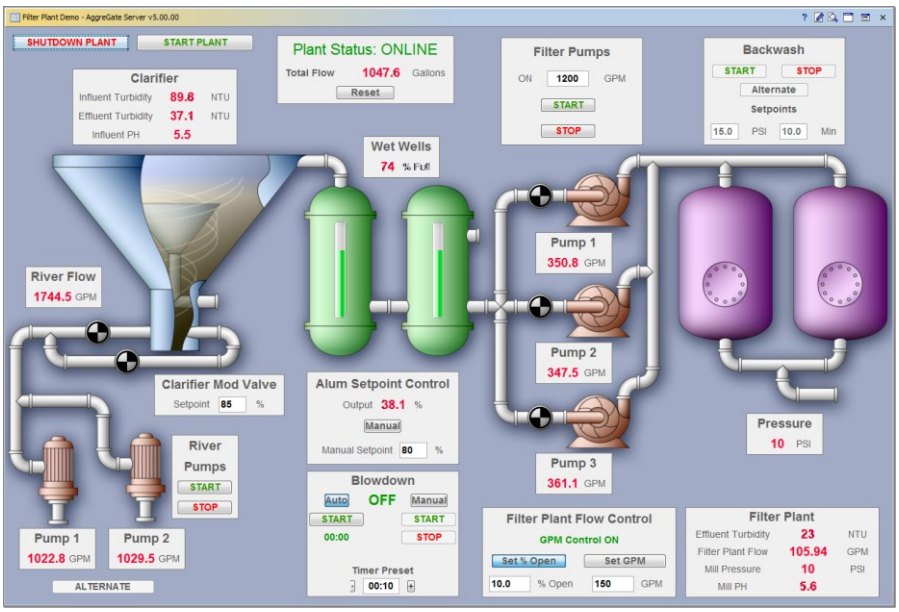


Ilustración 5 - Panel HMI de una planta de filtrado de agua (Extraída de: [15])

2.2.4 Nivel 3

En este nivel se sitúan los componentes que llevan un histórico de todos los datos de interés la planta y los que interactúan con el nivel 4 para optimizar el proceso de producción. Agregan la información de niveles inferiores y la envían al nivel superior para su estudio.

2.2.4.1 Historian

También son conocidos como historiadores. Recogen información relativa a sensores, estadísticas, entradas/salidas, etc.: cualquier tipo de información en grandes cantidades relativa al proceso de producción de las capas inferiores. Sus propósitos son múltiples: tener un backup de la información para que en el caso de que ocurra una catástrofe poder saber qué ha ocurrido; como medida regulatoria para tener los *logs* de las emisiones, producción, etc. En ocasiones se instalan historiadores redundantes para tener uno de respaldo en el caso de que se produzca la caída del principal o situando el secundario en una red diferente a la industrial para consultar este último y así aumentar la segmentación de la red y agregar una capa de seguridad.

2.2.4.2 MES

Un MES (Manufacturing Execution System) es un Sistema de Ejecución de Fabricación. Es la interfaz de conexión entre las capas inferiores y la capa superior donde se gestiona la organización y los procesos de negocio. Estos componentes se relacionan con otros sistemas de información de la capa superior como los Sistemas de Planificación de Recursos Empresariales (ERP), Gestión relacionada con Clientes (CRM), Inteligencia de Negocio (BI), etc. aportando información útil para la gestión del negocio y aplicando las planificaciones llevadas a cabo en esta capa superior en forma de órdenes de fabricación en niveles inferiores.

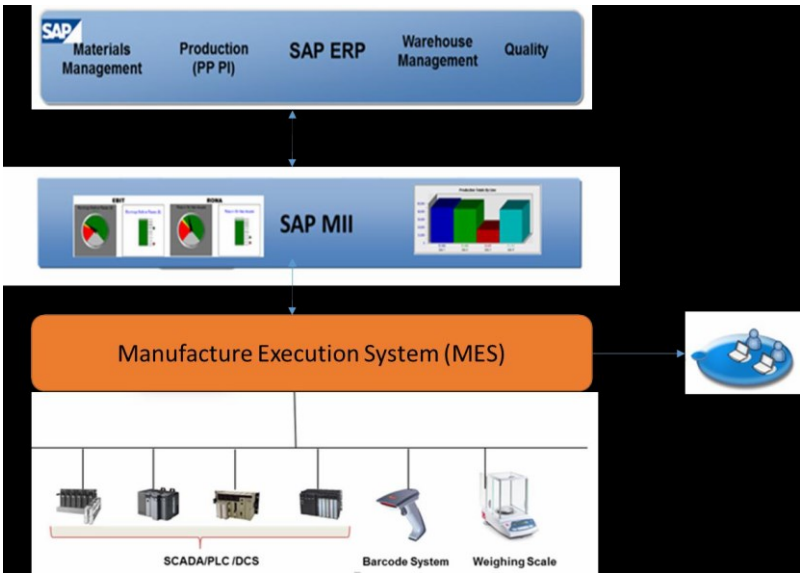


Ilustración 6 - Relación MES con niveles adyacentes (Extraída de: [16])

2.2.5 Nivel 4

A este nivel pertenecen componentes de la parte empresarial de la organización, albergados en la red IT, mientras que el resto de niveles se encuentran en la red OT (esto se verá detallado en este mismo capítulo, concretamente, en el apartado referente a la arquitectura de un ICS). Se trata de programas destinados a la gestión de inventarios, contabilidad, satisfacción del cliente, logística, entre otros. Ejemplos de ellos son: Sistemas de Planificación de Recursos Empresariales (ERP), Gestión relacionada con Clientes (CRM), Inteligencia de Negocio (BI), Recurso Humano (HR), Almacenamiento de Datos de Negocio (BW), etc. Este nivel no es de interés dentro de la auditoría de seguridad que se relata en este trabajo.

2.3 Protocolos: Definición y vulnerabilidades

Para el correcto funcionamiento de un Sistema de Control Industrial, como se ha relatado en la definición de los componentes, estos deben comunicarse, ya sea para trasmitir información de monitorización u órdenes que se acaban plasmando en los procesos físicos. Esta comunicación se lleva a cabo mediante una serie de protocolos.

Por un lado, existirán protocolos destinados a la comunicación entre componentes del nivel físico o nivel 0 (sensores, actuadores, válvulas, motores, etc.) y los del nivel 1 (PLCs, RTUs. IEDs, etc.); por otro lado, coexistirán los protocolos responsables de la comunicación entre los elementos de este nivel 1 y los pertenecientes al nivel 2 de supervisión, adquisición y control de datos (SCADA y HMI). Con el paso del tiempo, algunos de estos protocolos han sido encapsulados dentro de los protocolos de las tecnologías de la información (como TCP/IP y Ethernet) adaptándose a las necesidades de la era de la información. A continuación se describen algunos de los más relevantes.

2.3.1 Profibus

Estándar abierto para la comunicación a través de buses de campo desarrollado en Alemania al principio de la década de los 90. Se trata de un protocolo inteligente, simple y rápido que permite la conexión bidireccional entre dispositivos conectados en un único bus, lo que posibilita, entre otros, un ahorro económico considerable.

Al igual que muchos otros protocolos de un ICS, tiene una arquitectura maestro-esclavo, donde los componentes maestros realizan tareas de control, supervisión y coordinación y los esclavos son los encargados de tareas más específicas. Esto lo combina con la utilización de *tokens*: solo cuando el maestro tiene el *token* puede comunicarse con los esclavos. El esclavo Profibus solo puede comunicarse con un maestro. El nodo maestro de Profibus suele ser un PLC o RTU y los esclavos son sensores, motores u otros dispositivos de control [17]. Existen dos variantes: Profibus DP (comunicación entre sensores/actuadores desde un PLC/RTU) y Profibus PA (destinado a la automatización de procesos).

De la misma manera que muchos protocolos relativos a ICS, este protocolo carece de autenticación. Esto implica la posibilidad de suplantar la identidad del nodo maestro por parte de cualquier otro nodo, cuya consecuencia directa es el control de los esclavos. Los resultados de esto son la alteración del comportamiento del nodo o la Denegación de Servicio (DoS). La mayoría de los nodos maestros en una red Profibus DP están conectados a una red Ethernet haciéndolos susceptibles a casi cualquier tipo de ataque basado en Ethernet [17].

2.3.2 Modbus

Se trata de uno de los protocolos más utilizados. Modbus RTU fue desarrollado en 1979 por Modicon para su uso en los PLCs Modicon. Este protocolo opera en la capa de aplicación del Modelo OSI. Utiliza una arquitectura maestro-esclavo para la comunicación. Se trata de un protocolo solicitud-respuesta.

A través de este protocolo, los esclavos envían la información requerida a los maestros o llevan a cabo la acción solicitada por los mismos. Los maestros pueden dirigirse a los esclavos mediante mensajes broadcast o individualmente; estos últimos solo pueden responder al maestro individualmente [18]. El mensaje que construye el maestro está formado por un identificador del esclavo al que va dirigido, un código de función asociado a una acción y un campo de verificación de errores (en caso de Modbus RTU).

Como en el caso anterior, un esclavo podría ser una válvula, actuador, motor, sensor, etc.; un maestro un PLC, RTU, IED, etc. El maestro establece la conexión con el esclavo, espera a que la información sea transmitida y cierra la conexión [19].

La implementación sobre TCP/IP es la versión más utilizada de Modbus. Emplea el mismo modelo de maestro-esclavo y los mismos códigos funcionales relativos a las acciones. El control de errores realizado por Modbus RTU pasa a ser gestionado por la capa de enlace de TCP/IP. Además, se añaden un campo de identificador de transacción (permite la identificación de la transacción entre maestro-esclavo en el envío de mensajes), otro de protocolo de identificación (se indicará que se trata de Modbus con el valor 0) y uno de longitud (relativa a la longitud de los siguiente campos). Utiliza por defecto el puerto tcp/502 para las comunicaciones.

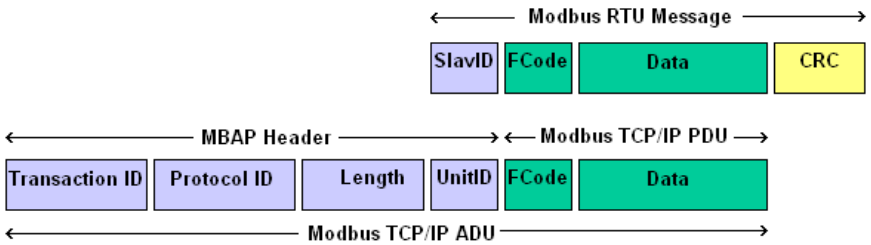


Ilustración 7 - Diferencias Modbus RTU y Modbus TCP/IP (Extraída de: [18])

En cuanto a las vulnerabilidades de este protocolo destacar las siguientes [18]:

- Falta de autenticación. Modbus no incluye un módulo de autenticación. Un atacante puede crear un paquete que sea correcto y mandarlo a través del protocolo.

- Información no cifrada. Vulnerable a un ataque MitM.

- Denegación de servicio. Un atacante podría causar una denegación de servicio mediante la inundación de mensajes a través de la emisión en broadcast a los esclavos.

2.3.3 DNP 3.0

DNP 3.0 (Distributed Network Protocolo 3.0) fue desarrollado en 1993 basado en los estándares IEC 60870-5. Se destaca su uso, sobre todo, en centrales eléctricas, siendo el puente que permite comunicar IEDs u otros controladores con sistemas SCADA, aunque también es utilizado en centrales de petróleo y gas y plantas de tratamiento de agua [20], entre otros. Se trata de un protocolo flexible, robusto y no propietario. En 1998 fue extendido para poder ser encapsulado en TCP/IP o UDP, siendo el primero el ampliamente más usado.

Opera en tres capas: capa de enlace, capa de transporte y capa de aplicación. En los siguientes párrafos se expone una breve explicación de cada una de las capas, no se pretende dar una explicación exhaustiva de las mismas.

La capa de enlace está conformada por [21]: 2 bytes que indican que se trata de un paquete DNP3, 1 byte que indica la longitud de los siguientes valores sin tener en cuenta los CRC; un byte de control que permite fijar los servicios del nivel de enlace, el sentido del flujo; 2 bytes con la dirección de destino en formato Big-endian; 2 bytes con la dirección de origen en el mismo formato y 2 bytes para la cabecera CRC.

La capa de transporte es usada para fragmentar los paquetes grandes. Está conformada por 1 byte FIN y 1 byte FIR, que permiten indicar si se trata del último/primer fragmento de la secuencia; y 6 bits para el número de secuencia utilizado para el reensamblado de los fragmentos [22].

La capa de aplicación [21] se encarga de procesar los fragmentos recibidos del nivel de transporte, y obtener la información de control y monitorización en ellos encapsulados. Los siguientes servicios son proporcionados por esta capa [21]: escritura y lectura de valores, congelación de contadores, selección y ejecución de mandos, etc. Destacar el código de función (1 byte) encargado de indicar qué operación debe realizarse en este nivel.

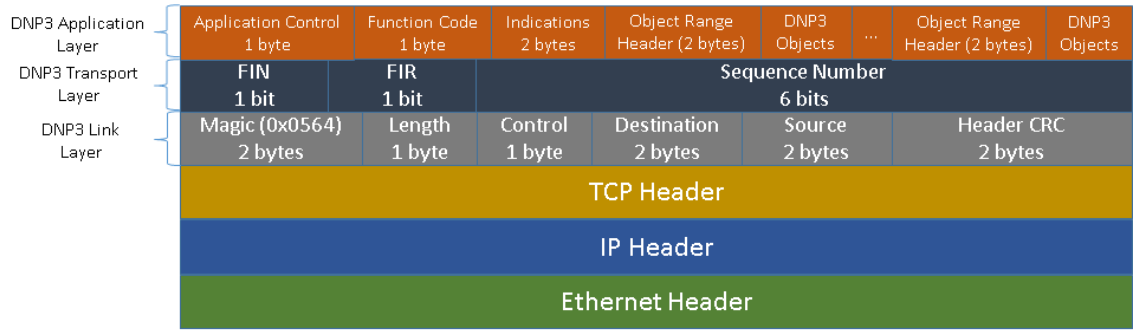


Ilustración 8 - DNP3 sobre TCP/IP y disección de capas (Extraída de: [22])

Al igual que otros protocolos, utiliza un esquema cliente-servidor [20], donde el cliente puede ser el centro de control y supervisión, como un SCADA o HMI, y los servidores son las unidades de control, ya sea un PLC, RTU o un IED, por ejemplo. Cualquiera de los dos nodos puede iniciar la comunicación, al contrario que ocurre en otros protocolos como Modbus. Utiliza por defecto el puerto 20000 (tcp/udp) para las comunicaciones.

Respecto a las vulnerabilidades del protocolo, DNP 3.0 no permite la autenticación ni soporta el cifrado de las comunicaciones. También es vulnerable a ataques DoS.

2.3.4 Profinet

Profinet es un protocolo abierto basado en Profibus que utiliza Ethernet en vez del bus de campo como medio de comunicación. Esto le permite ofrecer la funcionalidad TCP/IP para la transmisión de datos, así como la transferencia de datos a altas velocidades y el soporte de aplicaciones inalámbricas [23]. Facilidad de uso, comunicación en tiempo real y confiabilidad son algunos de los adjetivos de este protocolo.

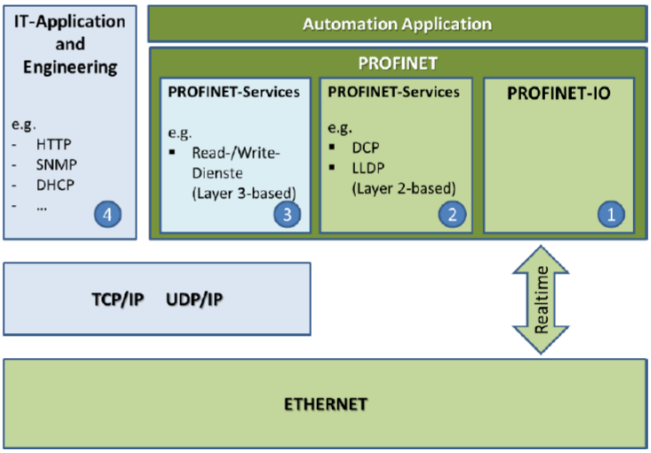


Ilustración 9 - Arquitectura de Profinet (Extraída de: [23])

Profinet presenta diferentes protocolos definidos dentro de su contexto: Profinet I/O, que conecta los dispositivos de campo usando comunicaciones en tiempo real (RT); Profinet CBA, utilizado por aplicaciones de automatización distribuida, basado en los estándares DCOM (Distributed Component Model) y RPC (Remote procedure Call); Profinte IRT, de transferencia de datos isócrono en tiempo real; Profinet MRP, etc.

Profinet I/O utiliza por defecto los puertos tcp/udp: 34962, 34963 y 34964; Profinet CBA usa por defecto el puerto tcp/135.

Este protocolo es vulnerable a las vulnerabilidades que afectan a Ethernet. También están sujetas al tipo de Profinet que se emplee. Por ejemplo, en el caso de Profinet CBA será vulnerable a las vulnerabilidades que afecten a DCOM y RPC.

2.3.5 Otros protocolos

Otros protocolos comúnmente utilizados en entornos de Sistemas de Control Industrial son:

- **OPC.** *Ole For Proccess Control*. Protocolo abierto de arquitectura cliente-servidor. Permite compartir información basada en COM, DCOM, OLE y RCP. Bolívar [19] propone un ejemplo de uso en la representación de la información en un HMI: este dispone de un cliente OPC que envía una petición a un servidor OPC sobre la información que necesita. El servidor OPC utiliza protocolos como DNP 3.0 o Modbus para conseguir la información de, por ejemplo, un RTU, y transmitírselo de vuelta al cliente OPC del HMI.

- **ICCP.** Protocolo abierto de comunicación entre centros de control en tiempo real. Basado en una arquitectura cliente-servidor fue diseñado para el envío de

información a través de redes WAN. Un ejemplo de uso sería el intercambio de datos entre dos SCADA de cada una de sus subestaciones.

2.4 Arquitectura de red

Toda arquitectura de red debe de ser diseñada con unos conceptos de seguridad en mente. Un concepto clave es la segmentación de la red en diferentes zonas protegidas por zonas desmilitarizadas y firewalls. Estas concepciones, que tienen su naturaleza en el modelo de defensa en profundidad, permiten la creación de capas para evitar un ataque directo a las zonas sensibles. Las DMZs entre segmentos de red harán que un atacante tenga que pasar por la misma para pivotar hacia otro nivel aumentando así la dificultad, el tiempo y la posibilidad de detección; los firewalls realizarán un control sobre el tráfico permitiéndolo sólo entre determinados equipos y puertos.

Esto mismo ocurre en un entorno industrial, donde se pueden situar, a nivel de red, a los componentes o equipos en los niveles jerárquicos marcados por el estándar ISA 95. Por un lado, existirá una separación clara entre el nivel 4 correspondiente a la red corporativa donde se sitúan elementos de las tecnologías de la información y el resto de niveles correspondientes al área de ICS donde se encuentran las tecnologías operacionales. Así, el compromiso de la red empresarial no implicará el compromiso directo del Sistema de Control Industrial.

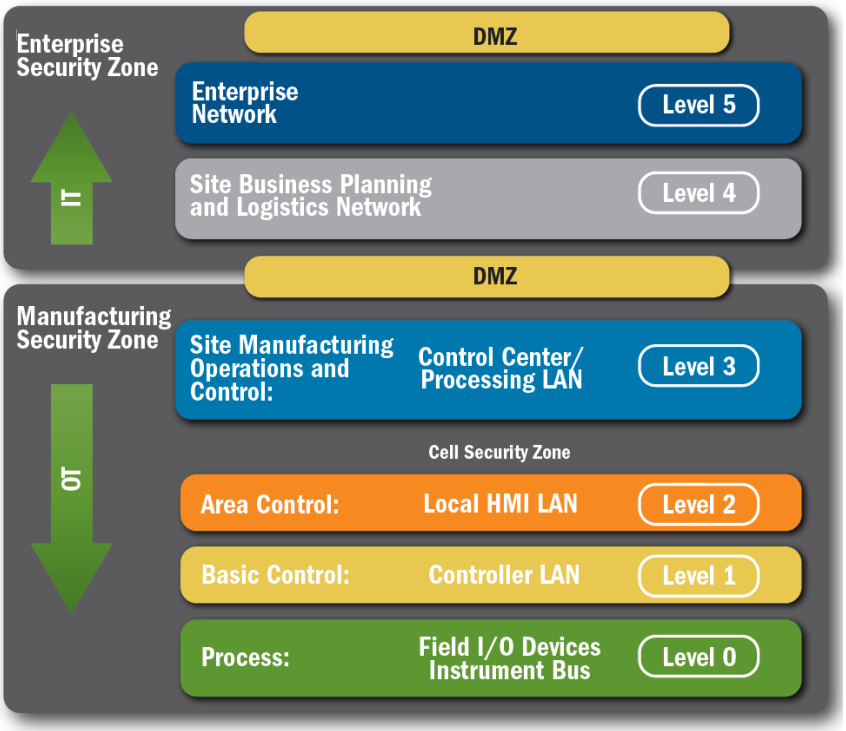


Ilustración 10 - OT/IT: Arquitectura de red segregada en los niveles de ISA 95 (Extraída de: [24])

En la siguiente ilustración se puede ver de manera más desarrollada lo que sería la recomendación segura de la implementación de la arquitectura de red en un Sistema de Control Industrial elaborada por el "*US Department of Homeland Security*" [24]:

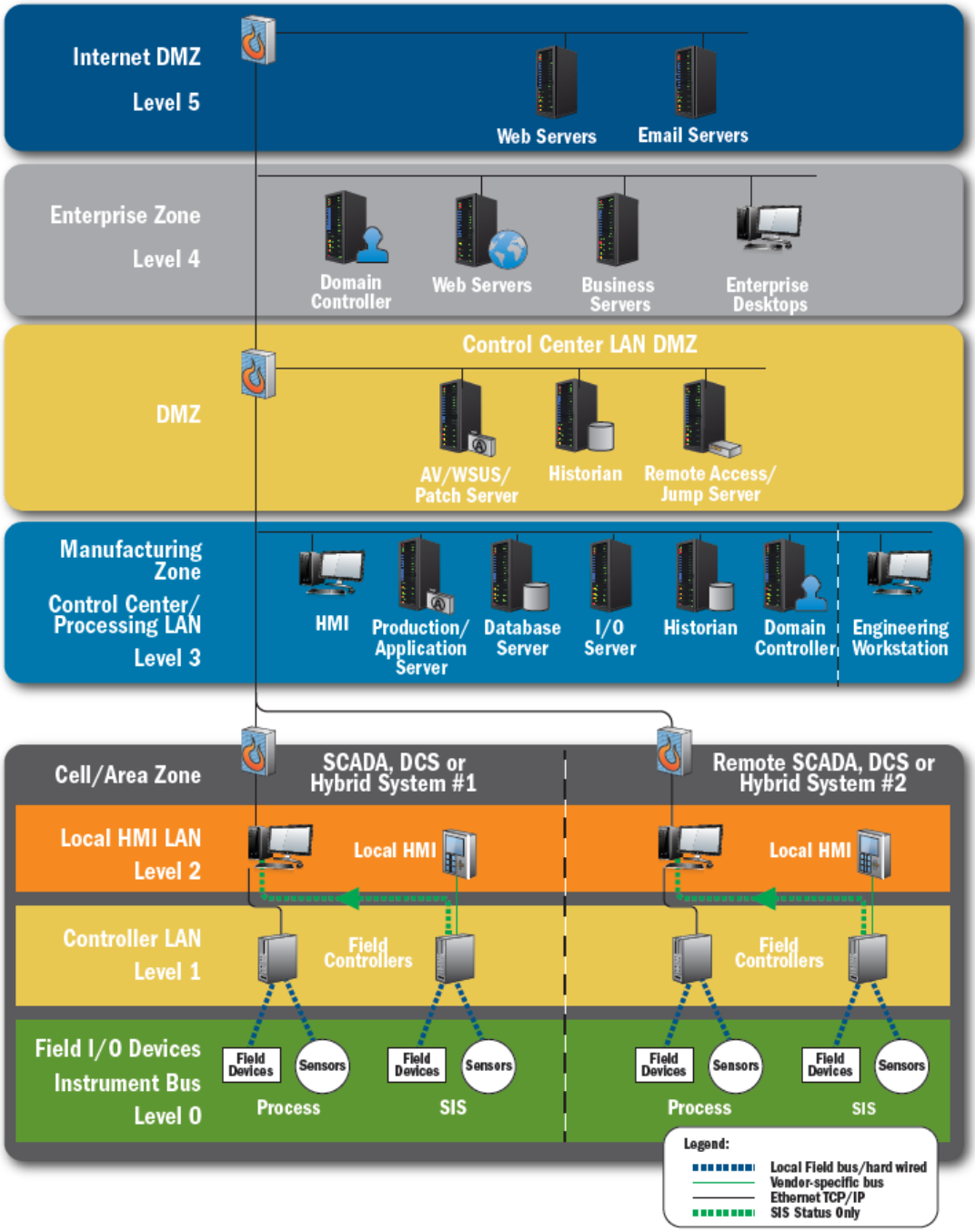


Ilustración 11 - Arquitectura de red recomendada en un ICS (Extraída de: [24])

Como se puede observar, se emplea un modelo de defensa en profundidad mediante el uso de firewalls e implementación de zonas desmilitarizadas. También se puede ver en práctica el concepto de redundancia a través del establecimiento de dos historians en diferentes segmentos de red; esto permite tener un respaldo ante un fallo y, lo más importante, que desde el nivel 4 se pueda acceder a los *logs* con la comunicación del historian secundario situado en la DMZ sin tener que comunicarse directamente con el historian principal del nivel 3 situado en la red OT. A esto se le podría sumar el uso de IDSs, IPSs (Sistemas de Detección/Prevención de Intrusos), SIEMS (Sistema de Gestión de Eventos e Información de Seguridad), honeypots, etc.

Esto, que representa un entorno idílico de máxima seguridad, no siempre es así. En el peor de los casos se encuentra una segmentación pobre de la red con una arquitectura relativamente plana donde muchos componentes se sitúan a un mismo segmento de red y la comunicación entre ellos es posible de forma directa; en otras ocasiones es posible hacer *bypass* las reglas establecidas por el firewall; o lo que es peor, Sistemas de Control y Supervisión, como paneles HMI que permiten tomar el control de los procesos físicos de la planta, expuestos y accesibles directamente desde Internet (estos pueden ser encontrados través de buscadores como Shodan).

2.4.1 Integración IT/OT: riesgos asociados

Como se ha visto, en muchas ocasiones, las aplicaciones o protocolos utilizados por los Sistemas de Control Industrial carecen de las medidas de seguridad necesarias o que cabría de esperar en un sistema cuyo compromiso puede tener tan graves consecuencias. Esto se debe a que, en un principio, este tipo de sistemas no estaban conectados a la red y no se tenía tan en cuenta las amenazas de seguridad desconocidas e impredecibles (*security*: ciberataque, desastre natural, etc.), priorizando la seguridad sobre los peligros y riesgos conocidos cuya probabilidad de suceso es calculable (*safety*: fallo de un proceso, TTL de un componente, etc). Además, mientras que en las tecnologías de la información las prioridades son: confidencialidad > integridad > disponibilidad, en un Sistema Industrial las prioridades siguen un orden completamente distinto, estando en cabeza la integridad seguida de la disponibilidad y la confidencialidad. Esto es debido a las consecuencias físicas que puede tener un fallo de integridad en un ICS puede poner en riesgo tanto la infraestructura crítica que controla como la vida de las personas (*Ej: Fallo en las cantidades de un medicamento, fallo mezcla componentes sector nuclear, etc.*); y en la disponibilidad grandes pérdidas económicas.

Hoy en día, la Industria 4.0 y la conexión entre los dispositivos de la red operacional a la red de las tecnologías de la información de una organización permiten un mayor control sobre los procesos. Esto, que se conoce como la integración de los servicios de la tecnología operacional (*OT: Operational Technology*) con las tecnologías de la información (*IT: Information Technology*) presenta, además de ventajas, grandes riesgos para la seguridad de las infraestructuras críticas.

La mayoría de los protocolos utilizados en Sistemas de Control Industrial fueron diseñados hace muchos años, cuando no existía una unión entre OT e IT. Por ello, prevalecían otros intereses, siendo la confidencialidad de los protocolos algo no prioritario. Así, la mayoría de estos son vulnerables a ataques de hombre-en-el-medio (MitM) debido a la falta de autenticación y cifrado de las comunicaciones. En otras ocasiones, se ha decidido encapsular los protocolos en TCP/IP, algo que tampoco es de ayuda ya que TCP/IP es un protocolo no seguro por definición. Todo esto, que en aquellos años podía parecer que no era peligroso, se convierte en una gran amenaza con la integración de la red IT con la OT. Si un atacante logra comprometer la red correspondiente a IT y consigue pivotar a la red OT tendrá grandes posibilidades de tener éxito en el ataque si no se han tomado las medidas de seguridad necesarias mediante la aplicación de conceptos como la defensa en profundidad.

Además, la estandarización de los códigos funcionales utilizados por los protocolos OT que permiten llevar a cabo las acciones, combinado con la falta de autenticación y cifrado en muchos de ellos, hacen que los ataques de suplantación, modificación o espionaje de las comunicaciones sean relativamente sencillos para un atacante.

2.5 Listado de activos importantes y vulnerabilidades comunes

Desde el punto de vista de un atacante y para llevar a cabo una auditoría de seguridad es importante destacar cuáles son los elementos más importantes de un ICS y en los que centrar el foco durante el proceso. Obviamente, aquellos dispositivos cuyo compromiso tenga consecuencias más críticas serán de mayor interés. A continuación, se expone una lista de los más relevantes [19]:

- Paneles HMI.
- SCADA.
- *Historians*.
- PLCs.
- RTUs.
- Dispositivos de red.
- Estaciones de trabajo de ingeniería.
- Dispositivos de acceso remoto.
- Servidores de autorización y autenticación.

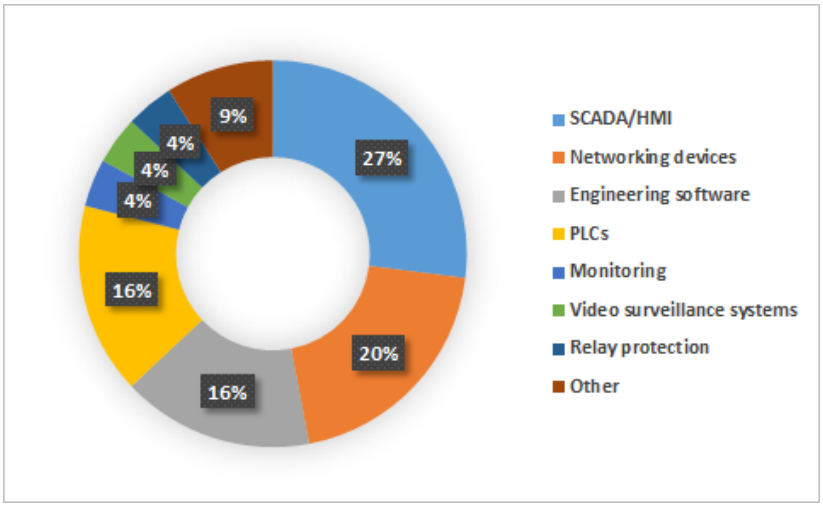


Ilustración 12 - Vulnerabilidades identificadas en componentes de un ICS (2017) (Extraída de: [25])

A su vez, se encuentran una serie de vulnerabilidades que son comunes en estos tipos de componentes de los Sistemas de Control Industrial. Conocerlas permitirá al auditor centrarse en lo importante y saber qué buscar y por dónde dirigir el test de penetración. Algunas de ellas son las siguientes [19] [26] [25]:

- Aplicaciones:
  - Código no seguro mediante el uso funciones inseguras (*Buffer Overflow*, *Integer Overflow*, *Heap Overflow* etc.).
  - Capacidades de seguridad del software no habilitadas.
  - Uso de configuraciones por defecto.
- Servicios web:
  - Acceso sin autenticación
  - Autenticación pobre.
  - Vulnerabilidades web: *Path Transversal*, *Code Injection*, *SQL Injection*, *Cross Site Scripting*, *Cross Site Request Forgery*, etc.
- Protocolos de red:
  - Falta de validación de las entradas: *Buffer Overflow*.
  - Falta de autenticación.

    - Autenticación pobre.
    - Falta de cifrado.
    - Comprobaciones de integridad pobre.
    - Manipulación de los paquetes.
    - Denegación de servicio.
- Actualizaciones:
    - Software antiguo desactualizado.
    - Software con vulnerabilidades conocidas desactualizado.
    - Software de terceros sin soporte.
    - Sistemas operativos no actualizados.
    - Aplicación de actualizaciones sin testeo exhaustivo.
- Autenticación débil:
    - Contraseñas débiles: fuerza bruta o ataque de diccionario.
    - Par de usuario y contraseña por defecto disponible en manuales: ataque de diccionario.
    - Evasión del método de autenticación.
    - Contraseñas filtradas en brechas de seguridad: ataque de diccionario.
    - Configuraciones incorrectas.
    - Envío de usuario y contraseña sin cifrar.
    - Cambio de contraseñas inexistente.
- Violación de permisos:
    - Acceso a recursos sin permisos: *Path Transversal*.
    - Acceso a dispositivos de control remoto sin permiso.
    - Servicios ejecutándose con demasiados permisos innecesarios.
    - Servicios activados por defecto sin ser utilizados (con la consecuente exposición de vulnerabilidades).
- *Information disclousure*:
    - Recursos compartidos accesibles.
    - Insuficiente protección de credenciales.
    - Fugas de información mediante terceros.
- Red: Diseño y configuraciones:
    - Diseño pobre de la red.
    - Segmentación pobre o inexistente.
    - Falta de control del flujo de red.
    - Configuraciones de dispositivos de red por defecto.
    - Cambio de contraseñas de dispositivos de red inexistente.
    - Mala aplicación del control de acceso a los dispositivos de red.
    - Falta de zonas desmilitarizadas o redes perimetrales.
    - Firewall mal configurado: *bypass* del firewall.
    - Falta de monitoreo de la red.
    - Ausencia de IDS y/o IPS.
    - Falta de cifrado en las comunicaciones: *Man-in-the-Middle*.

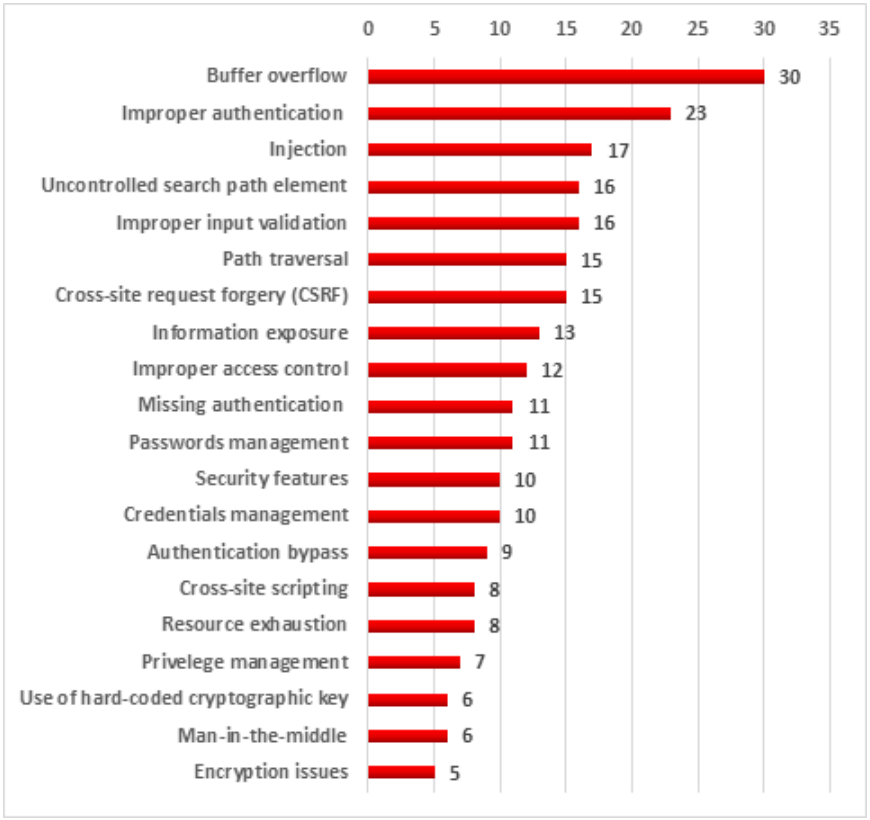


Ilustración 13 - Vulnerabilidades más comunes en ICS (2017) (Extraída de: [25])

3. Conpot

Para poner en práctica el estudio del enfoque de pentesting sobre un Sistema de Control Industrial se utilizará un entorno de pruebas. En dicho entorno se instalará Conpot, un honeypot de baja interacción que simula ser parte de un ICS. En este apartado se redactará información relativa a Conpot: qué es, los datos que se pueden simular y la información que se puede recoger. Por último, se definirá un ejemplo con Conpot para poder llevar a cabo las pruebas sobre el mismo.

3.1 Conpot: un honeypot de ICS

Un honeypot es un sistema vulnerable por definición creado para ser atacado. El objetivo de la instalación de un honeypot puede variar: desde recoger información de los atacantes, como técnicas, origen, metodología, etc.; hasta ser utilizados como sensores o sistemas de alarma que indiquen un ataque a la organización que lo despliega, ya que un acceso o intento de acceso al honeypot será visto como un ataque. Los honeypots se deben situar en zonas desmilitarizadas y bien protegidas. Así, si un atacante consigue el compromiso total del honeypot no podrá comprometer otras redes.

Existen tanto honeypots individuales como honeynets, que son un conjunto de honeypots que representan un escenario más complejo y realista.

Dentro de los honeypots existen dos categorías según su nivel de complejidad. Por un lado, los de alta interacción, representan sistemas más complejos y permiten una interacción mayor con el atacante. Estos son más difíciles de desplegar y configurar y, por lo general, representan aplicaciones y sistemas reales. Deben tener una capa de seguridad extra para que ningún sistema adicional se pueda ver comprometido [27]. Por otro lado, los honeypots de baja interacción emulan servicios y sistemas operativos, son más fáciles de desplegar y configurar, pero permiten una interacción menor con el atacante con la consecuente fácil detección del honeypot por parte del mismo.

Conpot pertenece a este último grupo: es un honeypot de baja interacción. Además, se trata de un honeypot de un entorno OT, con la posibilidad de simular diferentes elementos de diferentes tipos de Sistemas de Control Industrial. Está diseñado para ser fácil de implementar, configurar y desplegar. Permite la conexión de un HMI real y la interacción con hardware real de un ICS.

3.2 Datos simulables

Conpot ofrece diferentes plantillas a la hora de ser desplegado, las cuales permiten simular diferentes elementos de un ICS en diferentes entornos:

- **Default.** Permite simular un PLC Siemens S7-200. Será la plantilla seleccionada para realizar las pruebas de pentesting.
- **Guardian_ast.** Simula un sistema de monitoreo de tanques de gas. Permite monitorizar los niveles de las bombas, sistemas de bombeo, etc.
- **Ipmi.** simula un IPMI-371. Se trata de un dispositivo de Interfaz de Administración de Plataformas Inteligente.

- **Kamstrup_382.** Realiza la simulación de un dispositivo Kamstrup 382 que lleva a cabo la tarea de medidor de energía eléctrica.
- **Proxy.** Realiza la función de proxy.

La plantilla *default* permite desplegar Conpot de manera que simule un PLC Siemens S7-200. Este PLC es ampliamente utilizado en entornos OT. En concreto, esta plantilla desplegará los siguientes protocolos:

Protocolo	Descripción	Puerto
HTTP	Sirve la interfaz web del PLC para diagnóstico del mismo y HMI.	TCP/8800
S7comm	Protocolo propietario de Siemens utilizado para intercambiar datos entre PLCs y acceder a la información del PLC desde otros sistema como SCADAs o HMIs [28].	TCP/10201
Modbus TCP	Protocolo Modbus sobre TCP/IP. Definido anteriormente.	TCP/5020
FTP	Protocolo de transferencia de ficheros.	TCP/2121
IPMI	Intelligent Platform Management Interface. Permite la administración remota y total de un sistema.	UDP/6230
TFTP	Trivial File Transfer Protocol. Similar a una versión básica de FTP utilizado generalmente para transferir pequeños archivos.	UDP/6969
SNMP	Protocolo de administración de red.	UDP/16100
Bacnet	Protocolo utilizado por sistemas de automatización y control de edificios (calefacción, ventilación, aire acondicionado, control de iluminación, de acceso, etc.) para el intercambio de información.	UDP/47808

Tabla 5 - Protocolos de la plantilla *default* de Conpot (Siemes S7-200)

La asignación de los puertos a los protocolos no está asignada según el estándar normal de puertos y servicios. Así, al protocolo HTTP le corresponderá el puerto TCP/8800 en vez del TCP/80; a S7Comm el TCP/10201 en vez del TCP/102; a Modbus el TCP/5020 en vez del TCP/502; a FTP el TCP/2121 en vez del TCP/21. Lo

mismo ocurre con alguno de los protocolos que trabajan sobre UDP: a IPMI el UDP/6230 en vez del UDP/623 y a TFTP el UDP/6969 en vez del UDP/69.

3.3 Datos recolectables

Recolectar datos a partir del honeypot no es el objetivo de este trabajo, por lo que no se profundizará en este aspecto. Se indicará, a modo de ejemplo, qué información se puede recoger y dónde encontrarla.

Conpot permite recoger toda la información relativa a las interacciones con el mismo: IPs de conexión, puertos, protocolos, errores, etc. Aun así, por sí solo, no ofrece un sistema de correlación de datos ni paneles donde poder visualizar de una forma amigable los datos recogidos. Todo ello se realiza mediante un fichero de logs donde Conpot registra la información: *conpot.log*. Esta puede ser guardada en diferentes formatos, como texto plano, JSON o en un a Base de Datos; según se indique en el fichero de configuración, en este caso situado en: */home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/testing.cfg*. El log podrá ser encontrado en el directorio desde el cual se lance Conpot.

```
2019-03-29 09:35:18,772 HTTP/1.1 GET request from ('172.17.0.1', 46104): ('/testing-http', [('Host', '172.17.0.2:8800'), (
'User-Agent', 'curl/7.64.0'), ('Accept', '*/*')], None). bdc8f9a4-09ad-4e6e-beb9-2535f7b36a47
2019-03-29 09:35:18,774 HTTP/1.1 response to ('172.17.0.1', 46104): 404. bdc8f9a4-09ad-4e6e-beb9-2535f7b36a47
```

Ilustración 14 - Extracto de conpot.log

3.4 Definición de un ejemplo con Conpot

En este apartado se definirá un ejemplo con Conpot. Dicha definición o escenario será el posteriormente utilizado para realizar las pruebas de concepto una vez se haya definido el enfoque de pentesting para Sistemas de Control Industrial. Es importante recalcar que no se hará un uso convencional del honeypot, entendiéndose como convencional el uso del honeypot para detectar posibles ataques o para investigar acerca de las metodologías utilizadas por los atacantes, tal y como se ha comentado anteriormente en este capítulo, en concreto, en el punto *3.1 Conpot: un honeypot de ICS*. En su lugar, se utilizará el honeypot para realizar una Prueba de Concepto en forma de pentest y poder poner en práctica la metodología o enfoque de pentesting para ICSs desarrollado a lo largo de la memoria.

Para la definición del ejemplo con Conpot se hará uso de Docker. Se creará un contendor dentro de la máquina virtual atacante situada de una Intranet creada en VirtualBox. La comunicación entre el host de Docker (máquina atacante) y el contenedor (donde se desplegará Conpot) se hará a través de una conexión puente o *bridge* entre ambos. Así, Docker creará automáticamente la interfaz de red *docker0* y realizará las asignaciones de direcciones IP.

```
root@kali:~# ifconfig docker0
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:fcff:fef3:55d8  prefixlen 64  scopeid 0x20<link>
        ether 02:42:fc:f3:55:d8  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 936 (936.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Ilustración 15 - Interfaz de red "docker0" tras desplegar el contenedor

Una vez iniciado el servicio correspondiente a Docker en la máquina atacante, se procederá a desplegar el contenedor. En este caso, debido a un problema con las interfaces de red y la imposibilidad de cambiar los puertos de Conpot, se ha decido mapear los puertos de Conpot a los de la máquina local. Es decir, accediendo a un puerto de la máquina atacante se accederá directamente a otro de Conpot, mediante *port forwarding*, actuando como un puente entre máquinas. Así, los asociación entre puertos y servicios cumplirán el estándar por defecto (p. ej., la plantilla por defecto utiliza el puerto tcp/5020 para Modbus TCP, con el cambio introducido mapeando el puerto 5020 de Conpot al puerto 502 de la máquina local el problema queda solventado). Para lograr esto, a la hora de desplegar el contenedor se empleará la opción `-p puerto_host:puerto_container`. Además, se empleará la opción `-i` y `-t` del comando `run` de Docker para mantener la entrada (STDIN) abierta y para mostrar una pseudo-TTY, respectivamente.

```
root@kali:~/conpot# service docker start
root@kali:~/conpot# docker run -it -p 502:5020 -p 102:10201 -p 80:8800 -p 21:2121 -p 161:16100/udp
-p 69:6969/udp -p 47808:47808/udp -p 623:6230/udp -p 44818:44818  --network=bridge honeynet/conpot:
latest /bin/sh
~ $
```

Ilustración 16 - Inicialización del servicio docker y despliegue del contenedor

Posteriormente, debe ser levantado Conpot. En este caso se utilizará el fichero de configuración por defecto y, como se ha estipulado, se utilizará la plantilla *default* que simula un PLC Siemens S7-200.

```
~/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg $ bin/conpot -f --template conpot/templates/default
WARNING:scapy.runtime:No route found for IPv6 destination :: (no default route?)

                 _
 ___ ___ ___ ___| |_
|  _| . |   | . | . |  _|
|___|___|_|_|  _|___|_|
            |_|

 Version 0.6.0
 MushMush Foundation
```

Ilustración 17 - Despliegue de Conpot dentro del contenedor

A continuación, se mostrará en el log el despliegue de los servicios y puertos: Conpot estará listo para ser atacado.

```
2019-04-01 13:51:06,049 Modbus server started on: ('0.0.0.0', 5020)
2019-04-01 13:51:06,050 S7Comm server started on: ('0.0.0.0', 10201)
2019-04-01 13:51:06,050 HTTP server started on: ('0.0.0.0', 8800)
2019-04-01 13:51:06,807 SNMP server started on: ('0.0.0.0', 16100)
2019-04-01 13:51:06,809 Bacnet server started on: ('0.0.0.0', 47808)
2019-04-01 13:51:06,810 IPMI server started on: ('0.0.0.0', 6230)
2019-04-01 13:51:06,811 handle server PID [   22] running on ('0.0.0.0', 44818)
2019-04-01 13:51:06,811 handle server PID [   22] responding to external done/disable signal in object 139981437552200
2019-04-01 13:51:06,812 FTP server started on: ('0.0.0.0', 2121)
2019-04-01 13:51:06,812 Starting TFTP server at ('0.0.0.0', 6969)
```

Ilustración 18 - Conpot desplegado: servicios y puertos

En la máquina atacante, debido al mapeo de puertos, deberán quedar abiertos los puertos establecidos en el despliegue del contenedor:

```
root@kali:~/conpot# netstat -lntpu | grep docker
tcp6       0      0 :::102                  :::*                    LISTEN      21264/docker-proxy
tcp6       0      0 :::80                   :::*                    LISTEN      21278/docker-proxy
tcp6       0      0 :::44818                :::*                    LISTEN      21222/docker-proxy
tcp6       0      0 :::21                   :::*                    LISTEN      21318/docker-proxy
tcp6       0      0 :::502                  :::*                    LISTEN      21306/docker-proxy
udp6       0      0 :::623                  :::*                                21236/docker-proxy
udp6       0      0 :::47808                :::*                                21208/docker-proxy
udp6       0      0 :::69                   :::*                                21292/docker-proxy
udp6       0      0 :::161                  :::*                                21250/docker-proxy
```

Ilustración 19 - Puertos abiertos en la máquina atacante

De esta forma, a la hora de realizar la auditoría de seguridad contra Conpot, se llevará a cabo a través de la máquina atacante. En la máquina atacante no hay ningún puerto abierto; ahora únicamente estarán abiertos aquellos que sirven como puente hacia el contenedor de Conpot. Por tanto, aunque la IP que se ataque sea `127.0.0.1` técnicamente se estará atacando al Conpot. Esto se podría realizar creando una máquina virtual que actuara como docker host en la que se desplegara el contenedor de docker y se realizara todo este mapeo, y luego tener a parte otra máquina virtual a parte que fuera la atacante. Pero por motivos de potencia computacional se ha tenido que realizar de la manera redactada, en una única máquina virtual, siendo más que suficiente para realizar la prueba de concepto.

Así, el esquema final de la definición del ejemplo con Conpot quedará de la siguiente manera:

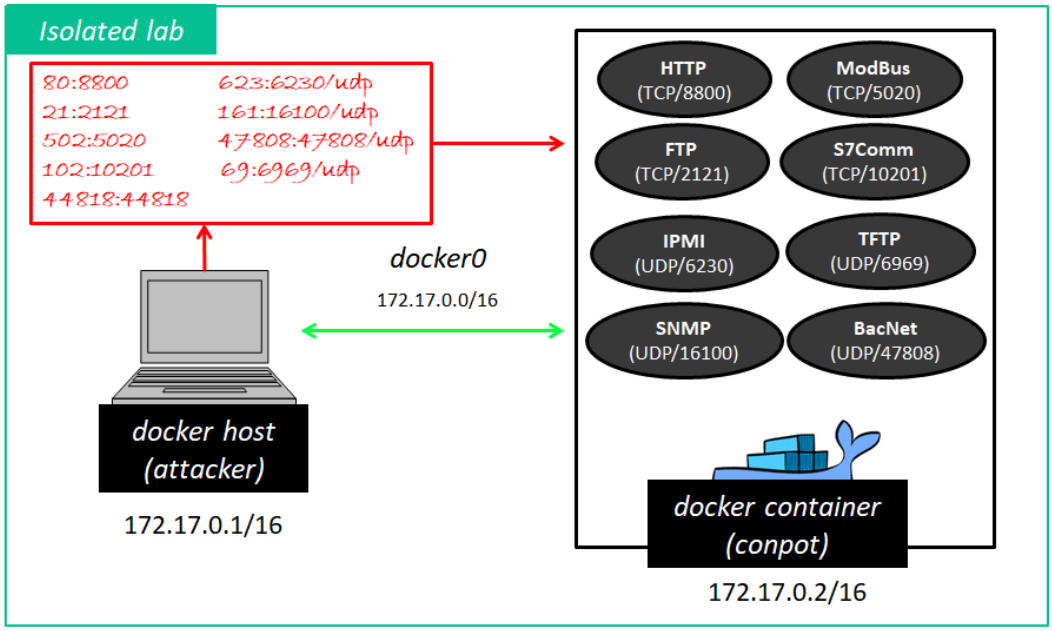


Ilustración 20 - Definición de ejemplo con Conpot

4. Pentesting en Sistemas de Control Industrial (ICS)

Una vez conocida la tecnología que se desea auditar; en este caso los componentes, protocolos y peculiaridades de los Sistemas de Control Industrial, ya es posible atacar dichos sistemas. Así, en este capítulo se realizará el estudio del enfoque de pentesting para Sistemas de Control Industrial desarrollando las fases a seguir para llevar a cabo una auditoría de seguridad contra estos sistemas.

4.1 Fases de un pentest IT

En este apartado se describirán brevemente las fases de un pentest contra sistemas de una red IT, es decir, un pentest convencional realizado sobre sistemas de la red empresarial. El pentest a un Sistema de Control Industrial tendrá partes semejantes con este, es por ello que se citan las fases principales a continuación. Si se desea profundizar más en las técnicas de ataque empleadas se puede consultar el framework de MITRE: ATT&CK [29], pero no es el objetivo de este texto.

- **Acciones de pre-contratación.** En esta fase se detallarán los objetivos de la auditoría, el enfoque de la misma y los métodos a utilizar. También se establecerán los contratos legales.

- **Recogida de información.** También conocida como *Information Gathering* o Reconocimiento. En ella se pretende recoger toda la información que sea posible para poder utilizarla en fases posteriores para comprometer el o los sistemas a auditar. Esta fase es crucial, ya que, cuanta más información se posea mayor será la probabilidad de éxito del ataque. En esta fase se utilizarán diferentes técnicas englobadas en los siguientes grupos:

  - *External fingerprinting* pasivo u OSINT. La información se obtiene desde fuera de la organización y de forma indirecta, sin interactuar directamente con el objetivo, desde fuentes abiertas: información indexada en buscadores web, redes sociales, antiguas brechas de seguridad, etc.

  - *External fingerprinting* activo. Más agresiva que la anterior. La información se recoge desde fuera de la organización pero interactuando directamente con el sistema o sistemas a auditar: consultas DNS, análisis de cabeceras HTTP, escaneo de puertos, etc.

  - *Internal fingerprinting*. Similar al *external fingerprinting* activo pero, en este caso, ya se cuenta con acceso parcial o total a la red a atacar.

- **Identificación de vulnerabilidades.** Una vez se ha obtenido gran cantidad de información sobre el objetivo, el siguiente paso es identificar las amenazas y vulnerabilidades que permitirán llevar a cabo el ataque con éxito. En este punto se identificarán las vulnerabilidades a partir de la información de la anterior fase: versiones de servicios sin actualizar, vulnerabilidades en servicios, malas configuraciones, etc.; acompañado de escáneres de vulnerabilidades. A partir de todo esto se elaborarán los diferentes vectores de ataque posibles que utilizar en la fase de explotación.

- **Explotación.** Con los vectores de ataque y los puntos de entrada posibles identificados, en esta fase se pretende explotarlos y conseguir acceso al sistema. Algunas de las técnicas pueden ser: ataques a aplicaciones web, ataques de red, ingeniería social (p. ej. *spear phising*), etc. En todo momento se intentará evadir los sistemas de defensa y detección. Se emplearán exploits públicos, y en otras ocasiones habrá que modificarlos o crearlos desde cero.

- **Post explotación.** Una vez establecido acceso en el sistema, el siguiente paso es recolectar toda la información posible del mismo: ficheros importantes, contraseñas, información del sistema, situación en la red, etc. Si no se ha conseguido acceso como administrador se intentará escalar privilegios en el sistema. Otro de las tareas posibles en esta fase es el movimiento lateral o *pivoting* hacia otras redes a las que tiene acceso el sistema comprometido y no se tiene acceso desde la máquina atacante, aunque todo ello dependerá de los límites establecidos en la definición de la auditoría. También se intentará establecer persistencia y se procederá a eliminar los *logs* y todo rastro dejado durante el pentest.

- **Reporting.** Fase de documentación de todo el proceso de la auditoría que será entregada al cliente. Es aconsejable incluir una tabla con los hallazgos y los riesgos asociados emitiendo una puntuación a cada uno de ellos. La documentación estará conformada por un resumen ejecutivo y un informe técnico más detallado.

4.2 ICS Cyber Kill Chain

Para entender mejor como llevar a cabo la auditoría, es preciso también analizar la "cadena de ataque" o "Cyber Kill Chain" utilizadas por los ciberatacantes en entornos industriales. Ya que el objetivo de un pentest es actuar tal y como lo haría un atacante, conocer las fases del ataque es clave para el desarrollo del mismo.

El concepto de Cyber Kill Chain fue definido por analistas de *Lockheed Martin Corporation* en el año 2011 [30]; se trata de un conjunto de pasos dirigidos contra un objetivo en un ataque avanzado, donde la mitigación de alguno de los pasos de la cadena supondría la parada del ataque. A modo gráfico, los pasos de la cadena son los siguientes:

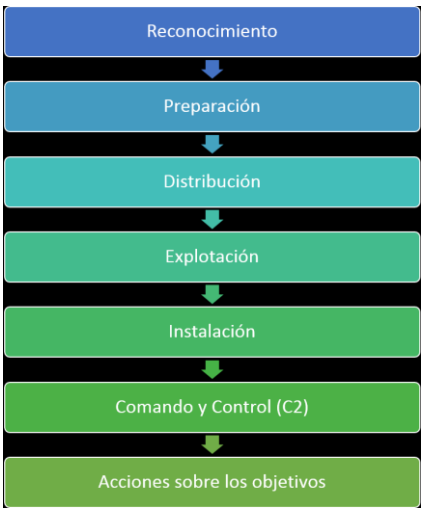


Ilustración 21 - Fases de Cyber Kill Chain (Extraída de: [30])

Vistas en anteriores capítulos las claras diferencias entre el mundo IT y OT, es correcto afirmar que los conocimientos necesarios para realizar un ataque a un ICS con éxito son claramente mayores. Por ello, la Cyber Kill Chain, orientada más hacia el mundo IT o núcleo empresarial, no será del todo acertada para entornos industriales. Debido a esto, el *SANS Institute* elaboró un informe en 2015 [31] donde adaptó la Cyber Kill Chain a entornos ICS: ICS Cyber Kill Chain. Así, dividen la cadena en dos etapas.

4.2.1 Etapa 1

Según los autores del ICS Cyber Kill Chain [31], la primera etapa está catalogada como espionaje industrial o inteligencia. Esta etapa es muy similar a la Cyber Kill Chain original. Su objetivo es conseguir información sobre el ICS, sobre el sistema en su conjunto y así poder elaborar mecanismos para romper la seguridad perimetral y ganar acceso a entornos de producción. Esta etapa se divide en las siguientes fases:

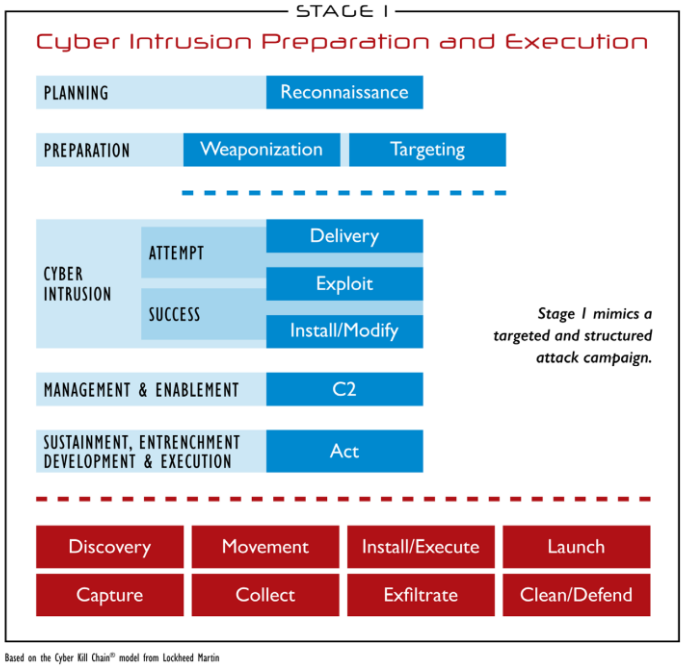


Ilustración 22 - Etapa 1 de ICS Cyber Kill Chain (Extraída de: [31])

- **Planificación.** Fase de reconocimiento. El objetivo de esta fase es encontrar debilidades e información que ayuden al atacante en su esfuerzo por elegir y explotar el objetivo. Información sobre empleados, la red, el host, protocolos, políticas, procesos, procedimientos, etc. se recabarán en esta fase; además de información técnica sobre el ICS, el funcionamiento de la planta y posibles vulnerabilidades. Además, se emplearán también técnicas OSINT.

- **Preparación.** Segunda fase de la primera etapa en la que se pretende preparar el camino de la intrusión. Puede estar formada por "el armamento" y "la fijación del objetivo", aunque pueden resultar prescindibles dependiendo de las circunstancias del ataque. El armamento se refiere a la modificación de archivos no maliciosos que permitan al atacante pasar a la siguiente fase (p. ej. inserción de exploit en fichero PDF, macros maliciosas, etc.); la fijación del

objetivo, que puede ocurrir también en esta fase, corresponde a la identificación de víctimas potenciales y la decisión de qué herramientas y métodos utilizar estableciendo una balanza entre esfuerzo, tiempo, probabilidad de éxito y riesgo de detección.

- **Intrusión.** Fase en la que se pretende lograr acceso. Se trata de cualquier intento por parte del atacante, exitoso o no, de acceso a la red objetivo. Incluye el paso de "Entrega", en el que el atacante emplea cualquier método que le permita interactuar con la red de la víctima (a través de un correo utilizando técnicas de *phising*, por ejemplo, en el que envía el fichero malicioso de la anterior fase relativa al armamento). Incluye también el paso de la "Explotación", en la que el atacante llevará a cabo acciones maliciosas (apertura del fichero malicioso por parte de la víctima, siguiendo el ejemplo, o acceso a VPN mediante credenciales, etc.). Si la explotación tiene éxito tendrá lugar la "Instalación y modificación", donde el atacante instalará o modificará cualquier fichero en el sistema atacado para asegurar futuros accesos (troyanos, uso de Powershell, etc.).

- **Gestión y habilitación.** En esta fase se realiza la gestión de los accesos a través del establecimiento de sistemas de comando y control (abusando de comunicaciones confiables como VPN, conexiones a las capacidades previamente instaladas al final de la anterior fase, etc.).

- **Logística, fortificación, desarrollo y ejecución.** En esta fase el atacante ya tiene un buen conocimiento sobre el sistema comprometido y llevará a cabo una serie de acciones: instalación de capacidades adicionales, descubrimiento de equipos, movimiento lateral entre redes, exfiltración de información, borrado de huellas, etc. Se trata de una fase crítica para poder pasar a la etapa 2 de manera satisfactoria donde se identificarán y estudiarán los componentes del ICS a atacar.

Es importante aclarar que el atacante puede dirigir la etapa 1 contra un proveedor o colaborador del ICS para conseguir la información necesaria como rutas relativas a archivos del ICS o acceso remoto a componentes del ICS. La etapa 1 es muy similar a una brecha de seguridad en una red IT. Aquí finaliza la etapa 1, cuando el atacante ha comprometido con éxito la red objetivo, dando paso a la etapa 2.

4.2.2 Etapa 2

En esta etapa el atacante aprovechará la información recogida en la etapa 1 para desarrollar y probar una capacidad que pueda atacar de manera directa al ICS. Consta de las siguientes fases:

- **Desarrollo y ajuste del ataque.** El atacante desarrolla una capacidad (herramienta, método, etc.) que pretende afectar al Sistema de Control Industrial con el impacto deseado. Lo normal es que no se experimente directamente en el proceso de producción para llevar a cabo los ajustes necesarios; lo que hace esta fase difícil de detectar. Es por la necesidad de todos estos ajustes que puede existir un retardo de tiempo entre la etapa 1 y esta etapa.

- **Validación.** En esta fase el atacante validará la capacidad desarrollada en un entorno similar al que se desea atacar para asegurarse de tener un impacto

significativo. En algunas ocasiones el atacante adquirirá software y componentes físicos para llevar a cabo las pruebas en un laboratorio.

- **Ataque.** En este último punto el atacante descargará la capacidad desarrollada en el sistema objetivo, realizará la instalación o modificaciones pertinentes en funcionalidades del sistema y ejecutará el ataque. El ataque puede tener varias fases, como la habilitación, iniciación y soporte del ataque, que activen las condiciones necesarias para que el ataque tenga el efecto deseado o, simplemente, éxito. Según INCIBE [30], las consecuencias habituales que se dan en un ataque sobre sistemas de control son la pérdida (de datos o de control), la denegación (habitualmente de servicio) y la manipulación (de datos, de visualizaciones, etc.). En última instancia, el atacante debe manipular el proceso para causar un daño significativo como la destrucción física, el daño del equipo bajo control o elementos del proceso.

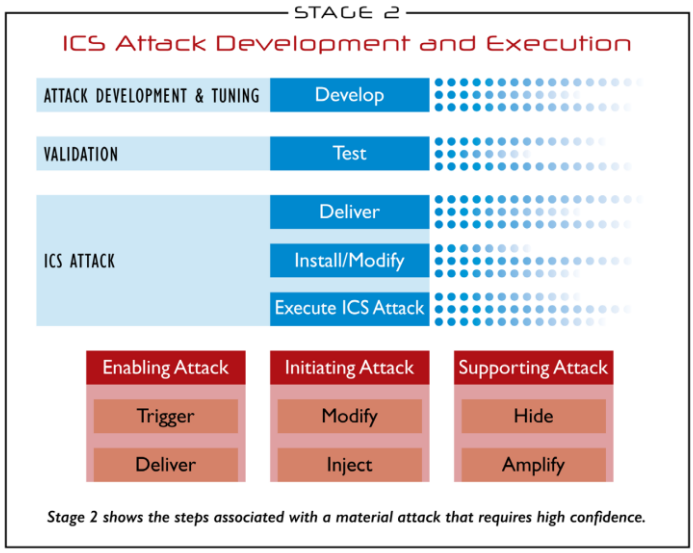


Ilustración 23 - Etapa 1 de ICS Cyber Kill Chain (Extraída de: [31])

4.2 Enfoque de pentesting en ICS

Repasados los puntos de un test de penetración convencional se procederá a detallar los pasos o fases a seguir durante el desarrollo de una auditoría de seguridad orientada a Sistemas de Control Industrial. Gracias a la explicación de la ICS Cyber Kill Chain, se tendrá una visión más general del proceso. En este apartado, núcleo del proyecto, se pretende detallar el estudio de un enfoque de pentesting sobre ICS haciendo énfasis en las fases y la metodología a seguir en cada una de ellas, de una manera genérica y global, ya que cada entorno es diferente, mostrando el abanico de herramientas y sus posibilidades pero nunca siendo una guía de uso de las mismas.

4.2.1 Acceso a la red OT

El primer paso es el acceso a la red OT donde se encuentra el Sistema de Control Industrial. Dependiendo del tipo de auditoría existen tres posibilidades:

1. **Ya se tiene acceso a la red OT.** La auditoría parte por este punto y no es necesario comprometer la red empresarial desde fuera para pivotar hasta la

red OT, por ejemplo. Así, el objetivo de la auditoría de seguridad es comprobar la seguridad de la red OT y saber cuáles son las posibilidades de un atacante una vez tenga acceso a la red OT.

2. **No se tiene acceso a la red IT.** La auditoría parte desde el principio, tal y como lo haría un atacante desde el exterior; desde la etapa 1 de la ICS Cyber Kill Chain. El hacker ético podría seguir diferentes caminos para llegar a la red OT, entre ellos:

    a. Comprometer la red empresarial por completo y pivotar a la red industrial siguiendo las fases de un pentest convencional. Esta casuística es explicada en el punto 3.

    b. Acceder directamente a los componentes del ICS si alguno de ellos está expuesto a Internet. Esto se puede llevar a cabo mediante herramientas como Shodan [32] o ZoomEye [33], que son buscadores de sistemas expuestos en Internet; o incluso en ocasiones a través de buscadores clásicos como Bing o Google. En este caso, se realizarían búsquedas referentes a la empresa del ICS objetivo de la auditoría.

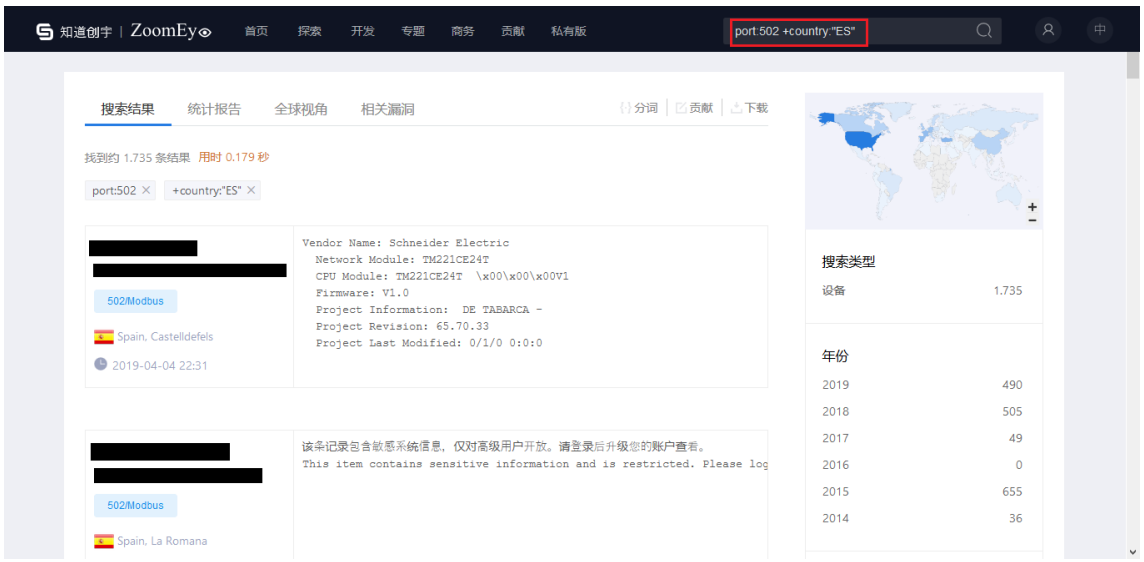


Ilustración 24 - Puerto 502 (Modbus) en España a través de ZoomEye

    c. Llevar a cabo una fase de reconocimiento, embeber un exploit en un fichero malicioso y utilizar técnicas de ingeniería social para entregarlo y que sea abierto por miembros de la red OT directamente (p. ej.: entrega: memorias USB cerca de la planta, comprometer el sistema personal de un ingeniero que es posteriormente conectado a la red OT, *spear phising* dirigido a empleados, etc.; método: macros con malware en documentos Office, inserción de exploits en PDFs, etc.). De esta forma la brecha se produciría directamente en la red OT y no sería necesario todo el compromiso de la red de la organización y el movimiento lateral hasta la misma.

    d. Dirigir la etapa 1 de la ICS Cyber Kill Chain contra vendedores/proveedores de ICS cuyos clientes son el Sistema de Control Industrial que se desea atacar. De esta forma, se podría comprometer, por ejemplo, la página web que es visitada desde la red

OT sobre dicho vendedor comprometido y así comprometer también la red OT del ICS. Esta técnica, conocida como *watering hole*, utilizada por atacantes, puede que no sea posible ya que implica el compromiso del vendedor que es ajeno al ICS; aunque todo depende del enfoque de la auditoría.

3. **Se ha comprometido la red IT.** Es un caso bastante común a la hora de enfocar una auditoría contra ICSs. Se parte del punto en el que se ha comprometido la red empresarial por completo, teniendo permisos de administrador del dominio en la misma. El objetivo de esta auditoría es saber qué podría llegar a hacer un atacante en caso de comprometer por completo la red de la organización; comprobar si es capaz de llegar a la red OT. Así, un factor clave será cómo de segmentada esté la red. Los pasos a llevar a cabo por el hacker ético se pueden resumir en los siguientes [13]:

    I. Recogida de información. Se pretende recoger todo tipo de información en la red IT relacionada con el ICS:

        i. Información de la red OT. Toda información que pueda aumentar el conocimiento acerca de la red OT: marcas, vendedores, componentes, protocolos, propósito, tipo de ICS, etc. Información de red: diagramas de red, documentación de la arquitectura, direcciones IPs, nombres de máquinas, información sobre dispositivos del ICS, etc.

        ii. Usuarios/recursos clave. En el caso de existir un Directorio Activo en la red IT, buscar usuarios clave, como ingenieros u operadores, que puedan tener información vital acerca del ICS, como nombres de servidores, direcciones IP, estaciones de trabajo, carpetas compartidas en red, documentación de la arquitectura de red, etc.; toda información que proporcione un camino hacia la red OT. Este tipo de usuarios también suelen tener permisos para acceder a la red OT desde la red IT (p. ej. VPN), por lo que pueden ser críticos en el proceso de movimiento lateral para llegar a la red operacional. Así, se deberán buscar también documentos que muestren cómo conseguir este propósito, llegando a estar en muchos casos información sensible como usuarios y contraseñas expuestas en texto plano.

        iii. Información de dispositivos de red. Backups de firewalls, routers, tablas de enrutamiento, etc. pueden revelar información muy útil acerca de cómo fluye el tráfico y la dirección del mismo, qué hosts tienen acceso a la red OT, a través de qué protocolos (SSH, VNC, RDP…), información sensible como nombres de usuarios, hashes de contraseñas, etc.

        iv. Identificación de equipos dual-homed. Se trata de sistemas con más de una interfaz de red que son capaces de comunicarse tanto con la red IT como con la red OT, con reglas especiales en el firewall que les permite el paso del tráfico. Estos sistemas son realmente interesantes ya que debido a sus características su compromiso puede proporcionar un acceso directo a la red OT o

información acerca de segmentos de red de la red operacional. Ejemplo de ello sería un Historian que tiene que comunicarse con la red OT para la recogida de datos y con la red IT para la transmisión de los mismos.

II. Identificación del punto de entrada a la red OT. Teniendo en cuenta toda la información recogida, se buscarán las grietas que permitan adentrarse en la red operacional. Generalmente, la red OT no suele estar completamente aislada de la red corporativa ya que suele estar situada en lugares remotos o de difícil acceso por lo que es común el acceso remoto para la administración de dispositivos y sistemas de la red operacional (SSH, VNC, RDP, Citrix, etc.). La segmentación de la red, las políticas de acceso y los firewalls, entre otros, jugarán también un papel importante en la dificultad del compromiso. Con toda la información recogida se deberá elaborar una estrategia de acceso.

El acceso a la red OT pone fin a esta primera fase. En caso de no tener acceso inicial (caso 2), existen numerosas técnicas, tal y como se ha citado en el punto 2 de este apartado. Algunas de ellas están basadas en incidentes de seguridad reales sobre ICSs como Stuxnet [34] o Havex [35]. No por ello son las únicas técnicas posibles, el límite está en la imaginación, el esfuerzo, tiempo y limitaciones del hacker ético y la auditoría de seguridad. Por otra parte, en el caso de haber comprometido la red corporativa (caso 3) se ha definido una metodología clara basada en la recogida de información, identificación de vulnerabilidades y elaboración de un plan de acceso. Existen numerosas técnicas a llevar a cabo y dependerán directamente de todos los factores y características del sistema a auditar.

Así, al finalizar esta fase el auditor se situaría, a nivel de red, en el nivel 2 del modelo jerárquico ISA-95:

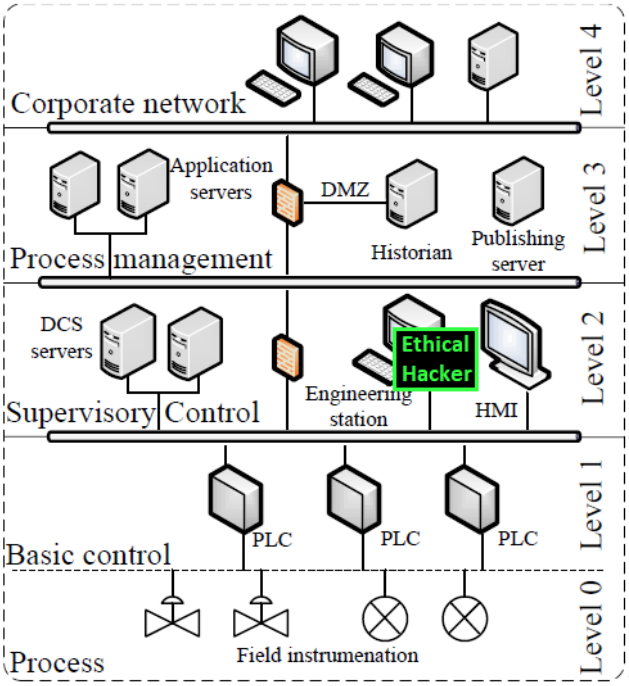


Ilustración 25 - Situación del hacker ético tras acceder a la red OT, sobre ISA-95 (Extraída/modificada de: [23])

4.2.2 Reconocimiento

Una vez se tenga acceso a la red operacional, el primer paso es el reconocimiento. En caso de haber llegado a este punto a través de un ataque directo a un sistema de la red OT (p. ej. *spear phising* ingeniero + PDF malicioso) se deberá recoger información acerca del sistema comprometido para ver en qué situación se encuentra el auditor dentro del esquema de red (suponiendo que en el caso de haber llegado a la red OT a través del compromiso de la red IT se sabe ya en qué sistema o situación se está; sino también habrá que llevar a cabo este estudio): información acerca de tarjetas de red, información sobre el sistema: nombre del equipo, puertos abiertos, conexiones, logs, tablas de enrutamiento, reglas del firewall, etc.

El siguiente paso es la identificación del resto de dispositivos a los que se tiene acceso a través de las interfaces de red. Esto se puede llevar a cabo mediante escáneres de red convencionales utilizados para auditorías de seguridad de redes IT o escáneres específicamente desarrollados para este tipo de auditorías en redes OT.

4.2.2.1 Escáneres convencionales

Ejemplos de este tipo de escáneres de redes son el escáner por antonomasia Nmap[7] o Masscan[8]. Este último es muy rápido, muy útil para grandes redes, pudiendo escanear Internet en 6 minutos transmitiendo hasta 10 millones de paquetes por segundo.

```
root@kali:~# nmap -sP 10.0.2.14/24
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-16 11:43 CEST
Nmap scan report for 10.0.2.1
Host is up (0.00027s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.2
Host is up (0.00019s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00014s latency).
MAC Address: 08:00:27:23:CF:44 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.9
Host is up (0.00029s latency).
MAC Address: 08:00:27:39:B5:70 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.19
Host is up (0.00042s latency).
MAC Address: 08:00:27:CD:41:15 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.21
Host is up (0.00051s latency).
MAC Address: 08:00:27:7B:E0:5B (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.14
Host is up.
Nmap done: 256 IP addresses (7 hosts up) scanned in 28.05 seconds
```

Ilustración 26 - Descubrimiento de dispositivos en una red mediante Nmap

En este punto se tendrá una lista de IPs correspondientes a dispositivos situados en el nivel 2 y 1 del modelo jerárquico de ISA-95: HMIs, SCADAs, DCSs, PLCs, RTUs, IEDs, etc., respectivamente. El siguiente paso será la identificación de los dispositivos,

[7] Página oficial de nmap: https://nmap.org/
[8] Github de masscan: https://github.com/robertdavidgraham/masscan

ya que hasta ahora son solo una lista de IPs. Esto se efectuará a través de escáneres de red, al igual que anteriormente, pero esta vez no se utilizarán para descubrir dispositivos sino para detectar puertos abiertos y servicios, tanto TCP como UDP, corriendo en cada uno de ellos. Así, el descubrimiento de según qué protocolos permitirá al hacker ético identificar el tipo de componente, lo que capacitará la focalización del ataque.

Un punto importante a tener en cuenta es, como apunta Bolívar [19], considerar que este tipo de dispositivos usan una serie de protocolos específicos. Por este motivo, un escaneo por defecto de nmap no lo realizará sobre estos puertos, por lo que o bien habrá que especificarlos manualmente (también hay que tener en consideración que es posible que los servicios no estén asociados a los puertos por defecto) o realizar un escáner de todos los puertos del dispositivo a escanear, desde el 1 al 65535.

Protocolo	Puerto
Modbus TCP	TCP/502
Modbus RTU	TCP/2000
EtherNET/IP	UDP/2222 TCP/44818
Profinet	TCP/ 34692-34964
FieldBus	TCP/1089- 1091
DNP 3.0	TCP/20000
Bacnet	UDP/47808
Siemens S7	TCP/102

Tabla 6 - Recordatorio de puertos y servicios industriales comunes (Basada en: [19])

Aun así, tampoco hay que olvidar los protocolos comunes en un dispositivo IT, ya que estos componentes también pueden tener activos dichos protocolos (véase, por ejemplo, un servidor web en un HMI usando el protocolo http, un protocolo de administración remota como SSH, un servidor FTP o web en un PLC, etc.).

```
root@kali:~# nmap -p- 10.0.2.17
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-16 12:37 CEST
Nmap scan report for 10.0.2.17
Host is up (0.00022s latency).
Not shown: 65531 closed ports
PORT    STATE SERVICE
21/tcp  open  ftp
80/tcp  open  http
102/tcp open  iso-tsap
502/tcp open  mbap
MAC Address: 08:00:27:F1:44:47 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 16.94 seconds
```

Ilustración 27 - Escaneo de todos los puertos (TCP) de un disp. de un ICS mediante Nmap

La potencia de nmap no acaba aquí, ofreciendo numerosos scripts para la obtención de mayor información o vulnerabilidades en el servicio detectado, lo que facilitará la elaboración de posteriores ataques. Dichos scripts se pueden encontrar, en el caso de Kali Linux, bajo la ruta `/usr/share/nmap/scripts`. Por defecto, nmap cuenta con pocos scripts para protocolos ICS, pero cualquier usuario puede crear los suyos. En este caso, haciendo una búsqueda web, se han encontrado y descargado diferentes paquetes de scripts orientados a ICSs: nmap-scada [36], scada-tools [37] y Redpoint [38] que complementarán a los incluidos por defecto en nmap (`modbus-discover.nse`, `s7-info.nse`, `pcworx-info.nse` y `enip-info.nse`.).

```
root@kali:~/Desktop/ics# find . -name "*.nse"
./nmap-scada-master/Siemens-HMI-miniweb.nse
./nmap-scada-master/Siemens-Scalance-module.nse
./nmap-scada-master/Siemens-CommunicationsProcessor.nse
./nmap-scada-master/Siemens-WINCC.nse
./nmap-scada-master/Siemens-SIMATIC-PLC-S7.nse
./Redpoint-master/cspv4-info.nse
./Redpoint-master/modicon-info.nse
./Redpoint-master/dnp3-info.nse
./Redpoint-master/codesys-v2-discover.nse
./Redpoint-master/omronudp-info.nse
./Redpoint-master/atg-info.nse
./Redpoint-master/proconos-info.nse
./Redpoint-master/pcworx-info.nse
./Redpoint-master/BACnet-discover-enumerate.nse
./Redpoint-master/enip-enumerate.nse
./Redpoint-master/s7-enumerate.nse
./Redpoint-master/omrontcp-info.nse
./Redpoint-master/fox-info.nse
./scada-tools-master/mms-identify.nse
./scada-tools-master/iec-identify.nse
```

Ilustración 28 - Listado de scripts ICS de nmap descargados

```
root@kali:/usr/share/nmap/scripts# nmap --script=+s7-enumerate -p102 172.17.0.2
Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-01 19:42 CEST
Nmap scan report for conpot (172.17.0.2)
Host is up (0.000050s latency).

PORT      STATE SERVICE
10201/tcp open  iso-tsap
| s7-enumerate:
|   Version: 0.0
|   System Name: Technodrome
|   Module Type: Siemens, SIMATIC, S7-200
|   Serial Number: 88111222
|   Plant Identification: Mouser Factory
|_  Copyright: Original Siemens Equipment
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: Device: specialized

Nmap done: 1 IP address (1 host up) scanned in 1.38 seconds
```

Ilustración 29 - Uso de script de nmap orientados a ICS (prot. S7Comm)

El framework Metasploit también dispone de una serie de módulos auxiliares que podrán ser de ayuda en esta fase de reconocimiento. Por tanto, no se debe olvidar esta opción en esta fase, aunque juegue un papel más importante en etapas posteriores.

```
Name                                                  Disclosure Date  Rank       Check  Description
----                                                  ---------------  ----       -----  -----------
auxiliary/scanner/scada/digi_addp_reboot                               normal     Yes    Digi ADDP Remote Reboot Initiator
auxiliary/scanner/scada/digi_addp_version                              normal     Yes    Digi ADDP Information Discovery
auxiliary/scanner/scada/digi_realport_serialport_scan                  normal     Yes    Digi RealPort Serial Server Port Scanner
auxiliary/scanner/scada/digi_realport_version                          normal     Yes    Digi RealPort Serial Server Version
auxiliary/scanner/scada/indusoft_ntwebserver_fileaccess                normal     Yes    Indusoft WebStudio NTWebServer Remote File Access
auxiliary/scanner/scada/koyo_login                    2012-01-19       normal     Yes    Koyo DirectLogic PLC Password Brute Force Utility
auxiliary/scanner/scada/modbus_findunitid             2012-10-28       normal     No     Modbus Unit ID and Station ID Enumerator
auxiliary/scanner/scada/modbusclient                                   normal     No     Modbus Client Utility
auxiliary/scanner/scada/modbusdetect                  2011-11-01       normal     Yes    Modbus Version Scanner
auxiliary/scanner/scada/moxa_discover                                  normal     Yes    Moxa UDP Device Discovery
auxiliary/scanner/scada/pcomclient                                     normal     No     Unitronics PCOM Client
auxiliary/scanner/scada/profinet_siemens                               normal     No     Siemens Profinet Scanner
auxiliary/scanner/scada/sielco_winlog_fileaccess                       normal     Yes    Sielco Sistemi Winlog Remote File Access
```

Ilustración 30 - Módulos auxiliares de Metasploit para el reconocimiento ICS

```
msf5 auxiliary(scanner/scada/modbusdetect) > run

[+] 172.17.0.2:502        - 172.17.0.2:502  - MODBUS - received correct MODBUS/TCP header (unit-ID: 1)
[*] 172.17.0.2:502        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Ilustración 31 - Detectando modbus en el host objetivo a través de Metasploit

Si el protocolo SNMP se encuentra activo también nos puede aportar información útil acerca del dispositivo en la red. Herramientas como *snmpwalk* o *snmpcheck* pueden ser útiles para recabar información relativa a este protocolo; lo que nos dará información del dispositivo en la red, como pueden ser puertos abiertos, sin necesidad de llevar a cabo un escáner completo sobre el componente objetivo.

4.2.2.2 Escáneres ICS

En este apartado se verán algunas de las herramientas creadas específicamente para la ejecución de pentesting contra ICSs y sus protocolos, señalando aquellas útiles en la fase de reconocimiento.

PLCScan[9]

Herramienta escrita en Python útil para escanear PLCs que se encuentren utilizando el protocolo Modbus TCP y S7Comm. PLCScan mostrará toda la información que pueda recabar a través de estos dos protocolos: Unit ID, Response Error, Device info y módulo, firmware, nombre del PLC, número de serie, tipo de módulo, etc., respectivamente. Se podrá escanear directamente tanto un dispositivo como una red completa. A continuación se muestra un ejemplo de uso y su salida.

[9] Github de PLCScan: https://github.com/meeas/plcscan

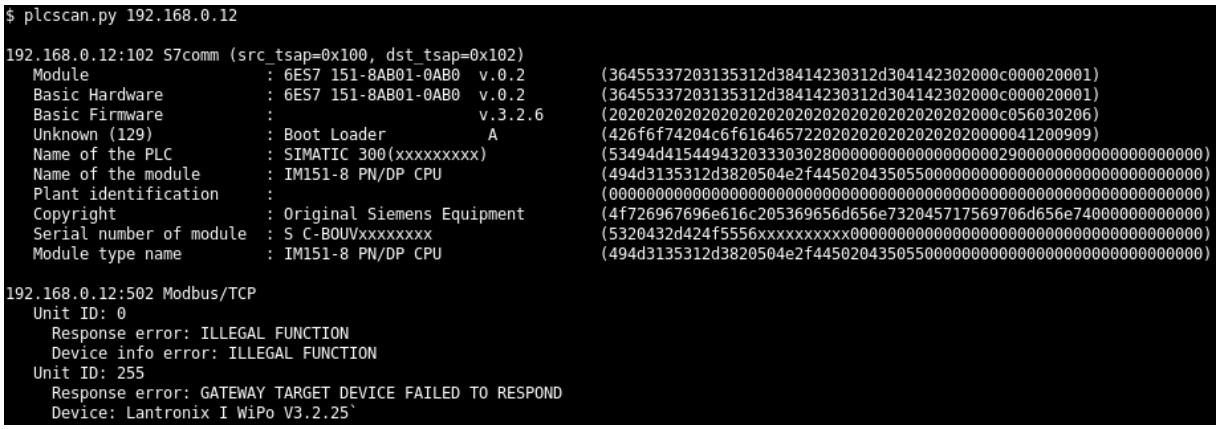


Ilustración 32 - Ejemplo de salida de PLCScan

Modscan[10]

Script en Python orientado a la detección de dispositivos Modbus TCP en la red. Cuando encuentra el puerto abierto relativo a Modbus trata de hallar el UID del esclavo a través de fuerza bruta. Es algo ruidoso e ineficiente ya que escanea todos los puertos y emplea fuerza bruta. También es posible escribir y leer valores en los registros con esta herramienta, pero eso corresponde a la fase de explotación.

SMOD[11]

Framework modular que consta de todo tipo de herramientas para el diagnóstico y realización de acciones ofensivas, útil a la hora de realizar pentests sobre el protocolo Modbus TCP. Su uso es similar al de Metasploit. Está programado en Python y utiliza el paquete Scapy, entre otros. Consta de módulos tanto para la recogida de información como para la fase de explotación. Cuenta con tres módulos de escaneo: uno para la detección del protocolo Modbus TCP, otro para conseguir mediante fuerza bruta el UID de dispositivos esclavos y el último para comprobar qué funciones están soportadas por el dispositivo:

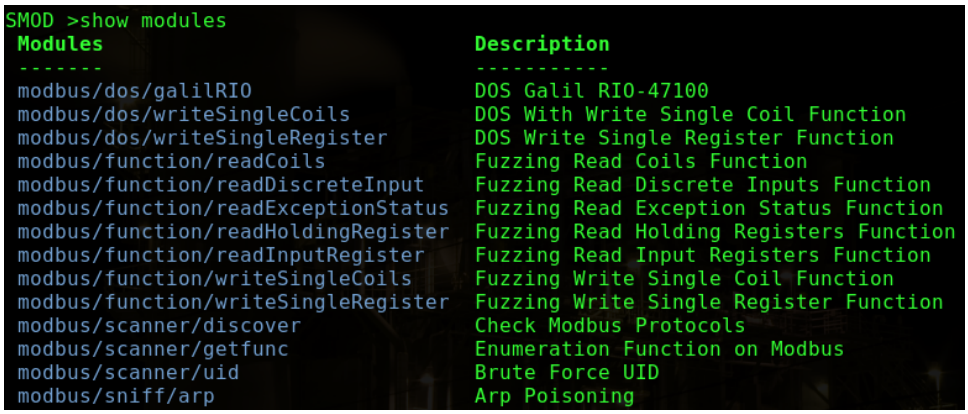


Ilustración 33 - Módulos de SMOD

[10] Github de Modscan: https://github.com/adarshdinesh/modscan
[11] Github de SMOD: https://github.com/Exploit-install/smod

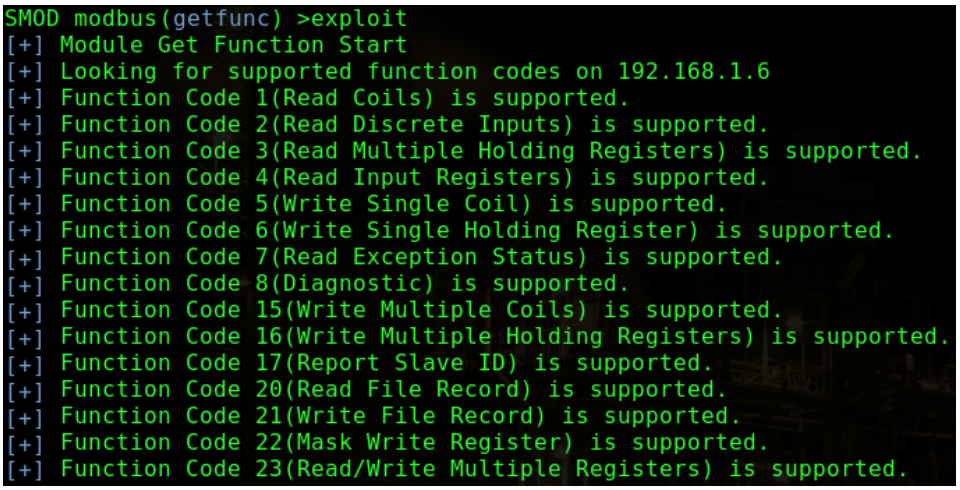


Ilustración 34 - Ejemplo de ejecución del módulo *getfunc* de SMOD

SCADAscan[12]

SCADAScan es un script escrito en Perl que permite encontrar esclavos SCADA en la red o dispositivo que se le proporcione a través de la línea de comandos. Actualmente admite la enumeración de Modbus y DNP 3.

ICSSPLOIT[13]

De apariencia y uso muy parecidos a Metasploit, ICSSPLOIT es un framework de explotación de Sistemas de Control Industrial escrito en Python. Dispone de módulos de explotación, credenciales y escáneres. En este apartado, son de interés los englobados en el último módulo. Soporta Modbus TCP, WdbRPC Version 2 (Vxworks); implementa módulos de escaneo para protocolos como Profinet DCP, Vxworks 6, S7Comm, Ethernet/IP y CIP.

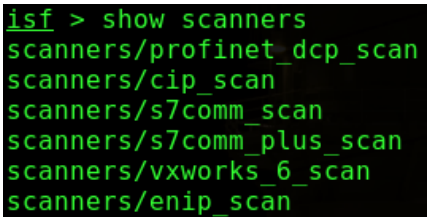


Ilustración 35 - Módulos de escaneo de ICSSPLOIT

[12] Github de SCADAScan: https://github.com/amolsarwate/scadascan
[13] Github de ICSSploit: https://github.com/dark-lbp/isf

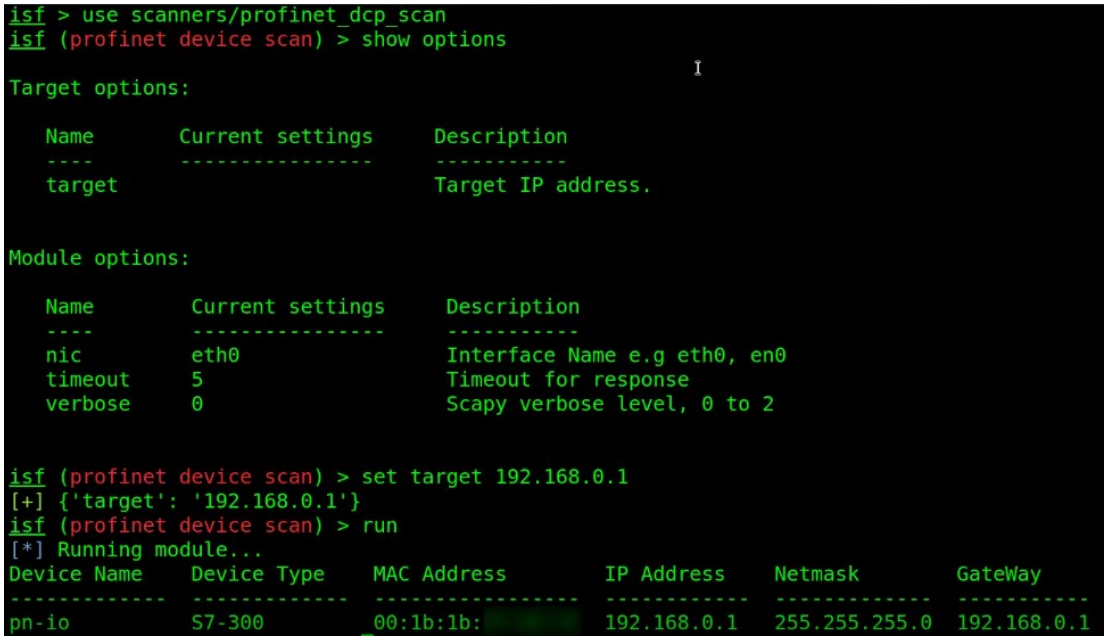


Ilustración 36 - Ejemplo de ejecución de escaneo Profinet DCP con ICSSPLOIT (Extraída de: [39])

Un punto importante a la hora acometer este tipo de escaneos, independientemente de las herramientas utilizadas, es tomar una serie de precauciones ya que no se sabe cómo puede reaccionar los componentes a escanear del ICS. Como destaca Iturbe [39], una regla de hora a la hora de ejecutar escáneres sobre este tipo de redes es no hacerlo con el sistema en funcionamiento. Un paquete malformado, por ejemplo, podría causar graves consecuencias físicas en el sistema y ser un peligro para la seguridad de las personas. Lo ideal sería simular en un laboratorio las pruebas a realizar sobre una copia exacta del entorno para observar y prever las consecuencias que tendrá en el mundo real, pero esto no siempre es posible, sobre todo si se trata de un pentest de caja negra, como es el caso de estudio. Tal y como señala Iturbe [39], los escaneos de redes industriales deben realizarse con máximo cuidado, realizándose la prueba de penetración durante un tiempo de inactividad planificado en la planta extremando así las precauciones de seguridad. Reducir los hilos de los escaneos y no utilizar scripts que no se conozcan a la perfección ayudará en esta tarea (en el caso de nmap, por ejemplo, utilizar la opción “--scan-delay=1” para no escanear más de un puerto a la vez y no utilizar la opción de scripts por defecto “-sC”).

4.2.3 Identificación de vulnerabilidades

Una vez llevada a cabo la fase de reconocimiento, ya se tendrá mucha información acerca de los componentes del Sistema de Control Industrial. Estarán identificados los tipos de componentes, sus direcciones IPs, protocolos, versiones de los mismos, versiones de firmware, etc. Con toda esta información, el pentester deberá identificar individualmente las vulnerabilidades de cada componente del ICS. Esto se puede llevar a cabo de dos maneras: manualmente, mediante búsquedas en bases de datos de vulnerabilidades o tratando de encontrar vulnerabilidades 0-day; o a través de herramientas automatizadas encargadas del reconocimiento de vulnerabilidades.

En el primero de los casos serán útiles bases de datos de vulnerabilidades o exploits como ExploitDB[14], el proyecto CVE[15] de Mitre, NVD[16] del NIST, etc. Se deberá aprovechar toda la información recolectada como marcas, versiones de dispositivos, tipos de protocolos, versiones de los mismos, etc. para enfocar las búsquedas e identificar así las posibles vulnerabilidades sobre los componentes del ICS. Por otro lado, si se dispone del código fuente de los servicios utilizados por los componentes, se podrá realizar un análisis del código para la búsqueda de vulnerabilidades de día 0 que aún no hayan sido reportadas; siendo esto mucho más costoso en tiempo.

En segundo lugar, los escáneres de vulnerabilidades son herramientas muy potentes que permitirán la identificación de vulnerabilidades en un tiempo mucho menor que en el caso anterior. Aun así, estas herramientas tienen sus inconvenientes ya que serán muy intrusivas y podrían causar un fuerte daño sobre el ICS al llevar a cabo la identificación de vulnerabilidades, ya que en muchos casos realizarán pruebas para la comprobación. Esto podría causar la denegación de servicio de los componentes y podría tener también consecuencias físicas en la planta. Por ello, lo óptimo sería el uso de este tipo de herramientas con el Sistema de Control Industrial parado y habiéndolo probado previamente contra un entorno similar en un laboratorio.

Un escáner de vulnerabilidades por antonomasia es Nessus[17] de la empresa Tenable. A la fecha de redacción de esta memoria, Nessus cuenta con un total de 319 plugins [41] para la detección de vulnerabilidades en Sistemas de Control Industrial.

Además, actualmente, la empresa Tenable ofrece una solución para Sistemas de Control Industrial llamada *Industrial Security*[18]. Dicha herramienta permite identificar las vulnerabilidades de un ICS pudiendo visualizarlas de manera amigable por criticidad, por niveles del ISA-95, por protocolos y por redes; aunque sí es verdad que está pensada para un uso más defensivo que ofensivo.

Al finalizar esta etapa del pentesting, el auditor poseerá una lista con los componentes del ICS y las posibles vulnerabilidades detectadas, ya sea manualmente o a través de herramientas automatizadas. Esta fase, tan importante como las demás, en muchos entornos industriales no será ni necesaria para comprometer el ICS ya que, como se ha redactado en capítulos anteriores, muchos de los protocolos utilizados en Sistemas de Control Industriales son vulnerables por definición. Todo ello se verá más detalladamente en la fase de explotación.

4.2.4 Explotación

La fase de explotación es una de las etapas de más criticidad dentro de una auditoría de seguridad. A partir de toda la información recogida y las vulnerabilidades identificadas se deberá explotar el sistema o sistemas a auditar. De esta manera, se deberán corroborar las vulnerabilidades identificadas en la fase anterior. Así, el

[14] Página oficial de ExploitDB: https://www.exploit-db.com/
[15] Página oficial de Mitre CVE: https://cve.mitre.org/
[16] Página oficial de NVD: https://nvd.nist.gov/vuln/search
[17] Página oficial de Nessus: https://www.tenable.com/products/nessus
[18] Página oficial de Industrial Security: https://www.tenable.com/products/industrial-security

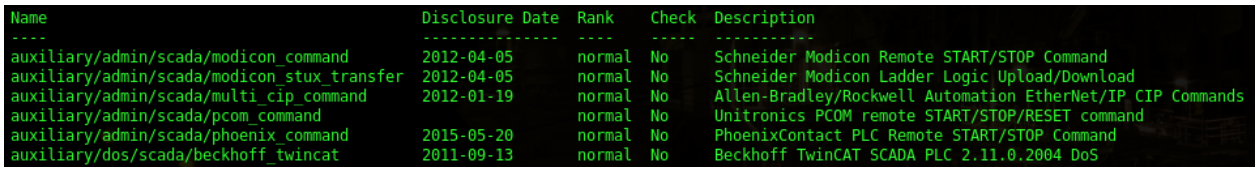
objetivo es similar al de una auditoría de seguridad sobre sistemas IT, pero con las peculiaridades de los sistemas OT. En esta fase el pentester deberá explotar tanto las vulnerabilidades de naturaleza OT como las vulnerabilidades IT que pueden existir también en entornos OT (vulnerabilidades en Sistemas Operativos, protocolos comunes en IT como http, ftp, rpc, etc.). En esta sección se tratarán sobre todo los primeros, ya que la explotación de vulnerabilidades propias de los entornos de las tecnologías de la información queda fuera del alcance de este proyecto.

4.2.4.1 Lanzamiento de exploits naturaleza OT

Una vez identificadas las potenciales vulnerabilidades en la fase anterior se deberán lanzar los exploits existentes. Para ello, al igual que en la fase de reconocimiento, se pueden utilizar varias herramientas:

Metasploit

Metasploit también cuenta con exploits dirigidos a ICS. La mayoría de ellos están dirigidos a componentes de la capa 2, como SCADAs o HMIs, los cuales están montados sobre un ordenador, generalmente, con un SO Windows. De ellos se hablará posteriormente, pero no en este apartado, con el que se pretende atacar a las vulnerabilidades detectadas de dispositivos de naturaleza OT. En este caso, se muestran algunos de los exploits encontrados que cumplan los requisitos de esta sección, relativos al control, descarga de la programación del PLC (*Ladder Logic*) y denegación de servicios de PLCs. La cuestión en una auditoría real sería buscar exploits o auxiliares en función de la información encontrada:



Name	Disclosure Date	Rank	Check	Description
auxiliary/admin/scada/modicon_command	2012-04-05	normal	No	Schneider Modicon Remote START/STOP Command
auxiliary/admin/scada/modicon_stux_transfer	2012-04-05	normal	No	Schneider Modicon Ladder Logic Upload/Download
auxiliary/admin/scada/multi_cip_command	2012-01-19	normal	No	Allen-Bradley/Rockwell Automation EtherNet/IP CIP Commands
auxiliary/admin/scada/pcom_command		normal	No	Unitronics PCOM remote START/STOP/RESET command
auxiliary/admin/scada/phoenix_command	2015-05-20	normal	No	PhoenixContact PLC Remote START/STOP Command
auxiliary/dos/scada/beckhoff_twincat	2011-09-13	normal	No	Beckhoff TwinCAT SCADA PLC 2.11.0.2004 DoS

Ilustración 37 - Exploits relativos a PLCs en Metasploit

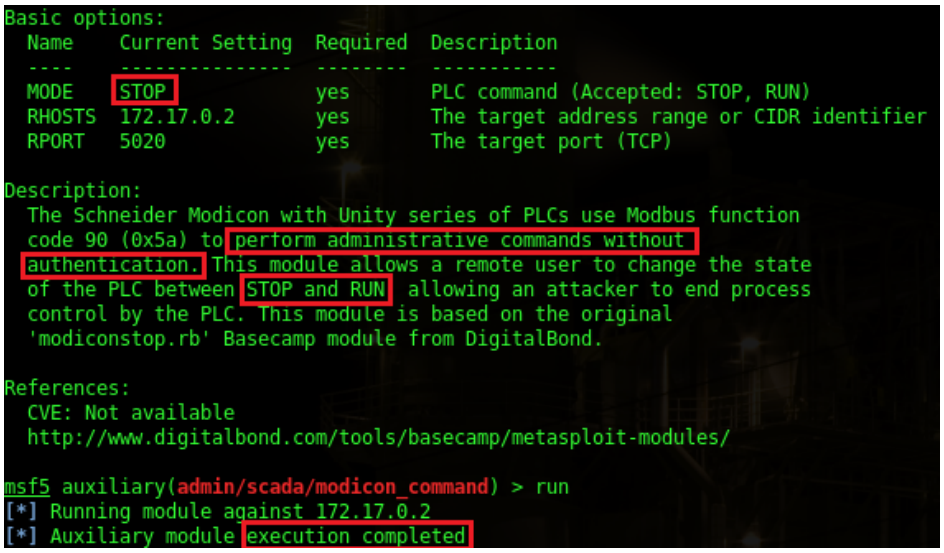


Ilustración 38 – Ejemplo de parada del PLC Schneider Modicon con Metasploit

SMOD

Tal y como se ha citado en la fase de reconocimiento, SMOD también dispone de módulos para la explotación del protocolo Modbus. Dispone de tres módulos de denegación de servicio. El resto de módulos, relativos a la escritura/lectura de esclavos del protocolo Modbus, se tratará en un apartado posterior. También cuenta con un módulo para la realización de un envenenamiento ARP, útil para realizar un ataque MitM. El uso de esta herramienta es muy similar a Metasploit, simplemente habrá que seleccionar el exploit, cargar las opciones y lanzarlo.

ICSSPLOIT

Al igual que el anterior, este framework de explotación de ICS cuenta con módulos para explotar dichos sistemas. En concreto, actualmente está dirigido a componentes PLC, dispositivos, como se ha visto, clave dentro de un ICS ya que permiten la interacción directa con el proceso de producción. ICSSPLOIT cuenta con siete exploits para atacar vulnerabilidades sobre estos componentes:

```
isf > show exploits
exploits/misc/fake_dhcp_server
exploits/plcs/siemens/s7_300_400_plc_control
exploits/plcs/siemens/s7_1200_plc_control
exploits/plcs/siemens/profinet_set_ip
exploits/plcs/qnx/crash_qnx_inetd_tcp_service
exploits/plcs/qnx/qconn_remote_exec
exploits/plcs/vxworks/vxworks_rpc_dos
exploits/plcs/schneider/quantum_140_plc_control
```

Ilustración 39 - Exploits disponibles en ICSSPLOIT

Entre ellos se encuentran exploits para el control de PLCs Siemens S7 300, 400 y 1200; control del PLC Schneider Quantum 140, DoS al SO VXworks usado por muchos PLCs, etc. El uso del framework y su estética es también similar al de Metasploit.

```
Module options:

   Name        Current settings     Description
   ----        ----------------     -----------
   slot        2                    CPU slot number.
   command     2                    Command 1:start plc, 2:stop plc.


isf (S7-300/400 PLC Control) > set target 192.168.0.1
[+] {'target': '192.168.0.1'}
isf (S7-300/400 PLC Control) > set slot 0
[+] {'slot': '0'}
isf (S7-300/400 PLC Control) > set command 2
[+] {'command': '2'}
isf (S7-300/400 PLC Control) > run
[*] Running module...
[+] Target is alive
[*] Sending packet to target
[*] Stop plc
```

Ilustración 40 - Exploit para la toma de control de un PLC Siemens S7 300/400 (Extraída de: [42])

4.2.4.2 Lectura/escritura de protocolos

El objetivo final de un atacante en un escenario real sería la interacción directa con el proceso de producción o elementos del nivel 0 del modelo ISA-95. Esto se puede conseguir mediante la lectura y escritura de valores en dispositivos como PLCs a través de los protocolos vistos en capítulos anteriores. Por tanto, el auditor de seguridad deberá de llevar a cabo también esta tarea.

Como se ha relatado a lo largo de la memoria, la mayoría de los protocolos utilizados carecen de las medidas de seguridad necesarias. Muchos de ellos permiten ser utilizados sin autenticación alguna (si no se aplican medidas de seguridad extra). Esto implica que cualquier dispositivo pueda actuar como nodo maestro enviando instrucciones o leyendo valores sobre los nodos esclavos. Así, por ejemplo, un atacante podría actuar como nodo maestro alterando valores de un PLC para afectar o tomar el control sobre los procesos físicos.

En esta sección se relatará la lectura y escritura mediante dos de los protocolos más usados: Modbus TCP y Siemens S7Comm. Cabe destacar que en ambos casos se estarán modificando valores del PLC y por tanto infiriendo en su funcionamiento. Por tanto se debe conocer qué se está haciendo y sobre qué procesos está actuando el PLC, no realizando estas pruebas, si se desconocen las implicaciones, nunca sobre sistemas en funcionamiento.

Modbus TCP

Como se ha descrito en capítulos anteriores, este protocolo permitirá la interacción con componentes del nivel 1, como PLCs, permitiendo, entre otras funciones, la lectura y escritura de los registros, que almacenan 16 bits de información, y de los coils, que almacenan 1 bit de información. Mediante estos valores, el PLC llevará a cabo diversas acciones finales, con lo que con la modificación de estos valores se estará actuando directamente sobre el proceso de producción, por lo que resultan altamente interesantes.

Existen numerosas herramientas para ejecutar este cometido que actúan como clientes Modbus: el módulo *auxiliary/scanner/scada/modbusclient* de Metasploit, la herramienta mbtget, módulos de SMOD, etc. El auditor deberá usar aquella que mejores resultados o capacidades le ofrezca. En este caso, por ejemplificar, se usará tanto el módulo de Metasploit como mbtget para la escritura y lectura de coils y registros de un PLC.

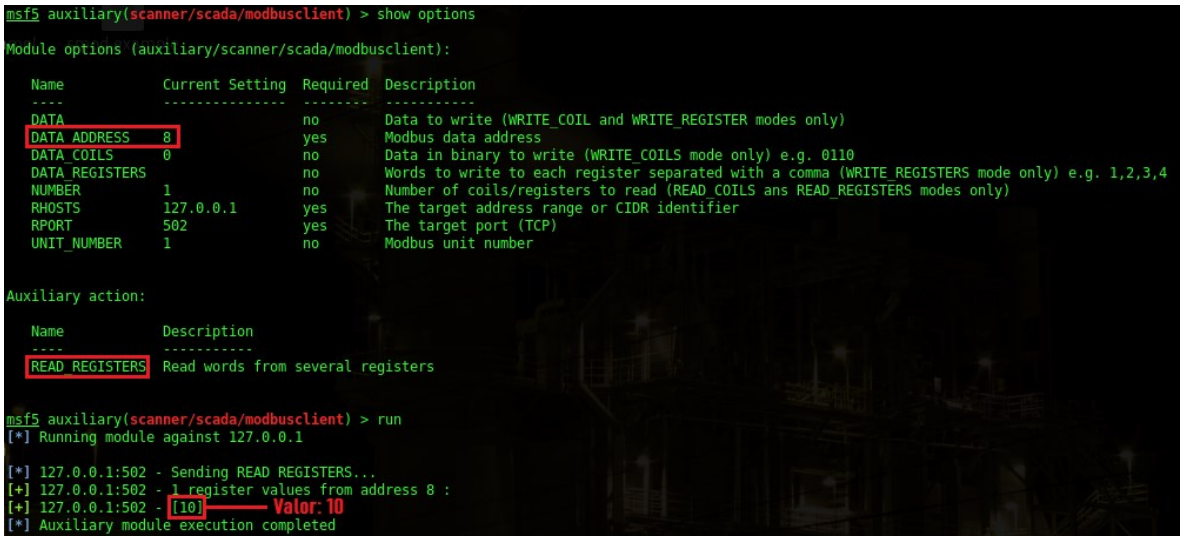


Ilustración 41 - Lectura de registro de esclavo a través de Modbus mediante Metasploit

Ilustración 42 - Escritura de registro de esclavo a través de Modbus mediante Metasploit

Mediante el mismo procedimiento anterior se comprueba que la escritura en el registro del PLC se ha llevado a cabo correctamente:

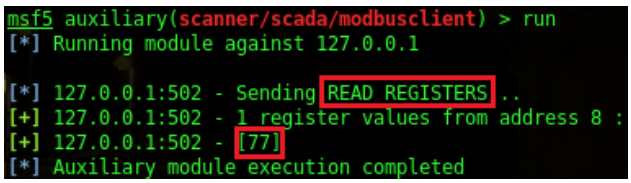


Ilustración 43 - Comprobación de escritura con éxito sobre el registro del PLC

A continuación se mostrará la lectura y escritura de valores en coils, pero esta vez mediante la herramienta mbtget.

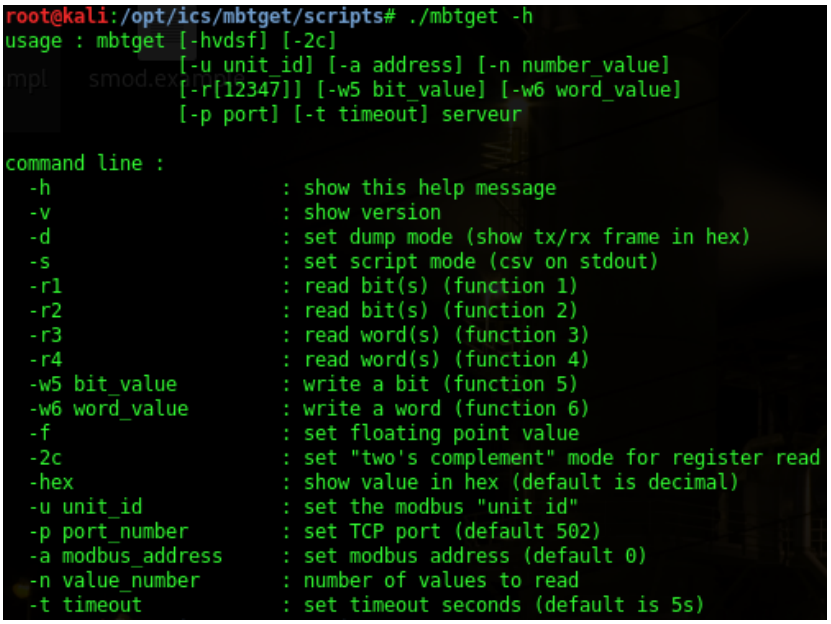


Ilustración 44 - Uso de mbtget

Ilustración 45 - Lectura de los 12 primeros coils mediante mbtget

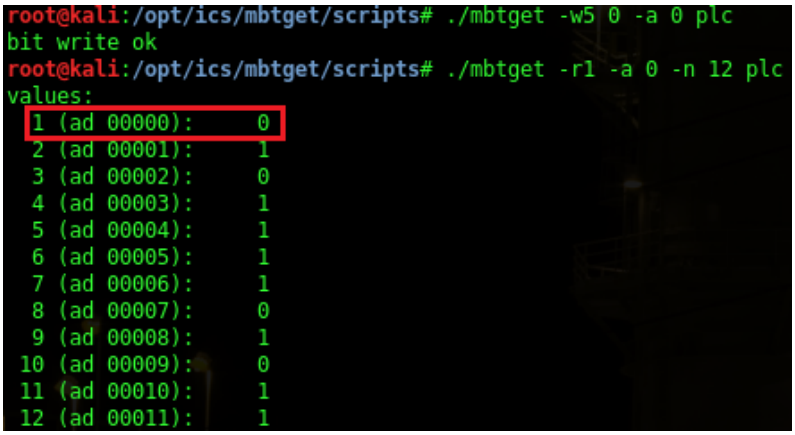


Ilustración 46 - Escritura del primer coil a 0 y comprobación mediante mbtget

Siemens S7Comm

S7Comm es un protocolo propietario de Siemens utilizado para la comunicación entre PLCs de la familia S7 de Siemens. Utiliza el puerto TCP/102 para las comunicaciones.

Existen tres versiones: S7Comm, para la comunicación con PLCs S7-200, S7-300 y S7-400; una versión de S7CommPlus “temprana”, para la comunicación entre PLCs S7-1200v3.0; y otra nueva para la comunicación de los PLCs S7-1200v4.0 y S7-1500. Este protocolo es utilizado para el intercambio de datos entre PLCs y para acceder a los datos del mismo desde un sistema SCADA o HMI. Está basado en COTP (Connection-Oriented Transport Protocol, RFC905). En este caso nos centraremos en el primero de ellos: S7Comm. Este protocolo carece de medidas de seguridad, siendo vulnerable a ataques de repetición, no posee mecanismos de autenticación, por lo que permitirá la lectura y escritura directa de valores en los PLCs.

Para la lectura y escritura de valores sobre dispositivos que utilicen este protocolo se utilizará la librería de código abierto Snap7[19]. En concreto, ofrece una demo para correr un cliente y un servidor que utilicen el protocolo S7Comm. Dicho cliente servirá para leer y escribir los valores de un PLC, que en este caso será el servidor levantado. Su uso es muy sencillo. En el cliente deberemos indicar la IP del PLC; se dispone de un botón para leer los valores y otro para la escritura una vez se hayan realizado las modificaciones.

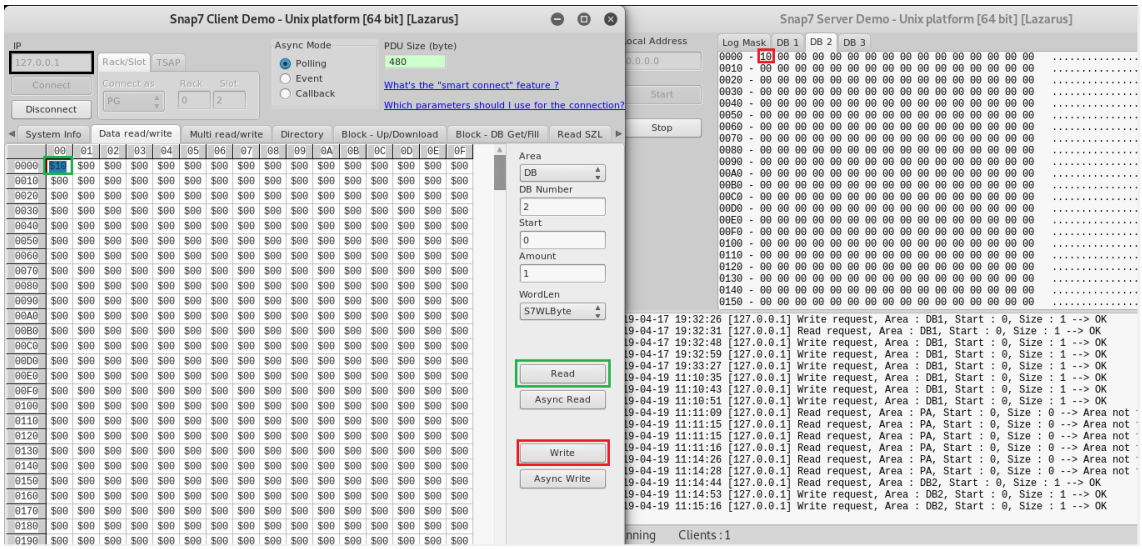


Ilustración 47 - Lectura y escritura PLC mediante S7Comm

4.2.4.3 Man in The Middle

Un ataque de Man in The Middle o Hombre en el Medio no es nada nuevo. Se trata de un tipo de ataque de red en el que el atacante logra situarse en el medio de las comunicaciones entre dos componentes de la red. Las implicaciones que este ataque tiene son desde el descubrimiento de secretos hasta la manipulación de la información o inyección de paquetes.

[19] Página oficial de Snap7: http://snap7.sourceforge.net/

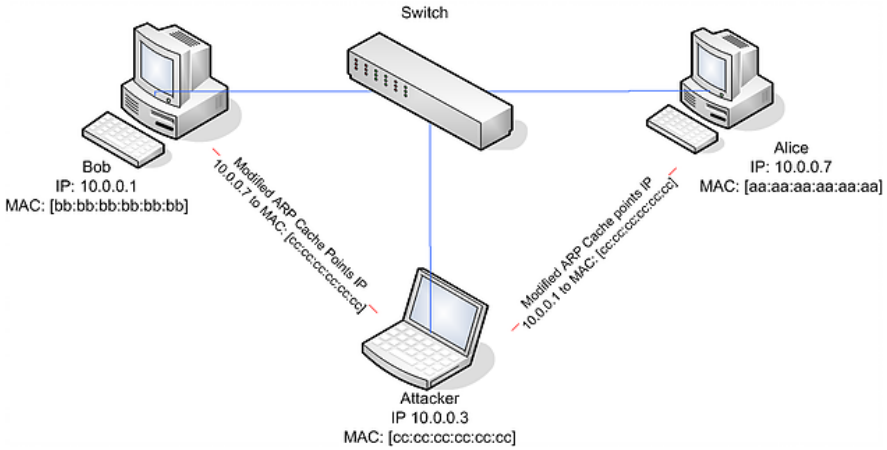


Ilustración 48 - Ataque de MitM (Extraída de: [43])

Teniendo en cuenta la falta de cifrado y la carencia de mecanismos de autenticación vista en la mayoría de los protocolos industriales, si un atacante logra situarse en el medio de las comunicaciones tendrá consecuencias devastadoras para el ICS: podría inspeccionar todo el tráfico para lograr un mejor entendimiento del sistema, o lo que es peor, podría modificar los valores de los paquetes o simplemente desecharlos, teniendo consecuencias críticas y directas sobre el proceso físico de producción.

Un caso de ejemplo sería la comunicación entre un PLC controlado a través de un panel de control en un HMI. Si un atacante logra con éxito el ataque podría manipular la información del estado del PLC enviada hacia el HMI provocando así que los ingenieros tomen decisiones erróneas sobre le ICS; o modificar las decisiones tomadas por los ingenieros a través del HMI sobre el PLC, pudiendo tener consecuencias críticas. En este caso de ejemplo, el atacante mediante técnicas de *ARP Poisoning* lograría hacer creer al PLC que su máquina es el HMI y al HMI que su máquina es el PLC. Una vez el atacante reciba los paquetes puede modificarlos según las necesidades del ataque. Tras esto, debe redirigirlos hacia su destino original, ya sea el PLC o el HMI. Esto es posible gracias a la falta de cifrado y autenticación en el protocolo.

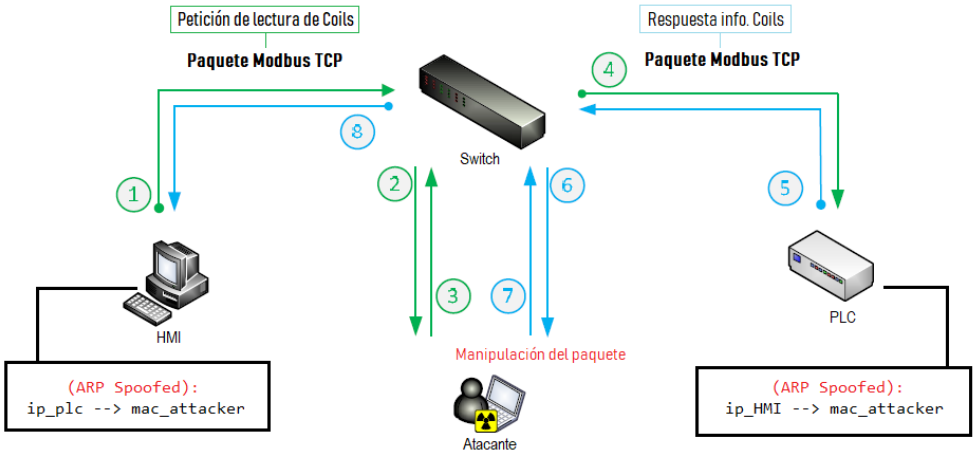


Ilustración 49 – Ejemplo de MitM en entornos ICS entre HMI-PLC

Una vez secuestrado el tráfico, la disección del contenido de los paquetes y la modificación de los mismos se puede realizar mediante el uso de librerías como

Scapy[20] (Python). Primero, el atacante deberá diseccionar el contenido de los paquetes. Si corresponden al protocolo que se desea atacar, por ejemplo Modbus TCP o DNP 3.0, deberá realizar las modificaciones pertinentes en sus parámetros. Por supuesto, esto requerirá el conocimiento del protocolo que se está atacando. Después deberá redirigir el paquete a su destino inicial.

Este tipo de ataque requiere también un conocimiento amplio de la topología de red que se está atacando y el flujo del tráfico, ya que en ocasiones los escenarios pueden ser complejos si existe una buena segmentación de red. Por ejemplo, puede que los componentes a los que se está atacando, siguiendo el caso de ejemplo: el PLC y el HMI, no estén situados en la misma red, por lo que en ese caso habría que atacar al componente que está en el mismo segmento de red que el atacante y a la puerta de enlace que comunica con el otro componente del ICS.

4.2.4.4 Explotación de vulnerabilidades de naturaleza IT

Como se ha ido relatando a lo largo de la memoria, existe una integración entre la parte IT y OT en un ICS. Por este motivo, además de los protocolos OT, existirán protocolos y funcionalidades de naturaleza IT en la parte OT de un Sistema de Control Industrial. La explicación exhaustiva de la explotación de este tipo de vulnerabilidades queda fuera del alcance de este proyecto, pero no por ello hay que obviarlas en la fase de explotación durante un pentesting a un ICS.

Vulnerabilidades en SOs

Microsoft Windows es uno de los Sistemas Operativos más utilizados en entornos OT. Esto no quiere decir que no se utilicen sistemas operativos como Unix o VxWorks, entre otros. Estos sistemas, situados en el nivel 2 del modelo jerárquico ISA-95, son utilizados en estaciones de trabajo de ingenieros y en ellos se encuentran programas o servicios que conforman un sistema SCADA o HMI que permiten tanto la monitorización como el control de componentes del nivel 1, ya sean PLCs, RTUs, IEDs, etc.

Muchos de estos sistemas se encuentran desactualizados, debido principalmente al concepto de “Safety” explicado anteriormente en esta memoria: un parche de seguridad puede tener consecuencias en el sistema impredecibles para el ICS causando desde la parada del proceso de producción y el coste económico que esto conlleva hasta implicaciones físicas. Por tanto, como indica Bolívar [19], muchas veces prevalece el concepto de “si funciona algo no lo toques”, lo que conlleva a sistemas desactualizados, agujeros de seguridad y posibilita la vulneración del sistema por parte de un ciberatacante. Por tanto, a partir de la versión del sistema operativo detectado en anteriores fases de reconocimiento, habrá que buscar exploits que exploten las posibles vulnerabilidades que puedan existir.

También habrá que tener en cuenta las aplicaciones utilizadas en estas estaciones de trabajo y sus posibles vulnerabilidades.

[20] Página oficial de Scapy: https://scapy.net/

```
exploit/windows/browser/keyhelp_launchtripane_exec                2012-06-26  excellent  No   KeyHelp ActiveX LaunchTriPane Remote Code Execution Vulnerability
exploit/windows/browser/teechart_pro                              2011-08-11  normal     No   TeeChart Professional ActiveX Control Trusted Integer Dereference
exploit/windows/browser/wellintech_kingscada_kxclientdownload     2014-01-14  good       No   KingScada kxClientDownload.ocx ActiveX Remote Code Execution
exploit/windows/fileformat/bacnet_csv                             2010-09-16  good       No   BACnet OPC Client Buffer Overflow
exploit/windows/fileformat/scadaphone_zip                         2011-09-12  good       No   ScadaTEC ScadaPhone Stack Buffer Overflow
exploit/windows/scada/abb_wserver_exec                            2013-04-05  excellent  Yes  ABB MicroSCADA wserver.exe Remote Code Execution
exploit/windows/scada/advantech_webaccess_dashboard_file_upload   2016-02-05  excellent  Yes  Advantech WebAccess Dashboard Viewer uploadImageCommon Arbitrary File Upload
exploit/windows/scada/advantech_webaccess_webvrpcs_bof            2017-11-02  good       No   Advantech WebAccess Webvrpcs Service Opcode 80061 Stack Buffer Overflow
exploit/windows/scada/citect_scada_odbc                           2008-06-11  normal     No   CitectSCADA/CitectFacilities ODBC Buffer Overflow
exploit/windows/scada/codesys_gateway_server_traversal            2013-02-02  excellent  No   SCADA 3S CoDeSys Gateway Server Directory Traversal
exploit/windows/scada/codesys_web_server                          2011-12-02  normal     Yes  SCADA 3S CoDeSys CmpWebServer Stack Buffer Overflow
exploit/windows/scada/daq_factory_bof                             2011-09-13  good       No   DaqFactory HMI NETB Request Overflow
exploit/windows/scada/delta_ia_commgr_bof                         2018-07-02  normal     No   Delta Electronics Delta Industrial Automation COMMGR 1.08 Stack Buffer Overflow
exploit/windows/scada/factorylink_csservice                       2011-03-25  normal     No   Siemens FactoryLink 8 CSService Logging Path Param Buffer Overflow
exploit/windows/scada/factorylink_vrn_09                          2011-03-21  average    No   Siemens FactoryLink vrn.exe Opcode 9 Buffer Overflow
exploit/windows/scada/ge_proficy_cimplicity_gefebt                2014-01-23  excellent  Yes  GE Proficy CIMPLICITY gefebt.exe Remote Code Execution
exploit/windows/scada/iconics_genbroker                           2011-03-21  good       No   Iconics GENESIS32 Integer Overflow Version 9.21.201.01
exploit/windows/scada/iconics_webhmi_setactivexguid               2011-05-05  good       No   ICONICS WebHMI ActiveX Buffer Overflow
exploit/windows/scada/igss9_igssdataserver_listall                2011-03-24  good       No   7-Technologies IGSS IGSSdataServer.exe Stack Buffer Overflow
exploit/windows/scada/igss9_igssdataserver_rename                 2011-03-24  normal     No   7-Technologies IGSS 9 IGSSdataServer .RMS Rename Buffer Overflow
exploit/windows/scada/igss9_misc                                  2011-03-24  excellent  No   7-Technologies IGSS 9 Data Server/Collector Packet Handling Vulnerabilities
exploit/windows/scada/igss_exec_17                                2011-03-21  excellent  No   Interactive Graphical SCADA System Remote Command Injection
exploit/windows/scada/indusoft_webstudio_exec                     2011-11-04  excellent  Yes  InduSoft Web Studio Arbitrary Upload Remote Code Execution
exploit/windows/scada/moxa_mdmtool                                2010-10-20  great      No   MOXA Device Manager Tool 2.1 Buffer Overflow
exploit/windows/scada/procyon_core_server                         2011-09-08  normal     Yes  Procyon Core Server HMI Coreservice.exe Stack Buffer Overflow
exploit/windows/scada/realwin                                     2008-09-26  great      No   DATAC RealWin SCADA Server Buffer Overflow
exploit/windows/scada/realwin_on_fc_binfile_a                     2011-03-21  great      No   DATAC RealWin SCADA Server 2 On_FC_CONNECT_FCS_a_FILE Buffer Overflow
exploit/windows/scada/realwin_on_fcs_login                        2011-03-21  great      No   RealWin SCADA Server DATAC Login Buffer Overflow
exploit/windows/scada/realwin_scpc_initialize                     2010-10-15  great      No   DATAC RealWin SCADA Server SCPC_INITIALIZE Buffer Overflow
exploit/windows/scada/realwin_scpc_initialize_rf                  2010-10-15  great      No   DATAC RealWin SCADA Server SCPC_INITIALIZE_RF Buffer Overflow
exploit/windows/scada/realwin_scpc_txtevent                       2010-11-18  great      No   DATAC RealWin SCADA Server SCPC_TXTEVENT Buffer Overflow
exploit/windows/scada/scadapro_cmdexe                             2011-09-16  excellent  No   Measuresoft ScadaPro Remote Command Execution
exploit/windows/scada/sunway_force_control_netdbsrv               2011-09-22  great      No   Sunway Forcecontrol SNMP NetDBServer.exe Opcode 0x57
exploit/windows/scada/winlog_runtime                              2011-01-13  great      No   Sielco Sistemi Winlog Buffer Overflow
exploit/windows/scada/winlog_runtime_2                            2012-06-04  normal     No   Sielco Sistemi Winlog Buffer Overflow 2.07.14 - 2.07.16
exploit/windows/scada/yokogawa_bkbcopyd_bof                       2014-03-10  normal     Yes  Yokogawa CENTUM CS 3000 BKBCopyD.exe Buffer Overflow
exploit/windows/scada/yokogawa_bkesimmgr_bof                      2014-03-10  normal     Yes  Yokogawa CS3000 BKESimmgr.exe Buffer Overflow
exploit/windows/scada/yokogawa_bkfsim_vhfd                        2014-05-23  normal     No   Yokogawa CS3000 BKFSim_vhfd.exe Buffer Overflow
exploit/windows/scada/yokogawa_bkhodeq_bof                        2014-03-10  average    Yes  Yokogawa CENTUM CS 3000 BKHOdeq.exe Buffer Overflow
```

Ilustración 50 - Exploits en Windows relativos a aplicaciones SCADA

Vulnerabilidades en protocolos IT

En muchos de los componentes de un ICS de la red OT existen tanto servidores web como FTP (u otro tipo de servicio IT), ya sea para el control de los dispositivos o para el almacenamiento de información. Un ejemplo de ello sería una interfaz web de un componente SCADA donde se pueden consultar datos de los procesos de la planta, la interfaz web desde controlar un PLC, el panel de control web de un HMI para el control de procesos, o la existencia de un servidor FTP en un PLC para el almacenamiento y descarga de datos.

Por este motivo habrá también que someter a examen a estas aplicaciones, ya sean web, ftp o de otra naturaleza.

Para las aplicaciones web habrá que realizar un testeo exhaustivo de las mismas, ya que no están exentas de vulnerabilidades comunes como XSS, SQLi, LFI, RFI, SSRF, XXE, RCE, etc., las cuales, en estos entornos, tendrán consecuencias mayores que en un escenario IT. No es lo mismo obtener un ejecución remota de código en un servidor de una red empresarial que en un sistema que soporta un HMI encargado del control de los procesos físicos de una infraestructura crítica.

Ilustración 51 - LFI en la interfaz web de un PLC Schneider (Extraída de: [44])

Respecto al resto de servicios IT identificados, y este último, habrá que buscar exploits relativos a las versiones de los mismos ya que puede que no estén parcheados. Esto

permitirá obtener más información o incluso el control de los sistemas que levantan dichos servicios.

Si es posible también será útil revisar el código fuente de dichos servicios, encontrando en muchas ocasiones información sensible como contraseñas de acceso a diferentes servicios.

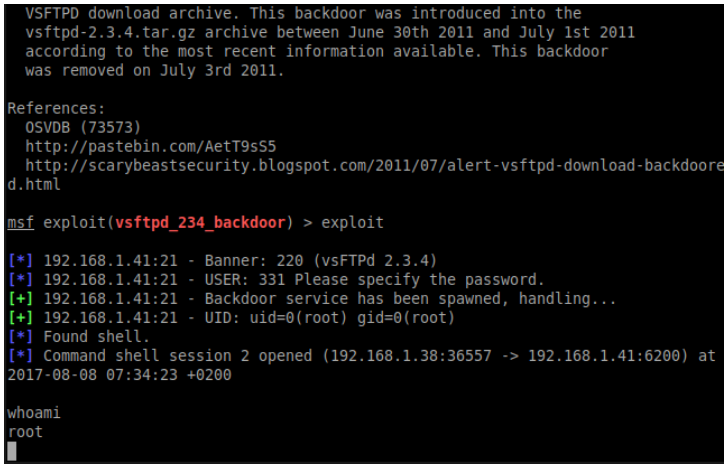


Ilustración 52 - Explotación de backdoor en servicio FTP

Otro punto a tratar en este apartado es el uso de contraseñas débiles, filtradas en brechas de seguridad o contraseñas por defecto, que no han sido cambiadas aplicando un procedimiento de seguridad adecuado. Por ello, también habrá que tener en cuenta los ataques a sistemas de autenticación, si es que los hay, sobre todos aquellos basados en el uso de diccionarios prestablecidos con el tipo de contraseñas citadas anteriormente.

La fase de explotación, al igual que las otras fases desarrolladas anteriormente, debe de acometerse con extremada precaución, en un momento en el que el Sistema de Control Industrial no esté en funcionamiento. Si es posible, se debe generar una copia del entorno en un laboratorio para lanzar previamente las pruebas, aunque esto no es siempre del todo viable, sobre todo en una auditoría de caja negra, como es el caso que acontece. Un exploit, o simplemente el envío de un paquete malformado, pueden generar desde la denegación de servicio de un componente de la planta hasta repercusiones físicas en los procesos de producción como fallo o rotura de componentes que pueden poner en riesgo incluso la seguridad física de las personas.

Como se ha podido observar, durante la fase de explotación el auditor se debe aprovechar principalmente de la carencia de medidas de seguridad en los protocolos OT: falta de autenticación y falta de cifrado. Todo ello permitirá realizar tanto la escritura y lectura en componentes del ICS del nivel 1 de la jerarquía ISA-95 que utilicen dichos protocolos, como la realización de un ataque de MitM para el espionaje y modificación de las comunicaciones; lo que tendrá consecuencias directas sobre los procesos físicos de producción. Además, se ha repasado el uso de exploits, tanto los dirigidos a componentes del ICS como PLCs, como los dirigidos al sistema operativo de componentes del nivel 2 y las aplicaciones levantadas en dichos sistemas. Tampoco se deben obviar los servicios de naturaleza IT disponibles en la parte OT del ICS y las posibles vulnerabilidades. Por ello, se destacan servicios como FTP o HTTP(S), poniendo especial empeño en la auditoría exhaustiva de este último.

4.2.5 Post-explotación

En el caso de estudio, con la explotación ejecutada en la fase anterior ya se habría finalizado la auditoría de seguridad: cabe recordad que el objetivo era interactuar directamente con los procesos de producción, cosa que ya se ha logrado con la anterior fase. La Post-explotación tendría sentido antes de llegar al punto desde el que parte la metodología propuesta: por ejemplo, justo al comprometer la estación de un ingeniero en la red OT. En el caso de estudio se supone que ya se tiene acceso de manera persistente a un dispositivo comprometido de la red OT. Así, no tiene sentido establecer persistencia sobre un PLC, por ejemplo; la explotación de este tipo de componentes del ICS supondría el fin de la auditoría de seguridad de forma satisfactoria.

5. Conclusiones

Todo proyecto debe tener un apartado en el que se contemplen las conclusiones del mismo. En este apartado se realizará una valoración de cumplimiento de los objetivos y de la planificación temporal marcados al inicio del proyecto, líneas de mejora, etc. Por otro lado, se finalizará el capítulo con las conclusiones personales obtenidas tras la realización de este proyecto.

5.1 Valoración de cumplimiento

El objetivo principal fijado al principio de este proyecto fue estudiar y desarrollar un enfoque de pentesting para Sistemas de Control Industrial (ICS), el cual se ha cumplido de forma satisfactoria. Mediante este estudio, se ha conseguido establecer una metodología base que sirva para enfrentar futuras auditorías de seguridad contra este tipo de sistemas. La consecución de este objetivo principal implicaba el logro de otros objetivos más específicos, los cuales han sido todos también cumplidos de forma óptima.

- ✓ **Estudiar y desarrollar un enfoque de pentesting para Sistemas de Control Industrial (ICS).**
    - ✓ Poner en marcha en una máquina virtual Conpot.
    - ✓ Definir un ejemplo con Conpot.
    - ✓ Clasificar los elementos que pueden ser simulados de un ICS en el honyepot.
    - ✓ Definir los datos que pueden ser recogidos en un honeypot.
    - ✓ Realizar pruebas sobre el entorno.
    - ✓ Realizar un enfoque de pentesting para Sistemas de Control Industrial.

Respecto a la planificación temporal estimada al inicio del proyecto, no ha habido ningún problema con las entregas que se han tenido que realizar a lo largo del mismo. Si bien algunas tareas han llevado algo más de tiempo y otras menos, estas variaciones han sido equilibradas y han permitido mantener la planificación temporal estimada de forma óptima y fluida.

5.2 Conclusiones técnicas

Con este Trabajo de Fin de Máster se ha realizado un estudio y desarrollo de un enfoque de pentesting para Sistemas de Control Industrial (ICS).

Para ello, se ha definido el concepto de Sistema de Control Industrial, haciendo un recorrido sobre sus componentes, protocolos más comunes y las vulnerabilidades que en estos subyacen. Además, también se ha detallado cómo es, o debe ser, una arquitectura de red en un Sistema de Control Industrial, destacando el concepto de segmentación de red en relación a los niveles establecidos en el modelo de la

jerarquía ISA-95. Para terminar con el concepto de ICS también se han listado los activos más importantes dentro de este tipo de sistemas y las vulnerabilidades que se suelen dar con más asiduidad, algo que será clave a la hora de cómo enfocar el pentest.

Siguiendo con el orden de redacción de la memoria, se ha presentado el honeypot de baja interacción Conpot, que simula, entre otros, un PLC de un Sistema de Control Industrial, el cual servirá para realizar una prueba de concepto aplicando la metodología o enfoque de pentesting sobre un ICS. A su vez, se han redactado qué datos se pueden simular y recolectar y se ha definido un ejemplo con Conpot.

Después de la definición de los aspectos más técnicos sobre un ICS, ya se ha podido proseguir con los conceptos relacionados con la ciberseguridad respecto a los Sistemas de Control Industrial. Para ello, primero se ha presentado cómo es un pentest dirigido sobre sistemas IT, del cual emanará el enfoque definido; y el concepto y fases de la *ICS Cyber Kill Chain*, la cual permite analizar cómo actúan los atacantes frente a Sistemas de Control Industrial, algo que será también clave para saber cómo desempeñar un auditoría de seguridad contra este tipo de sistemas, ya sea un pentest o un ejercicio de Red Teaming más extenso en el tiempo. Una vez definidos todos estos aspectos, se ha procedido a desarrollar el enfoque de pentesting para Sistemas de Control Industrial. Se han presentado las diferentes posibilidades respecto a en qué nivel de la red iniciar las auditorías de seguridad, resumiendo y ejemplificando los posibles caminos hasta llegar a la red OT, punto de partida del enfoque de pentesting definido. A partir de aquí, se han presentado diferentes fases a seguir en la metodología – Reconocimiento, Identificación de vulnerabilidades y Explotación – exponiendo para cada una de ellas un abanico de herramientas a utilizar y las posibles estrategias a seguir para el correcto desempeño de cada fase. Un punto realmente clave ha sido la carencia de seguridad en los protocolos OT: sin cifrado, falta de autenticación, etc., algo que puede ser abusado en la fase de explotación para lanzar comandos directamente sobre los componentes, como PLCs, e interactuar con el proceso físico de producción, interceptar las comunicaciones, modificarlas, etc. Para finalizar con este punto, recordar que es preferible realizar el pentest en un punto en el tiempo en el que el Sistema de Control Industrial se encuentre parado, ya que las consecuencias que puede tener son impredecibles y en ocasiones pueden generar desde la denegación de servicio, que supondrá grandes pérdidas económicas, hasta daños en los componentes que pueden generar, por ejemplo, explosiones, y que pueden poner en riesgo la vida de las personas.

Por último, a modo de anexo, se ha presentado una PoC (*Proof of Concept*), donde se aplica el enfoque de pentesting definido sobre el honeypot de baja interacción Conpot, que simula un componente, en concreto un PLC, de un ICS. En algunas ocasiones, sobre todo en los protocolos de naturaleza IT, Conpot ha presentado una baja interactividad, cosa que no ha permitido profundizar tanto en dichos protocolos. Si bien respecto a los protocolos OT, elemento principal que se pretendía auditar con esta PoC para poner en práctica lo descrito en la memoria, Conpot se ha comportado de manera relativamente correcta, hubiera sido interesante que el componente a auditar se hubiera comportado de manera más realista en su conjunto, dando más posibilidades al hacker ético para el desarrollo de la PoC. Así, una posible línea de mejora, o elemento que cambiaría del proyecto y que habría sido interesante, sería realizar la prueba de concepto contra un PLC real. Recalcar que se trata de detalle nimio que no influye en el núcleo y objetivo de la memoria, que es el desarrollo de un enfoque de pentesting sobre ICSs, pero que habría dotado a la puesta en práctica del mismo de una mayor diversidad.

5.3 Conclusión personal

El final de este Trabajo de Fin de Máster pone también fin a mi año cursando el Máster de Seguridad de las TIC en la UOC. Ha sido un año frenético. Ha sido un año dedicándome a la seguridad a tiempo completo, como quien diría, *24x7*: desde que me levantaba hasta que me acostaba; algo que he disfrutado muchísimo. He aprendido muchas cosas, tanto a nivel académico como a nivel extra-académico por mi propia cuenta. Todo ello se debe, sin dudad, a la pasión que tengo por este mundo y que me impulsa a ir más allá; la curiosidad innata en el ser humano.

Enfrentarme a un proyecto de este calado a nivel individual me ha aportado muchos conocimientos. Por un lado están los técnicos: cómo encarar un ethical hacking contra ICSs me ha llevado a conocer un ICS, sus componentes, protocolos, las posibles debilidades y como explotarlas de forma satisfactoria. Por otro lado están las lecciones aprendidas o los conocimientos que quedan fuera del espectro más técnico pero que son igual de importantes: hablo de la destreza para trabajar de forma individual, para investigar, para adquirir los conocimientos necesarios para lograr un objetivo final, etc. Además, enfrentarme a un proyecto relativo a seguridad ofensiva, o pentesting en concreto, sobre Sistemas de Control Industrial me ha permitido aumentar mi destreza como hacker ético para cuando tenga que encarar proyectos que estén relacionados con una tecnología que desconozco y de la que tendré que aprender para completarlo con éxito; algo clave en este campo.

Por último, agradecer a Tecnalia, en concreto a Xabier Larrucea Uriarte, por su colaboración, disposición y ayuda como co-director en este TFM, gracias. Agradecer también a mi tutor de la UOC, Juan Carlos Fernández Jara, por su ayuda y disposición, y a mi director del TFM por parte de la UOC, Carlos Hernández Gañán, por el rico feedback recibido. A mi familia y a mi novia por todo su apoyo incondicional y a los compañeros que he conocido que comparten mi pasión y hacen que esté aún más motivado, gracias.

4. Glosario

Ransomware: Software malicioso que al infectar un equipo le da al ciberdelincuente la capacidad de bloquear un dispositivo desde una ubicación remota y encriptar los archivos restringiendo el control y acceso de toda la información y datos almacenados. Lanza una ventana emergente en la que nos pide el pago de un rescate, dicho pago se hace generalmente en moneda virtual (bitcoins, por ejemplo).

TFM: Trabajo de Fin de Máster.

ECTS: Sistema Europeo de Transferencia y Acumulación de Créditos. istema utilizado por las universidades europeas para convalidar asignaturas y, dentro del denominado proceso de Bolonia, cuantificar el trabajo relativo al estudiante que trabaja bajo los grados auspiciados por el Espacio Europeo de Educación Superior (EEES).

Pivoting: Técnica que designa el movimiento lateral entre redes.

DMZ: Zona desmilitarizada. También conocida como red perimetral, se trata de una red que se ubica entre una red interna y una hostil.

Log: Fichero de bitácoras.

Broadcast: Forma de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo.

Firewall: Cortafuegos. Controla el acceso de una computadora a la red y de elementos de la red a la computadora.

Bypass: Palabra usada para referirse a una técnica de evasión de un sistema de seguridad.

Mapeo: Palabra utilizada, en el caso de la memoria, para definir la redirección de puertos o *port forwarding*. Acción de redirigir un puerto de red de un nodo de red a otro

OSINT: Inteligencia de Fuentes Abiertas. Inteligencia obtenida de fuentes disponibles públicamente.

Troyano: Malware que se presenta al usuario como un programa aparentemente legítimo e inofensivo, pero que, al ejecutarlo, le brinda a un atacante acceso remoto al equipo malware que se presenta al usuario como un programa aparentemente legítimo e inofensivo, pero que, al ejecutarlo, le brinda a un atacante acceso remoto al equipo infectado.

Powershell: Herramienta avanzada de configuración y control de un sistema basado en Windows. Se trata de una consola de sistema, un terminal o “CLI” bastante más avanzado y completo que MS-DOS o CMD desde el que se puede configurar completamente un equipo informático basado en Windows sin tener que depender de un escritorio para ello.

Spear phising: Técnica de phising dirigida a una persona, organización o empresa en particular especialmente personalizada, no de forma masiva.

VNC: Programa de software libre basado en una estructura cliente-servidor que permite observar las acciones del ordenador servidor remotamente a través de un ordenador cliente independientemente del sistema operativo.

RPD: Protocolo de Escritorio Remoto. Se trata de un protocolo propietario desarrollado por Microsoft que permite la comunicación en la ejecución de una aplicación entre una terminal y un servidor Windows.

Citrix: Solución que permite la virtualización de aplicaciones y escritorios.

Firmware: Programa informático que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo..

Coil (PLC): Referencia binaria de datos utilizada por el PLC para realizar operaciones. Son bits, contemplando los estados de encendido o apagado (1 o 0).

Registro (PLC): Referencia de tipo WORD (16 bits) utilizada por el PLC para realizar acciones (valores entre 0 y 65535).

Backdoor: Puerta trasera que permite el acceso ilícito a un sistema.

Bibliografía

[1] BOE, «Ley 8/2011 (Ley PIC),» 29 abril 2011. [En línea]. Available: https://www.boe.es/eli/es/l/2011/04/28/8/con.

[2] BBC, «Ukraine power cut 'was cyber-attack',» 11 enero 2017. [En línea]. Available: https://www.bbc.com/news/technology-38573074. [Último acceso: 22 febrero 2019].

[3] CONPOT, «CONPOT: ICS/SCADA Honeypot,» 4 febrero 2019. [En línea]. Available: http://conpot.org/. [Último acceso: 25 febrero 2019].

[4] UNED, «Estudiar en el Espacio Europeo de Educación Superior,» [En línea]. Available: http://portal.uned.es/portal/page?_pageid=355,3138322&_dad=portal. [Último acceso: 27 febrero 2019].

[5] CNPIC, «CNPIC - ¿Qué es una Infraestructura Crítica?,» [En línea]. Available: http://www.cnpic.es/Preguntas_Frecuentes/Que_es_una_Infraestructura_Critica/index.html. [Último acceso: 16 marzo 2019].

[6] ISA, «ISA-95,» [En línea]. Available: https://www.isa.org/isa95/. [Último acceso: 16 marzo 2019].

[7] M. Åkerman, «Implementing Shop Floor IT for Industry 4.0,» junio 2018. [En línea]. Available: https://www.researchgate.net/publication/326224890_Implementing_Shop_Floor_IT_for_Industry_40. [Último acceso: 16 marzo 2019].

[8] AutomationDirect, «PLC Communications | Coming of Age,» [En línea]. Available: https://library.automationdirect.com/plc-communications-coming-of-age/. [Último acceso: 18 marzo 2019].

[9] SCADATA, «RTU and PLC Protocols for SCADA Systems,» 16 septiembre 2016. [En línea]. Available: https://scadata.net/rtu-plc-protocols-scada-systems/. [Último acceso: 18 marzo 2019].

[10] whatis, «Intelligent electronic device (IED),» septiembre 2017. [En línea]. Available: https://whatis.techtarget.com/definition/intelligent-electronic-device. [Último acceso: 18 marzo 2019].

[11] plcdesign, «PLC o DCS – ¿Qué solución escoger?,» 6 agosto 2017. [En línea]. Available: http://plcdesign.xyz/plc-o-dcs/. [Último acceso: 18 marzo 2019].

[12] Rockwell Automation, «Qué esperar de un DCS,» [En línea]. Available: https://www.rockwellautomation.com/es_CEM/news/automation-today/detail.page?pagetitle=Qu%C3%A9-esperar-de-un-DCS&content_type=magazine&docid=c5ab55b5b15225e7ce4f6038902a5bd3. [Último acceso: 19 marzo 2019].

[13] T. Roybal, «Youtube: A Pentester's Intro to Attacking ICS/SCADA,» 1 junio 2017. [En línea]. Available: https://www.youtube.com/watch?v=LyzIrE6DpOM. [Último acceso: 19 marzo 2019].

[14] D. Miessler, «An ICS/SCADA Primer,» 4 febrero 2016. [En línea]. Available: https://danielmiessler.com/study/ics-scada/. [Último acceso: 19 marzo 2019].

[15] [En línea]. Available: https://aggregate.tibbo.com/images/aggregate/sh_hmi.png. [Último acceso: 19 marzo 2019].

[16] advenglobal, «Manufacturing execution systems (MES),» [En línea]. Available: http://adventglobal.com/mes.php. [Último acceso: 19 marzo 2019].

[17] hackers-arise, «SCADA Hacking: SCADA/ICS Protocols (Profinet/Profibus),» 17 junio 2017. [En línea]. Available: https://www.hackers-arise.com/single-post/2017/07/07/SCADA-Hacking-SCADAICS-Protocols-ProfinetProfibus. [Último acceso: 20 marzo 2019].

[18] hackers-arise, «SCADA Hacking: SCADA/ICS Communication Protocols (Modbus),» 5 enero 2017. [En línea]. Available: https://www.hackers-arise.com/single-post/2017/01/05/SCADA-Hacking-SCADAICS-Communication-Protocols-Modbus. [Último acceso: 20 marzo 2019].

[19] J. F. Bolívar, Infraestructuras críticas y sistemas industriales, Móstoles: 0xWORD, 2016.

[20] hacker-arise, «SCADA Hacking: SCADA Protocols (DNP3),» 10 febrero 2017. [En línea]. Available: https://www.hackers-arise.com/single-post/2017/02/10/SCADA-Hacking-SCADA-Prortocols-DNP3. [Último acceso: 21 marzo 2019].

[21] wikipedia, «DNP3,» 9 noviembre 2018. [En línea]. Available: https://es.wikipedia.org/wiki/DNP3. [Último acceso: 21 marzo 2019].

[22] Ixia Blog Team, «SCADA Distributed Network Protocol (DNP3),» 21 mayo 2015. [En línea]. Available: https://www.ixiacom.com/company/blog/scada-distributed-network-protocol-dnp3. [Último acceso: 21 marzo 2019].

[23] INCIBE, «Protocols and Network Security in ICS Infrastructures,» 15 mayo 2015. [En línea]. Available: https://www.incibe.es/extfrontinteco/img/File/intecocert/ManualesGuias/incibe_protocol_net_security_ics.pdf. [Último acceso: 21 marzo 2019].

[24] US Department of Homeland Security, «Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies,» septiembre 2016. [En línea]. Available: https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf. [Último acceso: 2019 marzo 22].

[25] Kaspersky Lab ICS CERT, «Threat Landscape for Industrial Automation Systems in H2 2017,» 26 marzo 2018. [En línea]. Available: https://securelist.com/threat-landscape-for-industrial-automation-systems-in-h2-2017/85053/. [Último acceso: 23 marzo 2019].

[26] NIST, «Guide to Industrial Control Systems (ICS) Security,» junio 2011. [En línea]. Available: https://www.ccn-cert.cni.es/publico/InfraestructurasCriticaspublico/Guide%20to%20Industrial%20Control%20.pdf. [Último acceso: 23 marzo 2019].

[27] INCIBE, «Honeypot, una herramienta para conocer al enemigo,» 14 junio 2018. [En línea]. Available: https://www.incibe-cert.es/blog/honeypot-herramienta-conocer-al-enemigo . [Último acceso: 27 marzo 2019].

[28] Wireshark, «S7 Communication (S7comm),» 13 mayo 2016. [En línea]. Available: https://wiki.wireshark.org/S7comm. [Último acceso: 27 marzo 2019].

[29] MITRE, «MITRE | ATT&CK,» 2018. [En línea]. Available: https://attack.mitre.org/. [Último acceso: 2 abril 2019].

[30] INCIBE, «Cyber Kill Chain en Sistemas de Control Industrial,» 27 octubre 2016. [En línea]. Available: https://www.incibe-cert.es/blog/cyber-kill-chain-sistemas-control-industrial. [Último acceso: 3 abril 2019].

[31] SANS, «The Industrial Control SystemCyber Kill Chain,» 5 octubre 2015. [En línea]. Available: https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297. [Último acceso: 3 abril 2019].

[32] Shodan, [En línea]. Available: https://www.shodan.io/. [Último acceso: 4 abril

2019].

[33] ZoomEye, [En línea]. Available: https://www.zoomeye.org/. [Último acceso: 4 abril 2019].

[34] Symantec, «Stuxnet - Modus Operandi,» Marzo 2011. [En línea]. Available: https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1493844778.pdf. [Último acceso: 4 abril 2019].

[35] Wikipedia, «Havex nuevo malware para sistemas ICS/SCADA,» 29 enero 2019. [En línea]. Available: https://en.wikipedia.org/wiki/Havex. [Último acceso: 4 abril 2019].

[36] jpalanco, 16 diciembre 2013. [En línea]. Available: https://github.com/jpalanco/nmap-scada. [Último acceso: 8 abril 2019].

[37] atimorin, 26 mayo 2014. [En línea]. Available: https://github.com/atimorin/scada-tools. [Último acceso: 8 abril 2019].

[38] digitalbond, 8 marzo 2016. [En línea]. Available: https://github.com/digitalbond/Redpoint. [Último acceso: 8 abril 2019].

[39] Enredandoconredes.com, «YouTube: Escaneo Profinet-DCP con ICSSPLOIT,» 1 octubre 2017. [En línea]. Available: https://www.youtube.com/watch?v=bPnzahuNzkQ. [Último acceso: 11 abril 2019].

[40] M. Iturbe, «Scanning industrial networks,» 7 octubre 2014. [En línea]. Available: https://iturbe.info/2014/10/scanning-industrial-networks/. [Último acceso: 8 abril 2019].

[41] Tenable, «SCADA Family for Nessus,» 2019. [En línea]. Available: https://www.tenable.com/plugins/nessus/families/SCADA. [Último acceso: 15 abril 2019].

[42] Enredandoconredes.com, «YouTube: STOP & RUN S7-300 with ICSSPLOIT,» 1 octubre 2017. [En línea]. Available: https://www.youtube.com/watch?v=O9LfEEpBvFA. [Último acceso: 16 abril 2019].

[43] hacker-arise, «Man-the-Middle (MiTM) Attack with ARPspoofing,» 25 julio 2017. [En línea]. Available: https://www.hackers-arise.com/single-post/2017/07/25/Man-the-Middle-MiTM-Attack-with-ARPspoofing. [Último acceso: 22 abril 2019].

[44] Ezequiel, «MÚLTIPLES VULNERABILIDADES (ICS) - SCHNEIDER ELECTRIC - 3,» 11 noviembre 2016. [En línea]. Available: http://misteralfa-hack.blogspot.com/2016/11/multiples-vulnerabilidades-en-plc_11.html. [Último acceso: 23 abril 2019].

Anexo 1: Pentest sobre Conpot

En el siguiente apéndice se aplicará la metodología descrita a lo largo del proyecto para someter a una auditoría de seguridad al honeypot Conpot, que simula un componente de un ICS, en concreto un PLC. Este anexo estará acompañado de tanto las evidencias en forma de capturas como breves explicaciones de las mismas.

Cabe recordar que se tendrá comunicación directa con el Conpot, es decir, el hacker ético y el componente del ICS estarán situados en la misma red. De esta manera, simulando un escenario real, se supondrá que el atacante tiene acceso a la red OT, ya sea por su compromiso directo o por la realización de movimientos laterales una vez comprometida la red corporativa o porque la auditoría comienza en este punto, tal y como se ha relatado en la memoria.

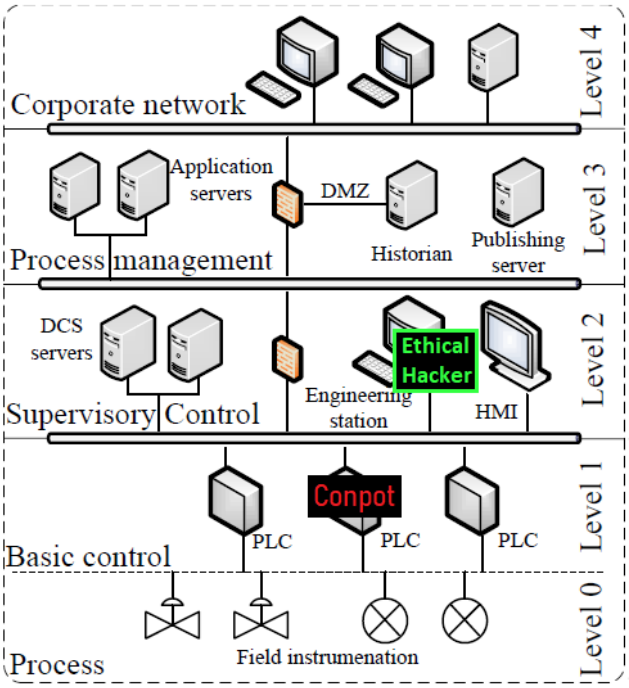


Figura 1 – Estado del hacker ético en la red en la auditoría simulada

1. Reconocimiento

En este caso particular se conoce la IP del objetivo, aun así se realizará un descubrimiento de hosts a través de nmap:

```
root@kali:~/conpot/recon# nmap -sP -o disc.nmap 172.17.0.0/16
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-03 16:34 CEST
Nmap scan report for conpot (172.17.0.2)
Host is up (0.000027s latency).
MAC Address: 02:42:AC:11:00:02 (Unknown)
Nmap scan report for 172.17.0.1
Host is up.
```

Figura 2 – Descubrimiento de objetivo

Una vez identificado el objetivo con la IP `172.17.0.2` se procede a realizar el escáner de puertos, tanto TCP como UDP, enumerando puerto, estado y servicio a través de nmap.

Recordatorio: Debido a un problema con las interfaces de red y la imposibilidad de cambiar los puertos de Conpot, se ha decido mapear los puertos de Conpot a los de la máquina local. Es decir, accediendo a un puerto de la máquina atacante se accederá directamente a otro de Conpot, mediante port forwarding, actuando como un puente entre máquinas. Así, herramientas que no permiten cambiar el puerto por defecto funcionarán, ya que, por ejemplo, PLCscan comprueba la información relativa al puerto 502 para Modbus TCP y no permite cambiar el puerto. En este caso, ya que la plantilla por defecto utiliza el puerto tcp/5020 para Modbus TCP, con el cambio introducido mapeando el puerto 5020 de Conpot al puerto 502 de la máquina local el problema queda solventado. Por tanto, a partir de este momento, se cambiará la dirección de Conpot por la local: 127.0.0.1; pero técnicamente se estará realizando la auditoría sobre el Conpot debido al mapeo de puertos. Todo esto se encuentra redactado y explicado en el capítulo 3, apartado 4 de la memoria: Definición de un ejemplo con Conpot.

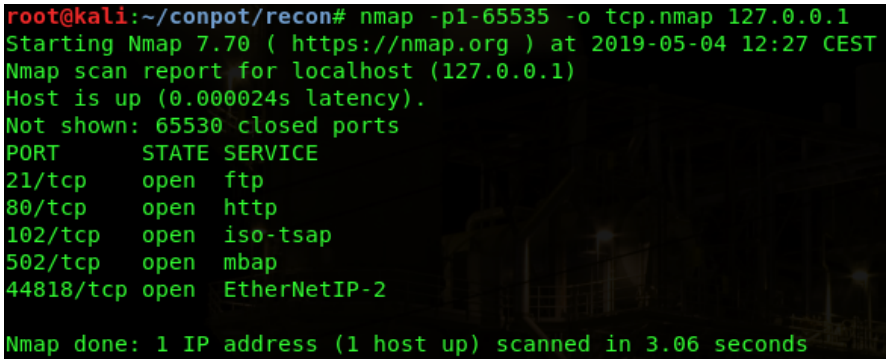


Figura 3 – Escáner de puertos TCP

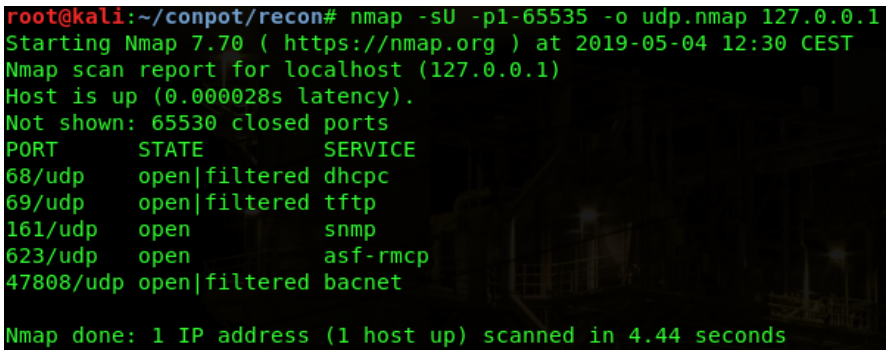


Figura 4 – Escáner de puertos UDP

La enumeración de versiones, a través de la opción `-sV` de nmap, provoca una excepción en Conpot debido al envío de paquetes que realiza nmap para saber la versión del servicio que se está ejecutando. Recuérdese que esto, que provoca la parada de Conpot, en un entorno real de producción podría provocar la parada del componente del ICS o tener consecuencias impredecibles.

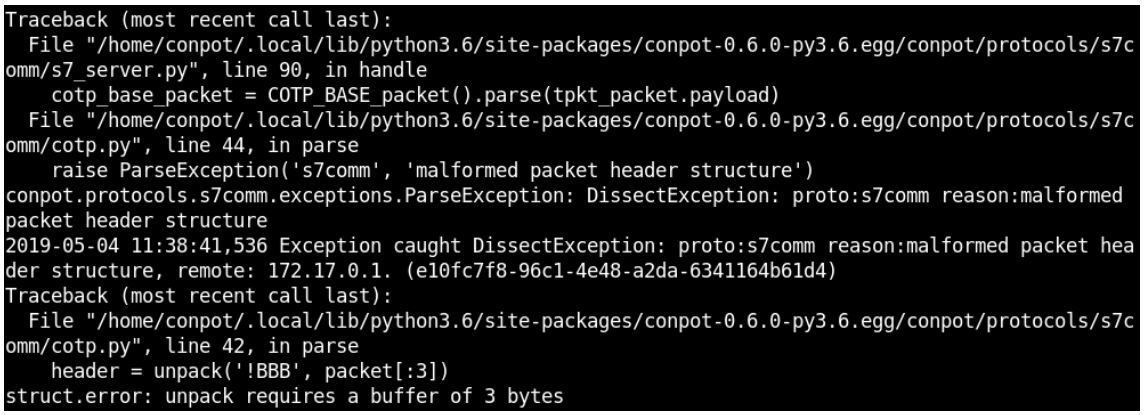


Figura 5 – Excepción y DoS a Conpot debido a opción -sV de nmap

Como paso siguiente, se realizará una verificación de los datos obtenidos y se intentará recabar más información de los protocolos en caso de que la verificación sea positiva. Para ello, se utilizarán tanto los escáneres exclusivamente diseñados para auditar componentes OT, tratados anteriormente en la memoria, como los scripts de nmap o herramientas como Metasploit.

Protocolo	Puerto	Naturaleza
HTTP	TCP/80	IT
S7comm	TCP/102	OT
Modbus TCP	TCP/502	OT
FTP	TCP/21	IT
IPMI	UDP/623	IT
TFTP	UDP/69	IT
SNMP	UDP/161	IT
Bacnet	UDP/47808	OT
EtherNetIP	TCP/44818	OT

Tabla 1 – Puertos y servicios activos

1.2 Protocolos OT

Tcp/102: Siemens S7 Comm

Se empezará con el puerto tcp/102. Dicho puerto, siguiendo los estándares por defecto, corresponde al protocolo de comunicaciones utilizado en los PLC Siemens S7: S7Comm. Nmap dispone de una serie de scripts para dicho protocolo:

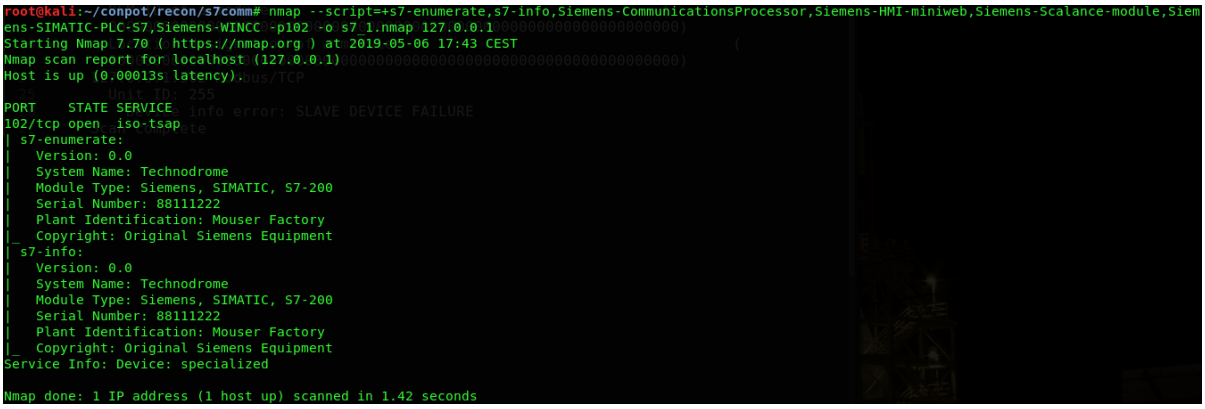


Figura 6 – Scripts de nmap sobre tcp/102

Como se puede ver en la figura anterior, en este caso no todos los scripts devuelven información. La información relevante recolectada es la siguiente:

System name: Technodrome
Module Type: Siemens, SIMATIC, S7-200
Serial Number: 88111222
Plant Identification: Mouser Factory
Copyright: Original Siemens Equipment

Se trata de un PLC de Siemens S7-200, cuyo serial es 88111222, su nombre es Technodrome y el identificador de la planta es Mouser Factory.

La herramienta PLCScan también servirá para recolectar información acerca de este protocolo (además del protocolo Modbus TCP, que se verá a posteriori):

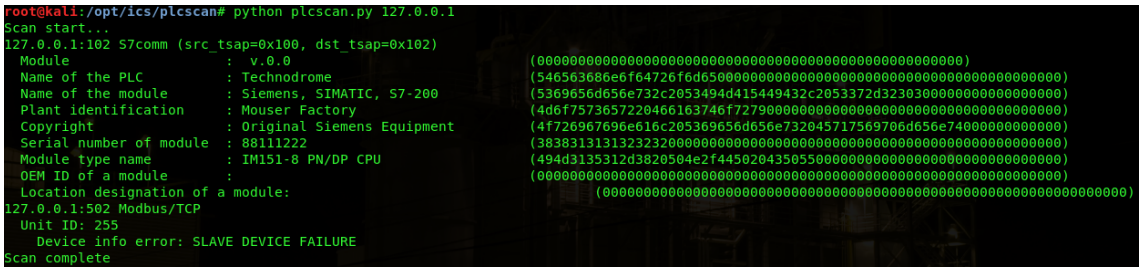


Figura 7 – PLCScan sobre el PLC

La información adicional adquirida con esta herramienta es:

Module type name: IM151-8 PN/DP CPU

PLCScan confirma la información recogida, aportando además el tipo de módulo, bajo la etiqueta Module type name: IM151-8 PN/DP CPU.

Tcp/502: Modbus TCP

Mediante el auxiliar de Metasploit `auxiliary/scanner/scada/modbusdetect` se detecta que el protocolo activo en el puerto tcp/502 se trata de Modbus TCP y que el UID del dispositivo esclavo es 1:

```
msf5 auxiliary(scanner/scada/modbusdetect) > show options

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target address range or CIDR identifier
   RPORT    502              yes       The target port (TCP)
   THREADS  1                yes       The number of concurrent threads
   TIMEOUT  10               yes       Timeout for the network probe
   UNIT_ID  1                yes       ModBus Unit Identifier, 1..255, most often 1

msf5 auxiliary(scanner/scada/modbusdetect) > setg RHOSTS 127.0.0.1
RHOSTS => 127.0.0.1
msf5 auxiliary(scanner/scada/modbusdetect) > run

[+] 127.0.0.1:502         - 127.0.0.1:502 - MODBUS - received correct MODBUS/TCP header (unit-ID: 1)
[*] 127.0.0.1:502         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figura 8 – Detección de Modbus TCP mediante Metasploit

[+] 127.0.0.1:502 - 127.0.0.1:502 - MODBUS - received correct MODBUS/TCP header (unit-ID: 1)

También mediante Metasploit, se recolectará información acerca de los esclavos y sus UID y sus identificadores:

```
msf5 auxiliary(scanner/scada/modbus_findunitid) > run
[*] Running module against 127.0.0.1
[+] 127.0.0.1:502 - Received: correct MODBUS/TCP from stationID  1
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 2 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 3 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 4 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 5 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 6 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 7 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 8 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 9 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 10 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 11 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 12 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 13 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 14 (probably not in use)
[*] 127.0.0.1:502 - Received: incorrect/none data from stationID 15 (probably not in use)
```

Figura 9 – Detección de estaciones Modbus

[+] 127.0.0.1:502 - Received: correct MODBUS/TCP from stationID 1

Se ha encontrado la estación con UID 1.

A través de la herramienta SMOD se enumerará las funciones de Modbus con las que es compatible el componente con UID 1. Esto será útil a la hora de interactuar con este componente del ICS a través de Modbus TCP o para la construcción de paquetes para, por ejemplo, llevar a cabo un MitM.

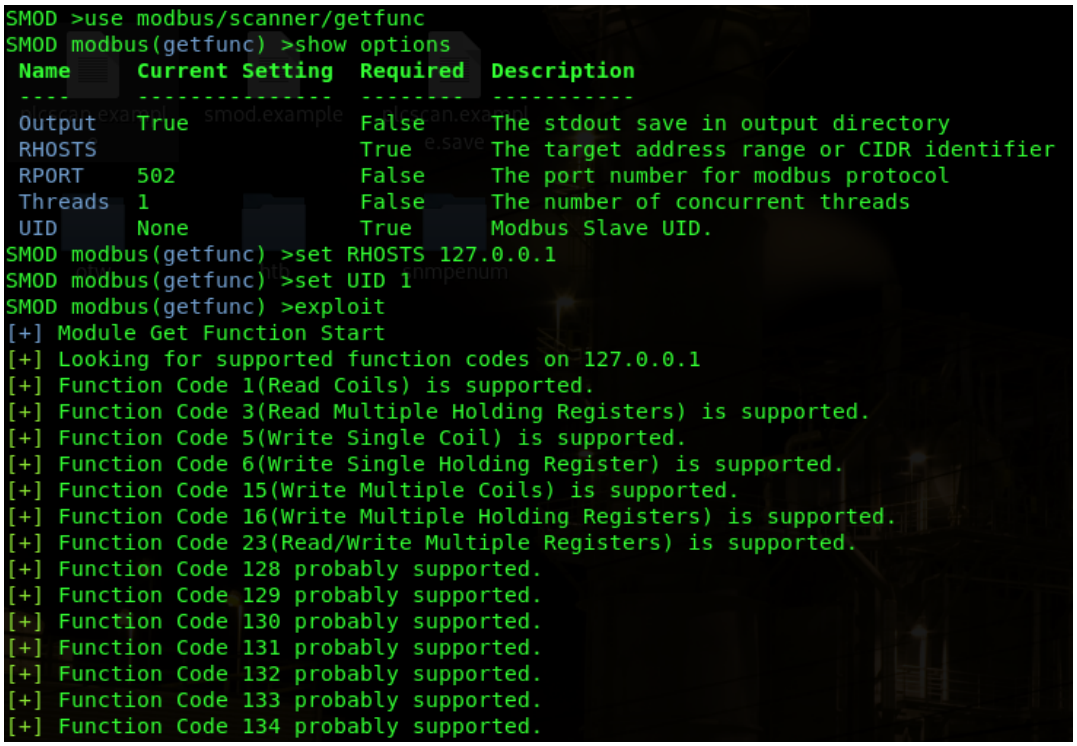


Figura 10 – Detección de funciones Modbus

```
[+] Looking for supported function codes on 127.0.0.1
[+] Function Code 1(Read Coils) is supported.
[+] Function Code 3(Read Multiple Holding Registers) is
supported.
[+] Function Code 5(Write Single Coil) is supported.
[+] Function Code 6(Write Single Holding Register) is supported.
[+] Function Code 15(Write Multiple Coils) is supported.
[+] Function Code 16(Write Multiple Holding Registers) is
supported.
[+] Function Code 23(Read/Write Multiple Registers) is
supported.
```

Las funciones compatibles la 1 (Read Coils), la 3 (Read Multiple Holding Registers), la 5 (Write Single Coil), la 6 (Write Single Holding Register), la 15 (Write Multiple Coils), la 16 (Write Multiple Holding Registers) y la 23 (Read/Write Multiple Registers). Del resto de funciones no se tiene certeza respecto a la compatibilidad.

Tcp/44818: EtherNet/IP

Para este protocolo se hará uso también de una serie de scripts de nmap: enip-enumerate.nse y enip-info.nse:

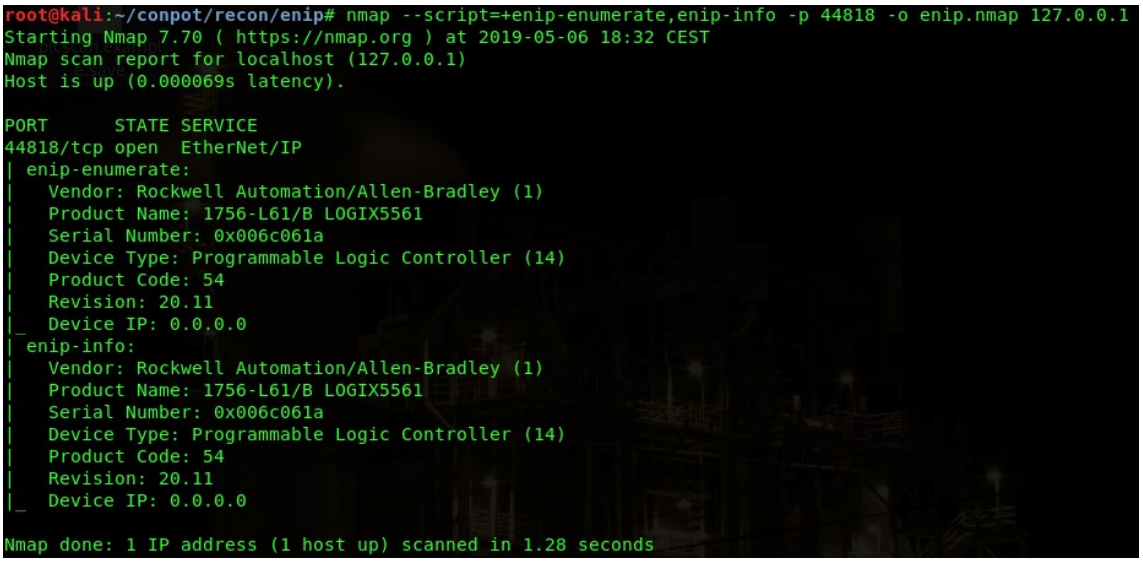


Figura 11 – Scripts de nmap para Ethernet/IP

La información recolectada por ambos scripts es la misma:

```
| enip-enumerate:
|   Vendor: Rockwell Automation/Allen-Bradley (1)
|   Product Name: 1756-L61/B LOGIX5561
|   Serial Number: 0x006c061a
|   Device Type: Programmable Logic Controller (14)
|   Product Code: 54
|   Revision: 20.11
|_  Device IP: 0.0.0.0
```

La herramienta ICSSPLOIT dispone de un módulo de escaneo para recuperar esta misma información pero no devuelve ningún tipo de información. Esto puede ser porque se trata de un simulador o bien porque realmente no funciona como debería:

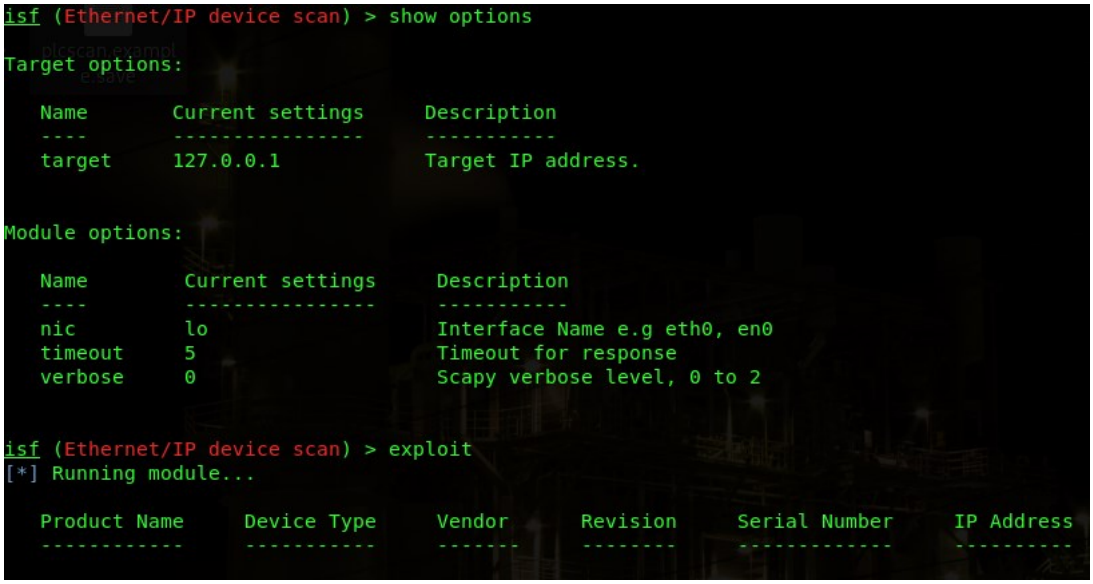


Figura 12 – Enumeración de EtherNet/IP con ICSSPLOIT

47808/tcp: BacNet

BacNet es un protocolo utilizado por sistemas de automatización y control de edificios (calefacción, ventilación, aire acondicionado, control de iluminación, de acceso, etc.) para el intercambio de información. Existe un script en nmap, *bacnet-info*, que devuelve información del fabricante, descripción, localización, versión de firmware, versión de software, etc. Al utilizar dicho script contra Conpot no ha devuelto información alguna.

```
root@kali:~/conpot/recon/ot/bacnet# nmap --script bacnet-info -o bacnet.nmap -sU -p47808 127.0.0.1
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-08 17:03 CEST
Nmap scan report for conpot (172.17.0.2)
Host is up (0.000045s latency).

PORT      STATE         SERVICE
47808/udp open|filtered bacnet
MAC Address: 02:42:AC:11:00:02 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 31.33 seconds
```

Figura 13 – Script de nmap relativo a BacNet

1.2 Protocolos IT

161/udp: SNMP

Si se encuentra este servicio activo en una auditoría de seguridad, es uno de los primeros a los que se tiene que acudir ya que, como se ha relatado en la memoria, puede proveer de mucha información acerca de la situación en la red del sistema a analizar. Por ejemplo, si se logra acceder a la información, se podría conseguir el listado de puertos abiertos, reduciendo así el ruido generado en la red al escanear el objetivo.

A través de Metasploit se comprobará la existencia de SNMP en el puerto 161/udp:

```
msf5 auxiliary(scanner/snmp/snmp_enum) > run

[+] 127.0.0.1, Connected.
[-] 127.0.0.1 SNMP request timeout.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figura 14 – Verificación de SNMP en Metasploit

En este caso se consigue la conexión pero arroja un error debido a un timeout por lo que no se logra obtener más información. Mediante otro auxiliar de Metasploit se confirma que sí se trata de SNMP, además se consigue información y acceso mediante el *comunity string public*. El *SNMP Community String* es como un id de usuario o contraseña que permite el acceso a los dispositivos que soportan SNMP (solo compatible con las versiones 1 y 2c de SNMP; la 3 utiliza el par usuario contraseña). Por defecto, el *community string* suele venir como “public”, como es el caso, siendo un error para la seguridad no cambiarlo:

```
msf5 auxiliary(scanner/snmp/snmp_login) > run

[!] No active DB -- Credential data will not be saved!
[+] 127.0.0.1:161 - Login Successful: public (Access level: read-write); Proof (sysDescr.0): Siemens, SIMATIC, S7-200
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figura 15 – Ataque de diccionario a SNMP

```
public (Access level: read-write); Proof (sysDescr.0): Siemens,
SIMATIC, S7-200
```

Esto confirma que se trata de un PLC Siemens S7-200 y que el *comunity string* es public, con acceso de lectura y escritura.

Esta información será utilizada para obtener toda la información posible a través de la herramienta snmpwalk:

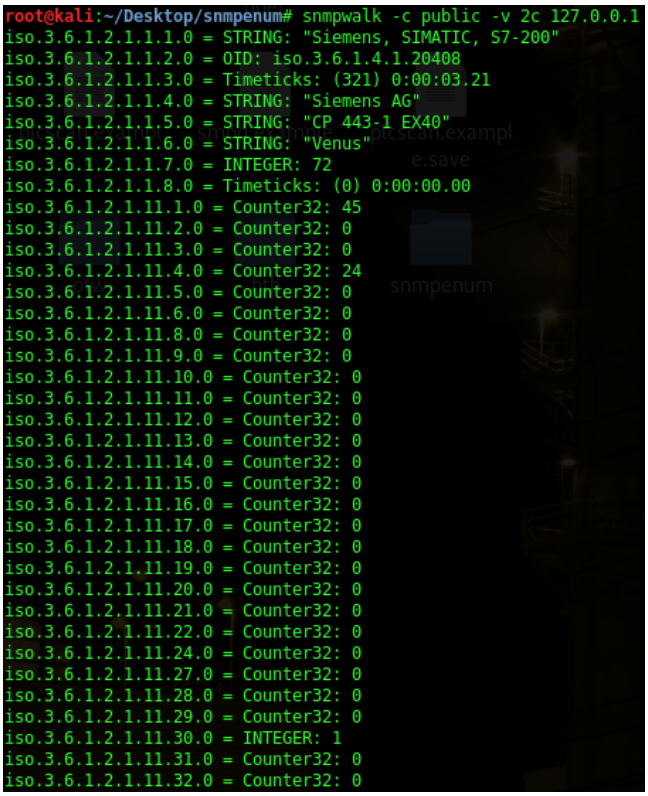


Figura 16 – Información SNMP mediante snmpwalk

iso.3.6.1.2.1.1.1.0 = STRING: "Siemens, SIMATIC, S7-200"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.20408
iso.3.6.1.2.1.1.3.0 = Timeticks: (321) 0:00:03.21
iso.3.6.1.2.1.1.4.0 = STRING: "Siemens AG"
iso.3.6.1.2.1.1.5.0 = STRING: "CP 443-1 EX40"
iso.3.6.1.2.1.1.6.0 = STRING: "Venus"
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
iso.3.6.1.2.1.1.8.0 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.11.1.0 = Counter32: 45
iso.3.6.1.2.1.11.2.0 = Counter32: 0
iso.3.6.1.2.1.11.3.0 = Counter32: 0
iso.3.6.1.2.1.11.4.0 = Counter32: 24
iso.3.6.1.2.1.11.5.0 = Counter32: 0
iso.3.6.1.2.1.11.6.0 = Counter32: 0
iso.3.6.1.2.1.11.8.0 = Counter32: 0
iso.3.6.1.2.1.11.9.0 = Counter32: 0
iso.3.6.1.2.1.11.10.0 = Counter32: 0
iso.3.6.1.2.1.11.11.0 = Counter32: 0
iso.3.6.1.2.1.11.12.0 = Counter32: 0
iso.3.6.1.2.1.11.13.0 = Counter32: 0
iso.3.6.1.2.1.11.14.0 = Counter32: 0
iso.3.6.1.2.1.11.15.0 = Counter32: 0
iso.3.6.1.2.1.11.16.0 = Counter32: 0

```
iso.3.6.1.2.1.11.17.0 = Counter32: 0
iso.3.6.1.2.1.11.18.0 = Counter32: 0
iso.3.6.1.2.1.11.19.0 = Counter32: 0
iso.3.6.1.2.1.11.20.0 = Counter32: 0
iso.3.6.1.2.1.11.21.0 = Counter32: 0
iso.3.6.1.2.1.11.22.0 = Counter32: 0
iso.3.6.1.2.1.11.24.0 = Counter32: 0
iso.3.6.1.2.1.11.27.0 = Counter32: 0
iso.3.6.1.2.1.11.28.0 = Counter32: 0
iso.3.6.1.2.1.11.29.0 = Counter32: 0
iso.3.6.1.2.1.11.30.0 = INTEGER: 1
iso.3.6.1.2.1.11.31.0 = Counter32: 0
iso.3.6.1.2.1.11.32.0 = Counter32: 0
iso.3.6.1.2.1.11.32.0 = No more variables left in this MIB View
(It is past the end of the MIB tree)
```

80/tcp: HTTP

Lo primero de todo es el acceso mediante el navegador y comprobar la funcionalidad de la página web, familiarizarse con su uso, etc.:

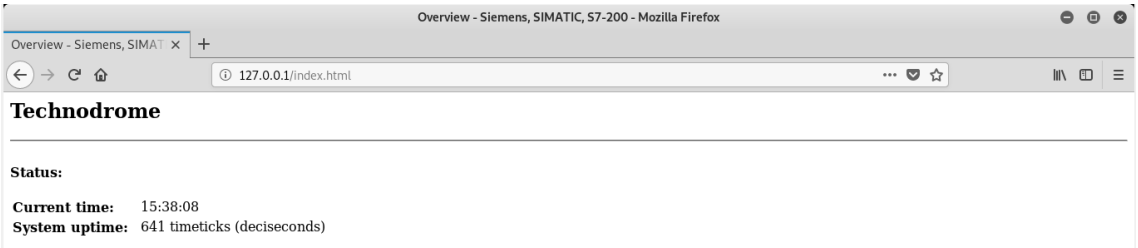


Figura 17 – Acceso mediante el navegador al servidor web del PLC

La interfaz web del PLC muestra información sobre el nombre del PLC, hora actual y tiempo activo.

El código fuente no muestra ninguna información relevante:

```
<HTML>

    <HEAD>
        <TITLE>Overview - Siemens, SIMATIC, S7-200</TITLE>
    </HEAD>

    <BODY>

        <h2>Technodrome</h2>
        <hr>
         <br>

        <b>Status:</b><br>
         <br>
        <table border="0">

            <tr>

                <td style="width:150px;"><b>Current time:</b></td>
                <td>15:38:08</td>

            </tr>

            <tr>

                <td style="width:150px;"><b>System uptime:</b></td>
                <td>641 timeticks (deciseconds)</td>

            </tr>

        </table>

    </BODY>

</HTML>
```

Figura 18 – Código fuente interfaz web PLC

No existe ningún enlace hacia otro punto de la web dentro de esta página. El siguiente paso es comprobar la información de las cabeceras de la petición web:

```
root@kali:~# curl -I http://127.0.0.1
HTTP/1.1 302 Found
Date: Tue, 07 May 2019 15:27:35 GMT
Content-Type: text/html
Location: /index.html
Content-Length: 0

root@kali:~# curl -I http://127.0.0.1/index.html
HTTP/1.1 200 OK
Date: Tue, 07 May 2019 15:27:35 GMT
Last-Modified: Tue, 19 May 1993 09:00:00 GMT
Content-Type: text/html
Set-cookie: path=/
Content-Length: 576
```

Figura 19 – Cabeceras HTTP

Las cabeceras HTTP no arrojan ningún tipo de información importante. Se realiza el establecimiento de la cookie `path=/`.

El siguiente paso es la fuerza bruta de directorios mediante un ataque de diccionario. Esto permitirá encontrar directorios o ficheros en el servidor web, para ello se utilizará gobuster. Se especificará, que además de directorios, busque ficheros con la extensión `.html`; se utilizará un diccionario con `22560` nombres:

```
root@kali:~/conpot/recon/it/http# gobuster -u http://127.0.0.1 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
-x html -o dir.gb -t 50

===============================================================
Gobuster v2.0.1              OJ Reeves (@TheColonial)
===============================================================
[+] Mode         : dir
[+] Url/Domain   : http://127.0.0.1/
[+] Threads      : 50
[+] Wordlist     : /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes : 200,204,301,302,307,403
[+] Extensions   : html
[+] Timeout      : 10s
===============================================================
2019/05/07 17:54:19 Starting gobuster
===============================================================
/index.html (Status: 200)
===============================================================
2019/05/07 18:58:32 Finished
===============================================================
```

Figura 20 – Fuerza bruta de directorios con gobuster

Lo que hace exactamente la herramienta es imprimir aquellos endpoints que a la hora de hacer una consulta devuelvan un código 200, 204, 301, 302, 307 o 403 en la respuesta. Ha encontrado el endpoint `/index.html`; que es la página de bienvenida analizada anteriormente.

Al acceder a una página que no existe muestra un mensaje de error personalizado, como se puede ver en la siguiente figura, pero no muestra ningún tipo de funcionalidad ni acceso a otro endpoint de la aplicación web. Dicho mensaje de error revela información acerca de un producto: el CP 443-1 EX40. Realizando una búsqueda en Internet se puede comprobar que se trata de un estándar que permite conectar al PLC a EtherNet/IP.

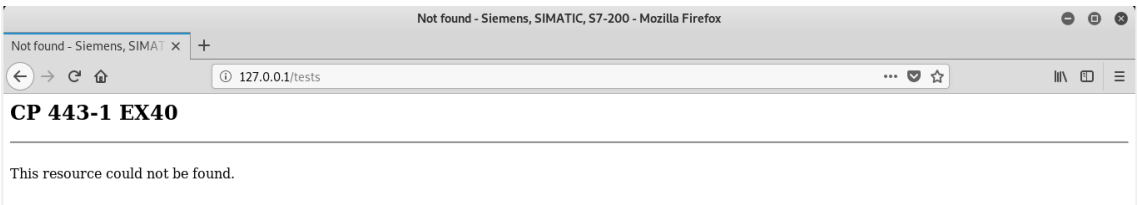


Figura 21 – Mensaje de error en el navegador

En un caso real, el PLC podría contar con un servidor web que sirviera para su administración remota, donde controlar su estado, etc. con formularios de autenticación, consultas a BBDD, comunicación con otros dispositivos… Toda esto tendría que ser auditado. En este caso, al tratarse de una simulación el servidor web carece de funcionalidades ni interacciones con los diferentes protocolos del PLC.

623/udp: IPMI

IPMI es un protocolo que permite la administración remota y total de un sistema. Metasploit dispone de un módulo auxiliar capaz de recoger la versión de este servicio:

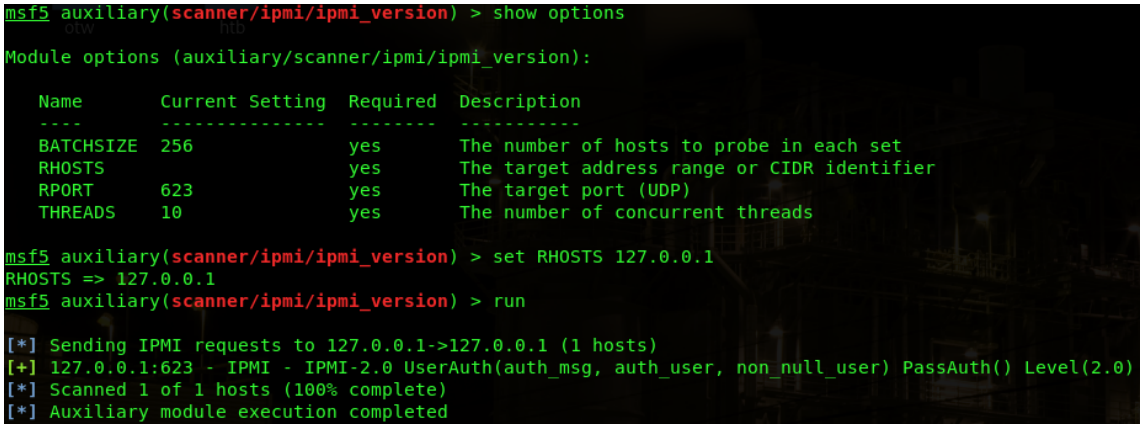


Figura 22 – Versión de IPMI con Metasploit

`IPMI-2.0 UserAuth(auth_msg, auth_user, non_null_user) PassAuth() Level(2.0)`

Soporta la autenticación 2.0 y el tipo de mensaje autenticación es `UserAuth(auth_msg, auth_user, non_null_user)`.

21/tcp: FTP

Por regla general, los servidores FTP muestran al cliente, al realizar la conexión, la versión del mismo. Por lo tanto, el primer paso será realizar esto, que se conoce como *banner grabbing*. Conseguir la versión del mismo permitirá la identificación de las posibles vulnerabilidades en la fase posterior.

Figura 23 – FTP banner grabbing

El servidor FTP no devuelve información acerca de la versión. Como siguiente paso en la enumeración de FTP se comprobará a ver si está habilitado el acceso como usuario anónimo, el cual no necesita contraseña:

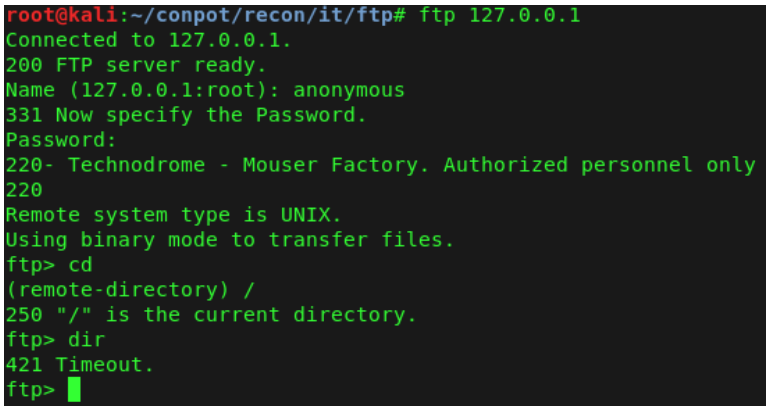


Figura 24 – Acceso anónimo al servidor FTP

Como se puede comprobar, se tiene acceso como anónimo al PLC sin proporcionar contraseña alguna pero a la hora de intentar listar los directorios da un timeout en la conexión.

69/udp TFTP

TFTP es similar a una versión básica de FTP utilizado generalmente para transferir pequeños archivos a través de UDP. Al tratarse de UDP no hay una definición de sesión, cliente ni servidor. Se considera servidor al que tiene abierto el puerto 69 y cliente al que se conecta. El "cliente" deberá enviar un mensaje al "servidor" solicitando el nombre de un archivo y el modo de transferencia. Metasploit ofrece un módulo para hacer un ataque de diccionario al nombre de los posibles archivos contenidos en el "servidor" tftp:

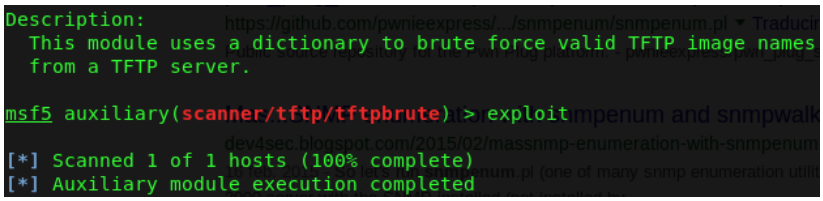


Figura 25 – Ataque de diccionario a archivos mediante Metasploit

El módulo no ha podido obtener ningún fichero. Esto se puede deber, al igual que ocurre con alguno del resto de protocolos, a que no se trata de un sistema de real, entendiéndose real como un sistema de producción, sino que se trata de un honeyot de baja interacción. Con lo que respecta a este protocolo no hay mucha más información que conseguir.

2. Identificación de vulnerabilidades

Respecto a la identificación de vulnerabilidades automatizada se utilizará Nessus, a través de un escaneo avanzado. Se ha configurado el escáner para que identifique vulnerabilidades tanto en protocolos OT como IT. Los resultados son los siguientes:

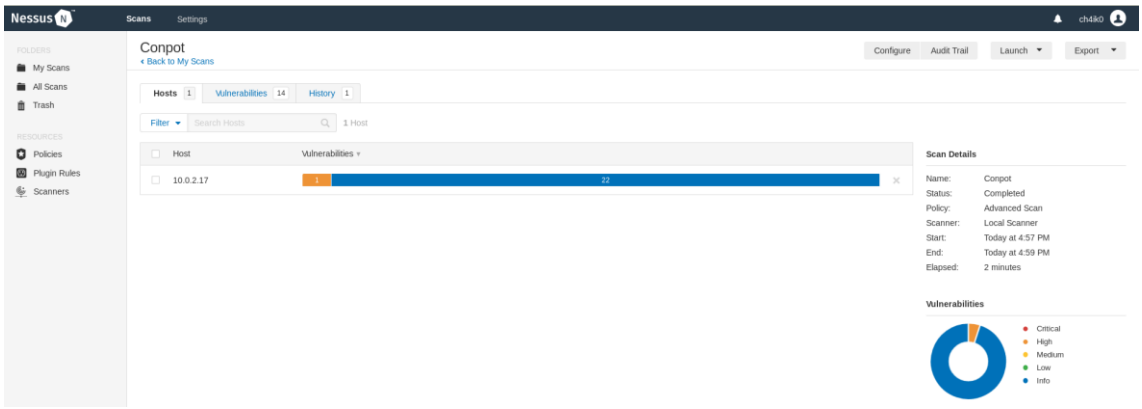


Figura 26 – Resumen de detección de Nessus sobre Conpot

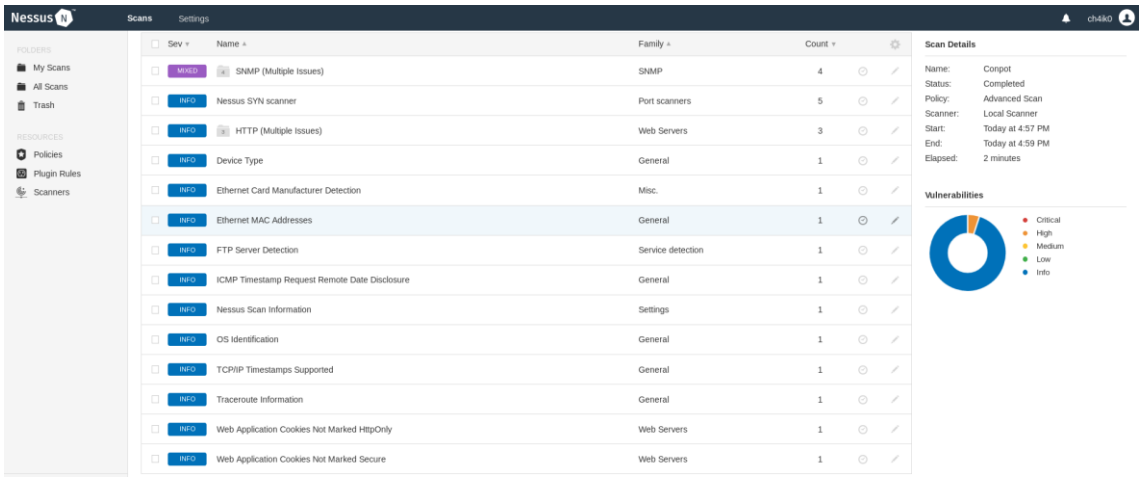


Figura 27 – Vulnerabilidades encontradas por Nessus I

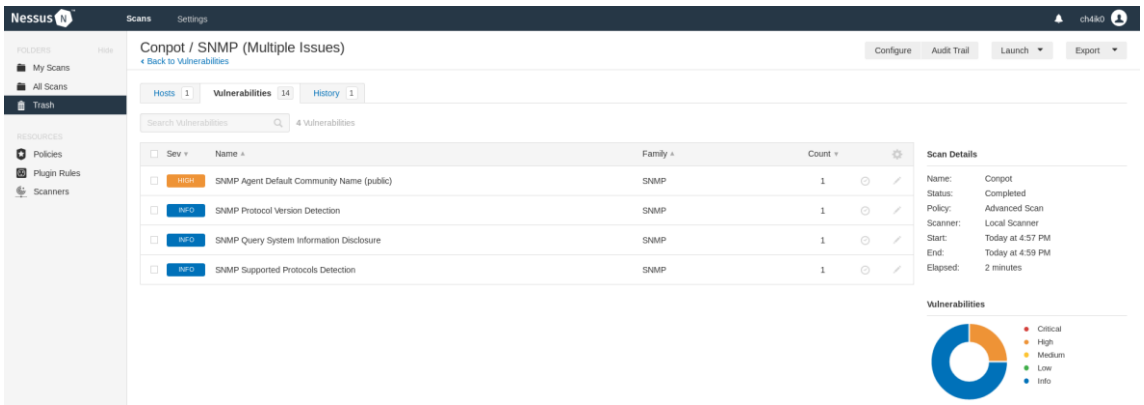


Figura 28 – Vulnerabilidades encontradas por Nessus II

Se han identificado un total de 14 vulnerabilidades, 13 de ellas informativas y una de nivel de criticidad medio. En concreto las más relevantes son las siguientes: detección de que se trata de un tipo de sistema embebido, que se trata de un dispositivo Siemens y que el *communiy string* de snmp es *public*, tal y como se detectó anteriormente:

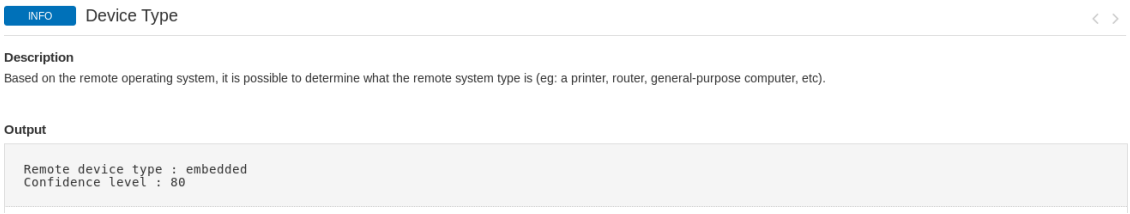


Figura 29 – Detección de tipo de sistema embebido

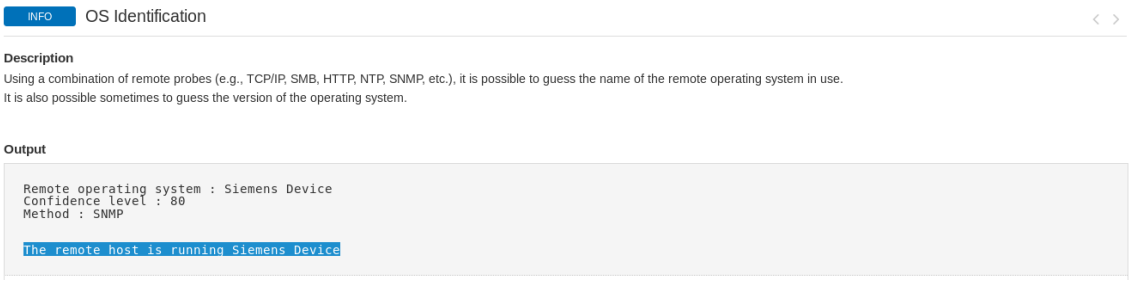


Figura 30 – Detección de sistema Siemens

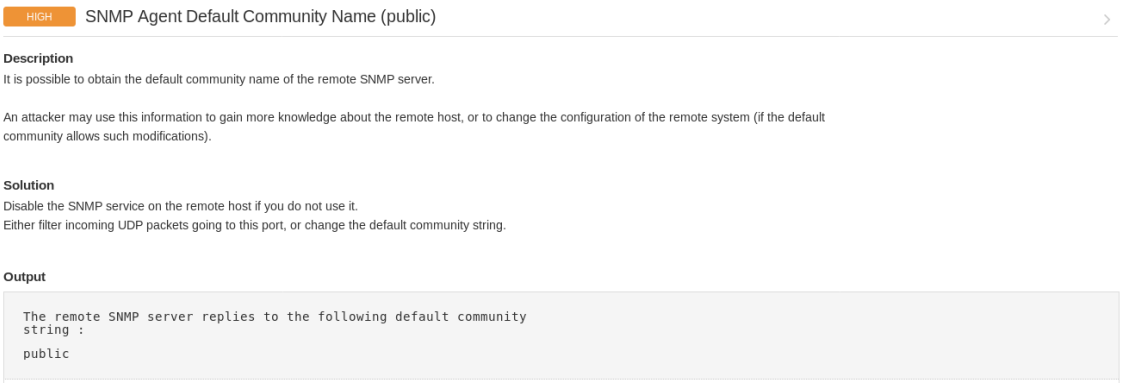


Figura 31 – Detección de community string public

En lo relativo a la identificación de vulnerabilidades manual, no se ha encontrado ninguna versión de ninguno de los protocolos IT en la fase de reconocimiento, en cuyo caso contrario se procedería a la búsqueda de vulnerabilidades en bases de datos de exploits, tal y como se relata en la metodología definida en la memoria. Tampoco se ha detectado, por ninguno de los dos métodos, ninguna vulnerabilidad grave o crítica. Esto se puede deber, en gran medida, a que el sistema a auditar es un honeypot de baja iteración, punto que se tratará en las conclusiones de la memoria.

El protocolo IPMI en su versión 2.0 tiene una vulnerabilidad que, conociendo un usuario, permite autenticarse sin conocer la contraseña. El cipher 0, que permite la autenticación en texto plano, contiene una vulnerabilidad que permite el acceso con cualquier password. Existe un módulo en Metasploit para comprobar la existencia de esta vulnerabilidad:

```
msf5 auxiliary(scanner/ipmi/ipmi_cipher_zero) > show options

Module options (auxiliary/scanner/ipmi/ipmi_cipher_zero):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   BATCHSIZE  256              yes       The number of hosts to probe in each set
   RHOSTS     127.0.0.1        yes       The target address range or CIDR identifier
   RPORT      623              yes       The target port (UDP)
   THREADS    10               yes       The number of concurrent threads

msf5 auxiliary(scanner/ipmi/ipmi_cipher_zero) > run

[*] Sending IPMI requests to 127.0.0.1->127.0.0.1 (1 hosts)
[+] 127.0.0.1:623 - IPMI - VULNERABLE: Accepted a session open request for cipher zero
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ipmi/ipmi_cipher_zero) > run

[*] Sending IPMI requests to 127.0.0.1->127.0.0.1 (1 hosts)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figura 32 – Ipmi_cipher_zero en Metasploit

Dicho auxiliar produce diferentes resultados en diferentes ejecuciones, por lo que puede que se trate de un falso positivo. Por el momento, no se conoce ningún usuario, por lo que no se puede probar a ciencia cierta de forma manual. El comando para probarlo sería, mediante una herramienta que permita la comunicación con este protocolo, como *ipmitool*, el siguiente:

```
$ ipmitool -I lanplus -C 0 -H IP -U USER -P CUALQUIERPASS
```

Si se realiza la prueba se obtiene un mensaje de error debido a que no soporta la autentiación por cipher 0, por lo que no es vulnerable:

```
root@kali:~/conpot# ipmitool -I lanplus -C 0 -H 127.0.0.1 -U Administrator -P AnyPassword
Authentication algorithm 0x01 is not what we requested 0x00

Error: Unable to establish IPMI v2 / RMCP+ session
```

Figura 33 – Intento de autenticación mediante cipher zero

De ser vulnerable, en la fase de explotación se podría crear una herramienta que hiciera un ataque de diccionario a todos los usuarios posibles. Una vez obtenido el usuario ya sería posible la autenticación mediante el cipher 0.

3. Explotación

3.1 Protocolos IT

Como se ha relatado en el final del punto anterior, la mayoría de los protocolos IT habilitados en el Conpot carecen de mucha interactividad que permita tanto la existencia como identificación y explotación de vulnerabilidades IT. El servidor web carece de funcionalidades, el servidor ftp no permite listar el contenido…; aunque cierto es que lo que más interesa con esta prueba de concepto es auditar los protocolos de naturaleza OT para poner en práctica lo descrito en la memoria.

Aun así, existen una serie de pruebas que realizar en algunos de los protocolos restantes. Respecto a SNMP, en la etapa de reconocimiento se ha detectado que, además de permisos de lectura, se tiene permisos de escritura. Para confirmarlo, se recuperará el valor de un OID, se modificará y se comprobará el cambio:

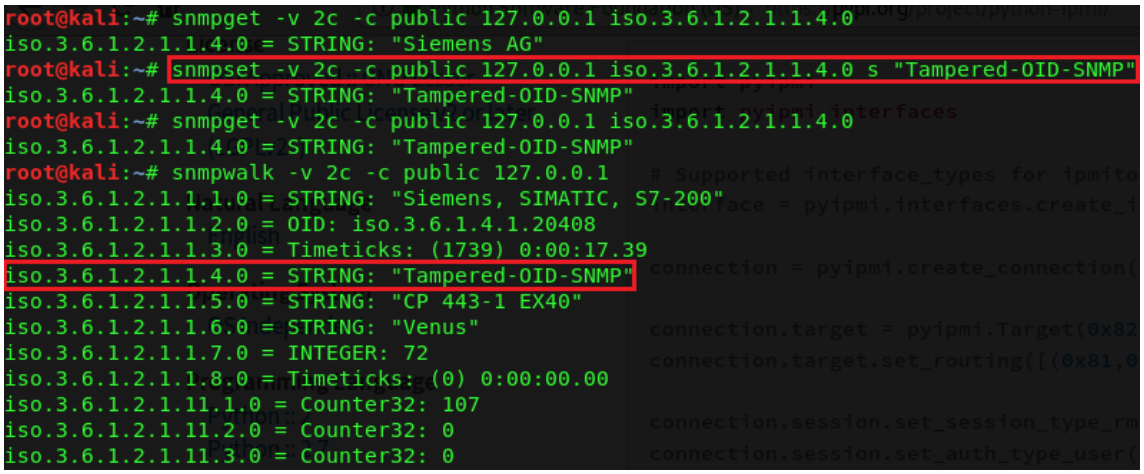


Figura 34 – Escritura mediante SNMP

3.2 Protocolos OT

A continuación se procede a detallar la fase de explotación de los protocolos OT habilitados en el PLC.

Lanzamiento de exploits naturaleza OT

Metasploit dispone de una serie de exploits pero que en este caso no aplican al PLC objetivo ya que no es de la marca para la cual están diseñados:

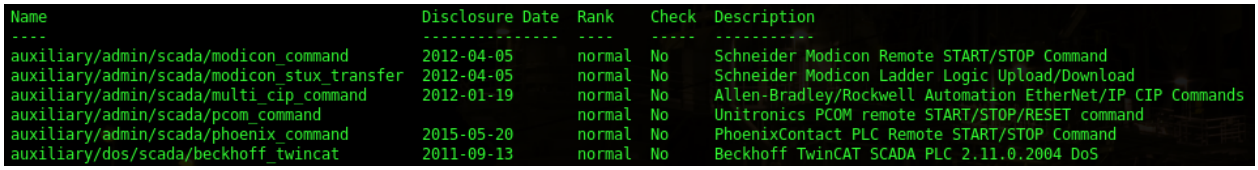


Figura 35 – Exploits relativos a PLCs en Metasploit

De entre esos exploits destacar los relativos a la toma del control del PLC – parada e inicialización – y el exploit dirigido hacia el PLC Schneider Modicon que permite descargar y modificar la programación del PLC (*PLC Ladder Logic*), una de las técnicas utilizadas por Stuxnet.

La herramienta ICSSPLOIT sí que ofrece exploits relativos a PLCs Siemens. Son los siguientes:

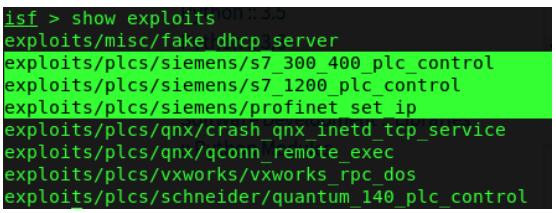


Figura 36 – Exploits relativos a PLCs Siemens en ICSSPLOIT

En este caso, Conpot simula un PLC Siemens S7-200, existe un exploit para S7-300/400, pero que podría funcionar también para esta versión, con el cual se puede parar y arrancar el PLC:

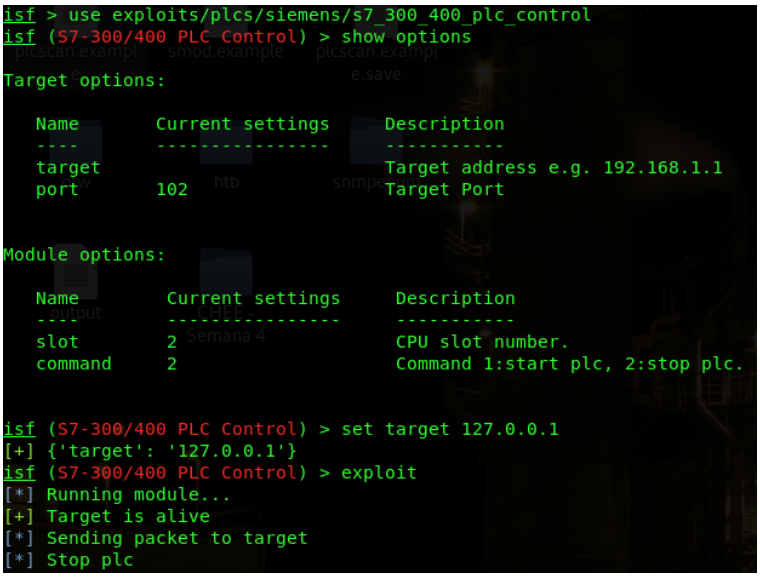


Figura 37 – Parada del PLC con ICSSPPLOIT

La salida del exploit no produce ningún error, con lo que parece que ha funcionado. En los logs de Conpot se puede observar una excepción, no queda muy claro si es porque no ha sabido cómo tratar el paquete o porque ha recibido una señal de apagado:

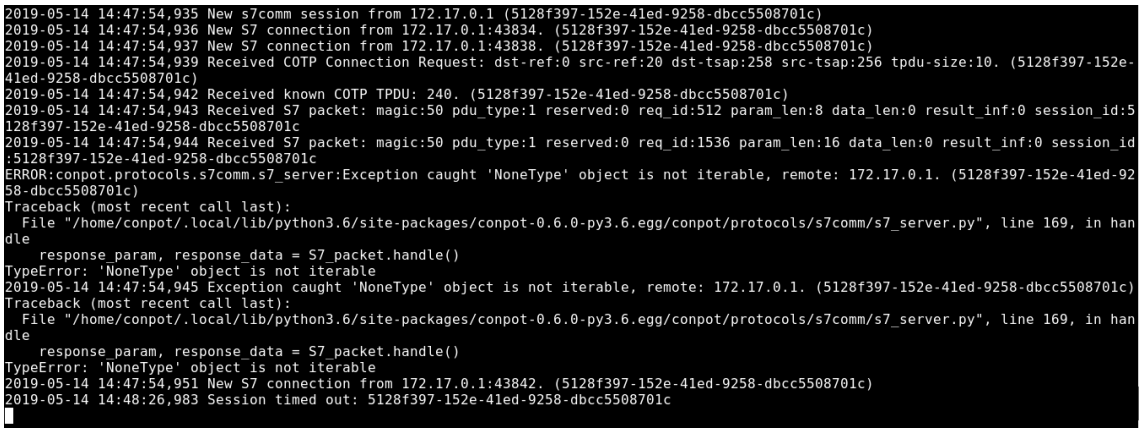


Figura 38 – Logs de Conpot tras exploit de parada del PLC con ICSSPPLOIT

Como curiosidad, si se lanza este exploit contra un simulador de un servidor que utilice S7Comm, en los logs de dicho simulador aparece que se ha recibido una señal de apagado:

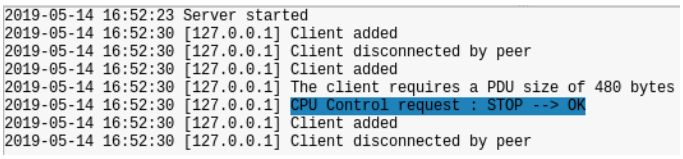


Figura 39 – Señal de apagado tras ejecución de exploit de ICSSPLOIT

Lectura/escritura de valores Modbus TCP

En este apartado se mostrará tanto lectura como escritura en registros y coils del PLC a través de Modbus TCP. En primer lugar se realizará sobre los registros, a través de Metasploit:

Lectura de registros

Para la lectura de registros habrá que especificar el UID o UNIT_NUMBER del esclavo, en este caso 1, el número de registros a leer, también 1, la dirección del registro, que también será 1, la dirección IP del PLC en RHOSTS, el puerto 502 en RPORT y la acción, READ_REGISTERS. Tras lanzar el módulo el resultado es 88.

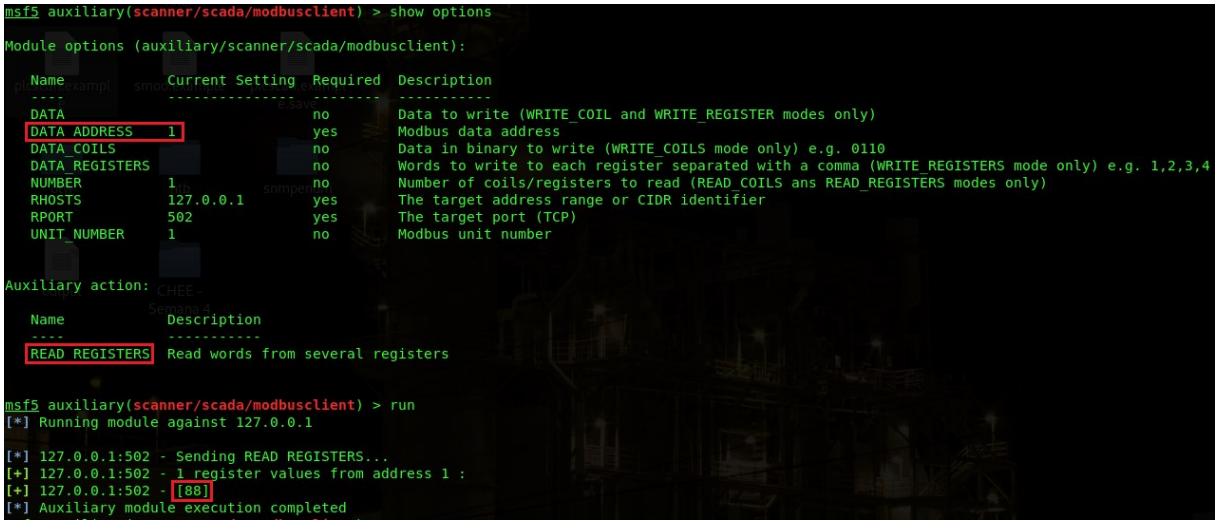


Figura 40 – Lectura de registro con Modbus TCP a través de Metasploit

Escritura de registros

La escritura se realizará sobre el mismo registro, para comprobar que se ha realizado correctamente; el UID será el mismo y el resto de opciones también. Lo único que variará es el campo DATA, donde se indicará el valor que tomará el registro tras la escritura, y la acción ACTION, que deberá ser WRITE_REGISTER:

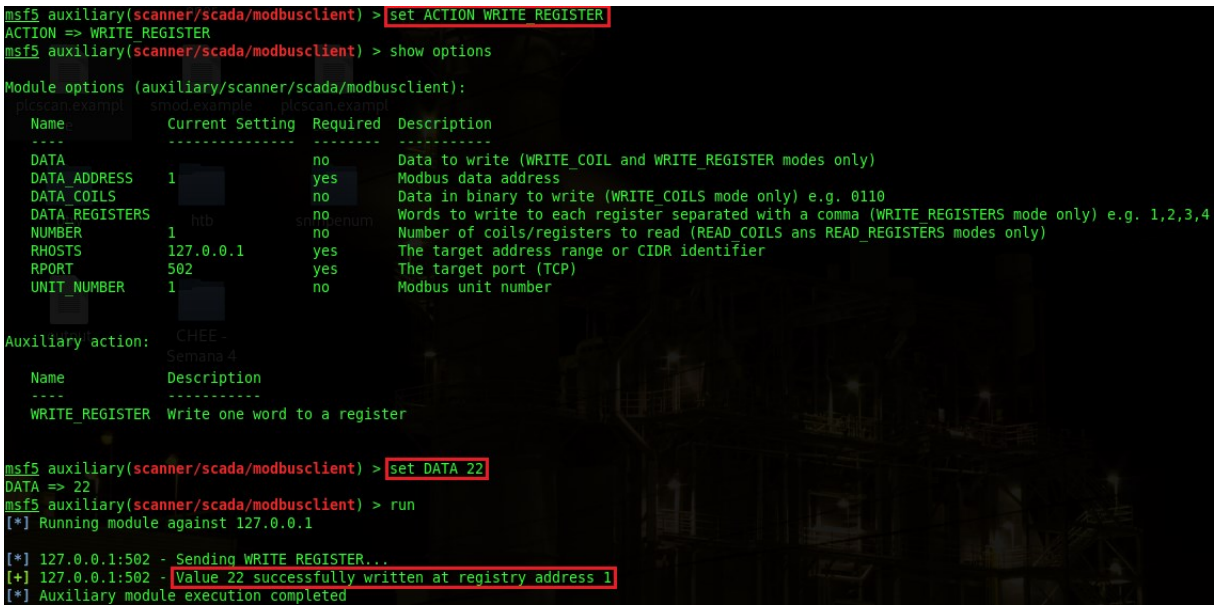


Figura 41 – Escritura de registro con Modbus TCP a través de Metasploit

Mediante el método anterior se realiza la comprobación para corroborar la escritura:

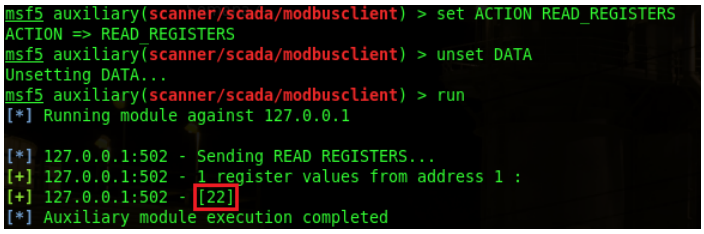


Figura 42 – Comprobación de escritura de registro con Modbus TCP a través de Metasploit

El siguiente paso es la lectura y escritura de Coils. Por alternar herramientas, se llevará a cabo mediante mbtget.

Antes de utilizar la herramienta se recuerda su uso mediante el despliegue de las opciones:

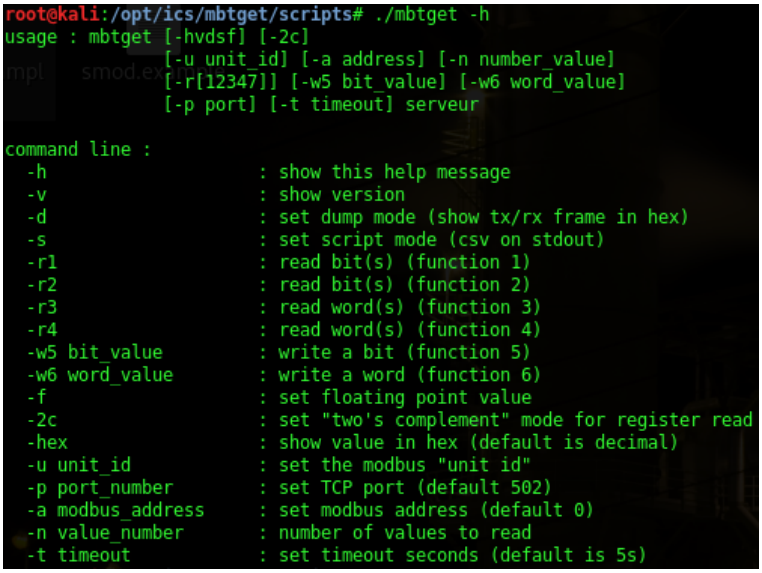


Figura 43 - Uso de mbtget

Lectura de coils

Se leerán los 12 primeros coils. Con la opción -u 1 se establece que el UID es 1, con -r1 la lectura mediante la función 1 y, por último, con -n 12 el número de coils a leer:

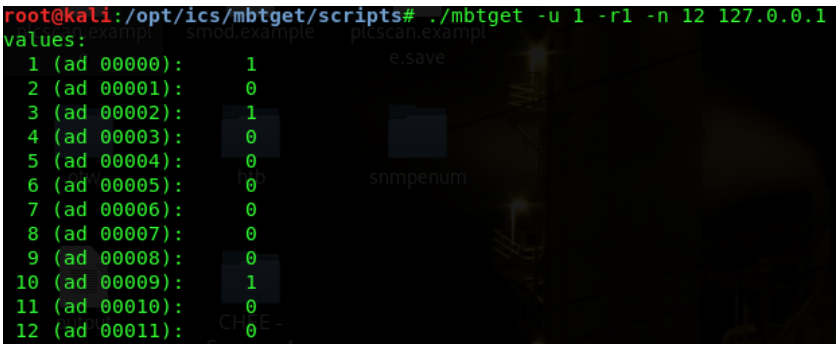


Figura 44 – Lectura de coils mediante mbtget

Escritura de coils

De la misma manera se realiza la escritura. Para ello, la función cambiará de -r1 a -w5 seguido del valor (0 o 1); con -a se indicará la dirección, en este caso la número 2 (ad 00002), cuyo valor actualmente es 1:

Figura 45 – Escritura de coil y comprobación mediante mbtget

Como se puede observar, llegado a este punto, con la posibilidad de comunicación directa con este componente del ICS, tanto la lectura y escritura de registros y coils se realiza sin ningún tipo de problema. Recordar que el PLC forma parte del nivel 1 de la jerarquía ISA-95 y está interactuando directamente con el proceso físico de producción, nivel 0. Así, poder realizar estos cambios en los registros supondrá una amenaza crítica para el ICS, al igual que ocurrirá con la lectura/escritura de valores sobre el protocolo S7Comm.

Lectura/escritura de valores S7Comm

La lectura y escritura con el protocolo S7Comm se llevará a cabo tal y como se ha realizado en la memoria: a través del cliente que proporciona la librería de código abierto Snap7. A la hora de conectar con ese cliente a Conpot, parece que no funciona del todo bien ya que a la hora de escribir/leer valores no se produce ningún cambio; además el log de Conpot muestra una serie de excepciones:

Figura 46 – Excepción S7Comm

Para ejemplificar la lectura/escritura, debido al error de Conpot, y ver que se puede aplicar de forma adecuada se utilizará el servidor de ejemplo de snap7:

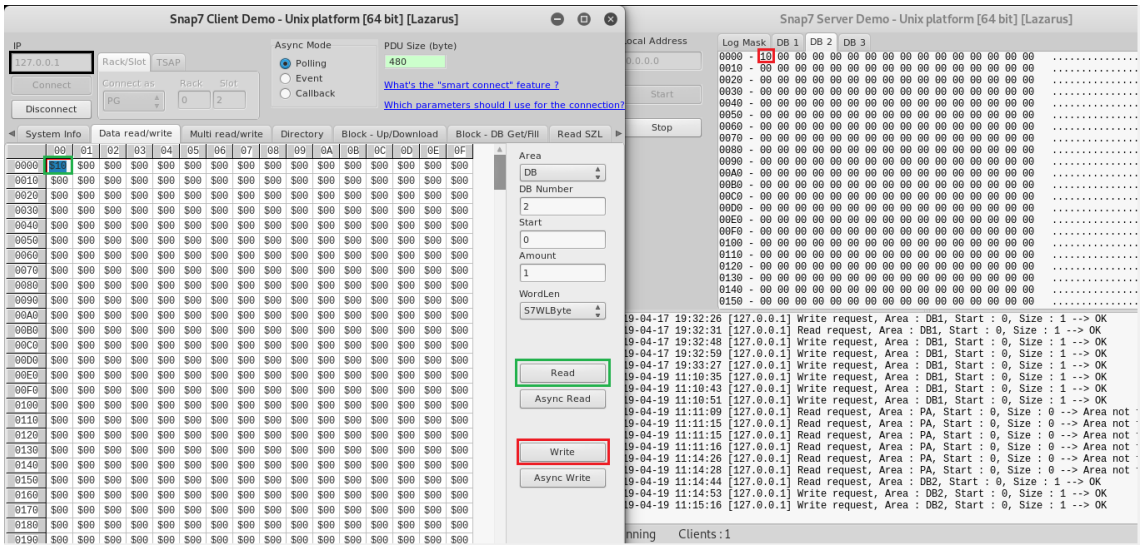


Figura 47 – Lectura y escritura PLC mediante S7Comm