Guía de análisis forense en Sistemas de Control Industrial

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

incibe cert_

Junio 2019

INCIBE_GUIA_ANALISIS_FORENSE_SCI _2019_v1

Índice

1. Sobre esta guía 5
2. Organización del documento 6
3. Introducción 7
3.1. La necesidad de tareas forenses en Sistemas de Control Industrial 8
4. Modelado de ataques en SCI 10
5. Análisis forense en SCI 13
5.1. Aspectos a tener en cuenta 13
5.2. Proceso de análisis forense 15
5.2.1. Identificación de activos 15
5.2.2. Detección de anomalías 18
5.2.3. Estudio de la amenaza 21
6. Herramientas para realizar un análisis forense en redes OT 25
6.1. Herramientas de extracción de información 26
6.2. Herramientas de análisis 27
6.3. Otras herramientas 28
7. Análisis forense de un malware específico de SCI 29
7.1. Identificación de activos 29
7.2. Detección de la anomalía 30
7.3. Estudio de la amenaza 32
8. Conclusiones 38
9. Glosario de términos y acrónimos 39
10. Referencias 40

Índice de figuras

Figura 1.- Dominios de análisis forense 8
Figura 2: Etapa 1 de Cyber Kill Chain para SCI 11
Figura 3: Etapa 2 de Cyber Kill Chain para SCI 12
Figura 4.- Pirámide de niveles según ISA-95 13
Figura 5.- Ubicación de los dispositivos en una red industrial (zonas y conductos) 17
Figura 6.- Arquitectura de red con sondas, IDS y SIEM 19
Figura 7.- Volcado de memoria con la herramienta FTK Imager 29
Figura 8.- Uso de reglas Yara para la detección de strings relacionados con el malware Stuxnet 30
Figura 9.- Estructura de la regla y búsqueda de la cadena de caracteres concreta ($op1) 31
Figura 10.- Salida por pantalla de la detección (Regla 1) 31
Figura 11.- Estructura de la regla y búsqueda de la cadena de caracteres concreta ($x1) 32
Figura 12.- Salida por pantalla de la detección (Regla 3) 32
Figura 13.- Información del sistema analizado a través del comando imageinfo 32
Figura 14.- Lista de procesos obtenida a través del comando pslist 33
Figura 15.- Opción 1 para obtener las instancias asociadas al ejecutable (lsass.exe) 33
Figura 16.- Opción 2 para obtener las instancias asociadas al ejecutable (lsass.exe) 34
Figura 17.- Sistema sin infección vs. sistema infectado (lsass.exe) 34
Figura 18.- Detección de inyección en proceso lsass.exe 34
Figura 19.- Detección de inyección en proceso svchost.exe 35
Figura 20.- Inyecciones DLL detectadas 35
Figura 21.- Nombres de ruta en blanco, PE en 0x80000 (ya se sabe inyectado) y falta de carga con llamadas LoadLibrary 35
Figura 22.- Comunicaciones detectadas a través del comando connscan 36
Figura 23.- Hostnames relacionados con los C&C de Stuxnet 36
Figura 24.- Ejemplo de Indicadores de Compromiso para Stuxnet 37

1. Sobre esta guía

Esta guía surge de la necesidad de concienciar sobre determinadas técnicas de análisis forense que se han aplicado de forma tradicional sobre los entornos corporativos y que se están empezando a aplicar sobre los sistemas de control.

En ella se recoge un conjunto de técnicas requeridas para llevar a cabo un análisis forense en el mundo industrial.

Se han tenido muy en cuenta las características que hacen especiales a los Sistemas de Control Industrial, planteando las diferentes fases del análisis forense, siempre orientadas hacia este entorno. Además, se proponen una serie de herramientas que pueden ayudar durante las diferentes fases, para concluir con un análisis forense llevado a cabo sobre una muestra de malware específica de un sistema de control.

2. Organización del documento

El documento comienza con una pequeña 3.-Introducción al mundo del análisis forense y a los últimos incidentes detectados a nivel industrial. Gracias a este primer apartado, el lector tendrá una visión global y resumida del estado actual de la ciberseguridad en los entornos industriales y de los peligros que entraña el malware desarrollado específicamente para este tipo de sistemas.

En el apartado 4.-Modelado de ataques en SCI, se describe el modelado de ataques en entornos industriales para entender, un poco más si cabe, los incidentes en Sistemas de Control Industrial. Esta parte ofrece información sobre las técnicas y tácticas utilizadas por los atacantes, centrándose de una manera especial en la fase inicial, donde se describen algunos de los vectores de ataque más comunes a la hora de iniciar un ataque en estos entornos.

Una vez dominados los conceptos más importantes, el lector se encontrará con una descripción del 5.-Análisis forense en SCI, donde se mostrarán los aspectos a tener en cuenta a la hora de realizar tareas forenses, destacando dónde encontrar información de valor a la hora de recopilar evidencias y cómo poder identificar activos y anomalías.

Presentados los métodos y procedimientos que utilizan los atacantes y los analistas forenses, se muestran algunas las 6.-Herramientas para realizar un análisis forense en redes OT más utilizadas.

Finalmente, para cerrar la guía y poner en valor todos los conceptos explicados, se incluye un caso práctico a modo de 7.-Análisis forense de un malware específico de SCI, donde se aplicarán muchos de los conceptos y herramientas mostrados, que proporcionarán una visión global de todo el proceso de modelado en ataques a entornos SCI.

Para finalizar, las 8.-Conclusiones exponen las principales ideas extraídas de esta guía.

3. Introducción

El mundo forense aplicado a la informática es una parte de las ciencias forenses que aprovecha técnicas de extracción y obtención de información para ser posteriormente examinada y clasificada.

Gran parte de las investigaciones forenses, incluidas las aplicadas en el ámbito de la informática, son realizadas por cuerpos de seguridad del estado, empresas que colaboran con los mismos o particulares que prestan cierto servicio como expertos externos. Estas investigaciones se realizan con el objetivo de obtener información que ayude a resolver casos en los que el uso de datos concretos, ha jugado un papel importante y de relevancia. La información a analizar suele encontrarse almacenada, de alguna forma, en algún dispositivo, por lo que el análisis forense para la extracción de evidencias cuenta con diferentes fases e implicaciones legales.

No obstante, el término forense suele relacionarse también con el análisis de datos en dispositivos tras un incidente, y es en este punto donde cobra todo el sentido en Sistemas de Control Industrial. Debido a la naturaleza única y poco común de las tecnologías que conviven dentro de los entornos industriales, a menudo existen grandes retos que hacen de una investigación forense una tarea bastante compleja. Gran parte de esa responsabilidad la tiene la disponibilidad, ya que ésta juega un papel fundamental en el mundo industrial. Por ese motivo, prácticamente la totalidad de las investigaciones forenses que se realizan a día de hoy, relacionadas con incidentes de ciberseguridad industrial, suelen durar varios meses tras la detección de los mismos, aunque la recuperación del sistema afectado no suele durar tanto tiempo.

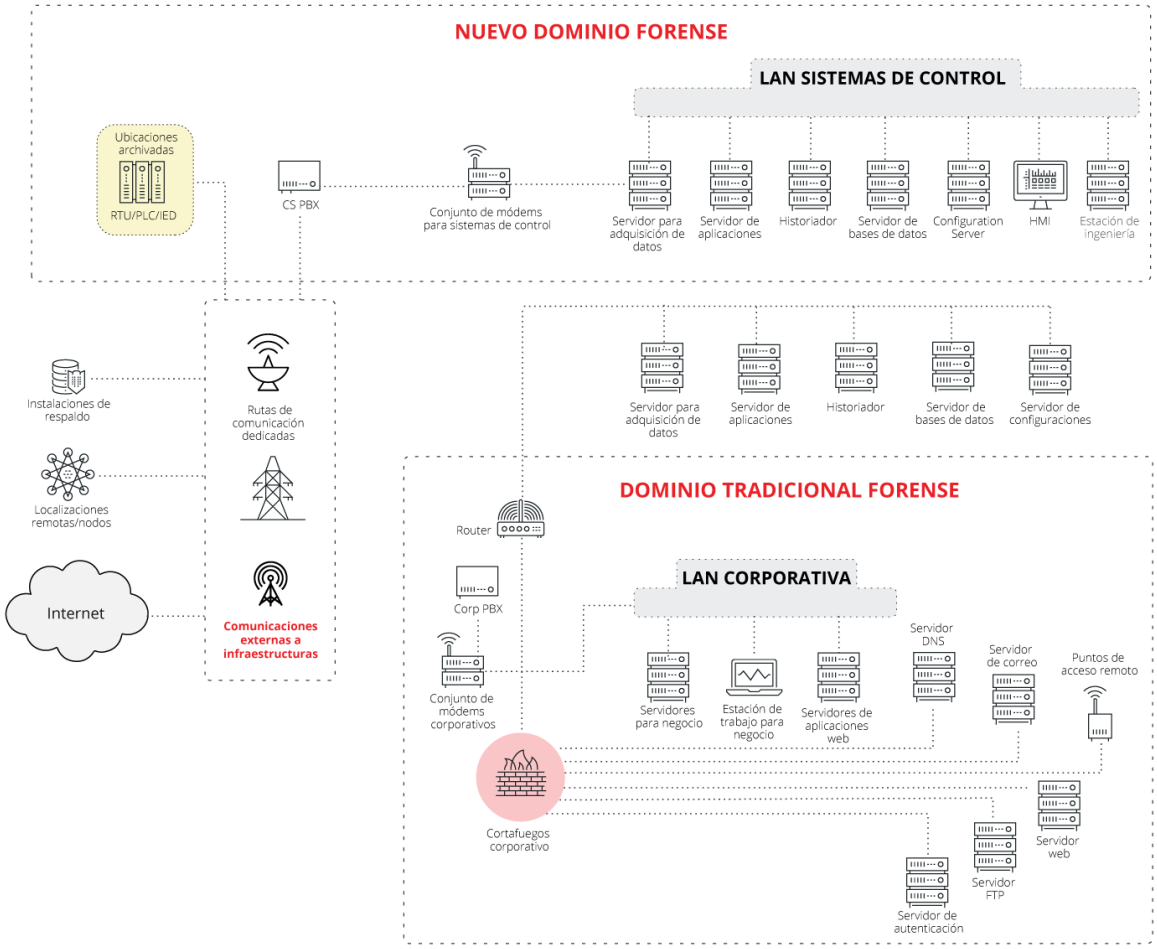

Figura 1.- Dominios de análisis forense

El uso de técnicas forenses para la obtención de información cuando se detecta un incidente, tanto en un Sistema de Control Industrial como en otro tipo de sistemas, ha de intentar responder siempre a las siguientes preguntas:

- ¿Quién ha realizado el ataque?
- ¿Qué ha pasado y qué impacto ha tenido?
- ¿Dónde se encuentran los sistemas comprometidos?
- ¿Por qué han sucedido los hechos?
- ¿Cuándo han sucedido?
- ¿Cómo se ha llevado a cabo el ataque?

3.1. La necesidad de tareas forenses en Sistemas de Control Industrial

A lo largo del tiempo, la proliferación de malware en diferentes entornos corporativos ha sido más que notable. Hoy en día, esta situación se ha extrapolado al mundo de las redes industriales, donde se están detectando diferentes evoluciones de muestras avanzadas relacionadas con malware, orientado directamente a entornos industriales.

Uno de los primeros casos detectados, y que supuso un punto de inflexión en la seguridad del mundo industrial tal y como se conocía anteriormente, fue Stuxnet[1]. Este malware fue una de las primeras "ciberarmas" desarrolladas para un objetivo y sector muy concreto dentro de los Sistemas de Control Industrial. Descubierto en 2010 por la firma de seguridad bielorrusa *VirusBlokAda*, Stuxnet fue una revolución en el mundo del malware en entornos industriales.

Stuxnet poseía muchas capacidades que, hoy en día, le harían un malware fácilmente detectable en la red gracias a las herramientas de análisis pasivo existentes en el mercado. Las alertas que se generarían por la forma de propagarse a través de recursos compartidos, colas de impresión, algunos servicios de Windows y actualizaciones punto a punto, originarían una gran cantidad de alarmas en prácticamente el total de las soluciones existentes. Desgraciadamente, por aquel entonces, la falsa sensación de seguridad heredada del pensamiento de que los sistemas aislados no podrían ser infectados tuvo gran parte de la responsabilidad. Igualmente, Stuxnet utilizaba varias vulnerabilidades *zero day* que le proporcionaban gran potencia, al ser indetectable en los entornos por la falta de monitorización y conocimiento sobre las propias vulnerabilidades.

Toda la información que sabemos hoy sobre la primera "ciberarma" detectada se debe en gran medida a las diferentes investigaciones forenses que se realizaron. Gracias a este tipo de investigaciones, tanto las organizaciones víctimas de los ataques como otras organizaciones relacionadas con el entorno atacado, pueden beneficiarse de los datos técnicos extraídos y de las lecciones aprendidas.

La creciente implantación de equipos de respuesta a incidentes, SOC - OT (*Secure Operations Center - Operational Technology*)[2] [Ref.- 1], que prestan un servicio de soporte a empresas industriales, supone un paso más en la lucha por la ciberseguridad dentro de los entornos industriales. Dentro de estos centros, estructurados por niveles, existe un equipo de forense dedicado exclusivamente a esta tarea.

Dado que las necesidades de un mayor conocimiento del malware y de los incidentes ocurridos es cada vez mayor, la especialización en análisis forense dentro de entornos industriales supone un nuevo campo a cubrir para muchos investigadores y centros de respuesta a incidentes. Sumado a esta necesidad, la falta de conocimiento sobre tipos de malware avanzados como GreyEnergy[3] [Ref.- 2], TRITON/HATMAN/TRISIS[4], CrashOverride[5] [Ref.- 3][Ref.- 19] , etc., detectados recientemente, hacen del campo forense en Sistemas de Control Industrial una disciplina interesante y que proporciona, ahora y en el futuro, grandes ventajas en la parte defensiva de muchas organizaciones industriales.

[1] https://www.incibe-cert.es/blog/amenazas-sci

[2] https://www.incibe-cert.es/blog/respondiendo-incidentes-industriales-soc-ot

[3]https://www.incibe-cert.es/alerta-temprana/bitacora-ciberseguridad/malware-greyenergy-amenaza-infraestructuras-criticas-2015

[4] https://www.incibe-cert.es/alerta-temprana/bitacora-ciberseguridad/triton-nuevo-malware-afecta-infraestructuras-industriales

[5] https://www.incibe-cert.es/blog/crashoverride-el-malware-sci-ataca-nuevo

4. Modelado de ataques en SCI

Uno de los pasos finales que suele aportar más valor en los análisis forenses es el de las lecciones aprendidas, así como la posibilidad de entender los vectores de ataque y funcionamiento del malware, para poder prevenirlo en caso de volver a verse afectado por él. Dentro de los incidentes, es importante tener en cuenta las Técnicas, Tácticas y Procedimientos (TTP) existentes a la hora de crear un modelado del ataque, para sacar unas conclusiones que puedan ayudar a centros de respuesta a incidentes, a las propias organizaciones al tanto de la información, etc.

Para poder realizar las diferentes clasificaciones dependiendo de la naturaleza del ataque, MITRE [Ref.- 4] posee diferentes matrices (dependiendo del sistema operativo) que podrían tener aplicación en los sistemas de control donde se recopilan las diferentes técnicas, tácticas y conocimientos comunes sobre el adversario. Gracias a estas matrices, y una vez analizado el malware en profundidad, los investigadores podrán ser capaces de realizar un modelado del ataque en base a las diferentes fases y el objetivo principal del mismo. Esta tarea no suele ser tan trivial como, por ejemplo, obtener los valores *hash* de ficheros implicados en el incidente, recopilar direcciones IP, servidores de dominio, etc.

A continuación, se muestra una breve descripción de los principales vectores de ataque utilizados en los Sistemas de Control Industrial:

- ***Spear phishing*:** variante de *phishing* que consiste en el envío de mensajes, generalmente de correos electrónicos, específicos y personalizados a un grupo de personas determinado, con el objetivo de obtener información sensible o infectar la máquina utilizada por la víctima. Esta es la principal diferencia respecto al *phishing* tradicional por email, que consiste en el envío de un mismo correo electrónico de forma masiva y al azar a millones de usuarios.
- **Adjuntos maliciosos:** prácticamente, la totalidad de los últimos malware detectados en Sistemas de Control Industrial posee como vector de ataque un correo con adjuntos maliciosos. Gran parte de estos documentos adjuntos necesitan de funcionalidades como las macros de Office o la activación de JavaScript en documentos para infectar a la víctima.
- ***Watering hole*:** esta técnica consiste en realizar un estudio del perfil u organización a atacar con el objetivo de detectar las webs más consultadas por sus víctimas. Una vez detectadas, serán analizadas en busca de vulnerabilidades que poder explotar para comprometer el sitio web y poder así infectar a sus visitantes, mediante diferentes recursos existentes, URL maliciosas, etc.
- **Superficie de exposición elevada:** este vector de ataque suele venir acompañado de un desconocimiento de las redes y servicios publicados que tienen algunas organizaciones, aumentando su nivel de exposición en Internet. En otros casos como el de GreyEnergy, la organización es consciente de que los servicios son públicos y accesibles desde Internet, pero no poseen todas las medidas de seguridad que deberían con respecto a la segmentación o el *hardening* del *host* que tiene público un servicio.

Para entender con mayor profundidad las diferentes fases implicadas en un incidente de ciberseguridad a nivel industrial, muchos profesionales se basan en el modelo Cyber Kill

Chain[6] acuñado por la empresa de seguridad Lockheed Martin. Este modelo fue escrito para comprender el camino seguido por un atacante a la hora de realizar una intrusión en un sistema.

Pasado un tiempo, a finales del año 2015, SANS Institute publicó un informe [Ref.- 5] adaptando el modelo Cyber Kill Chain a los sistemas de control. Este informe expande las fases de la Intrusion Kill Chain original para adecuarlos mejor a las características de la industria, además de dividirla en dos etapas. [Ref.- 6]

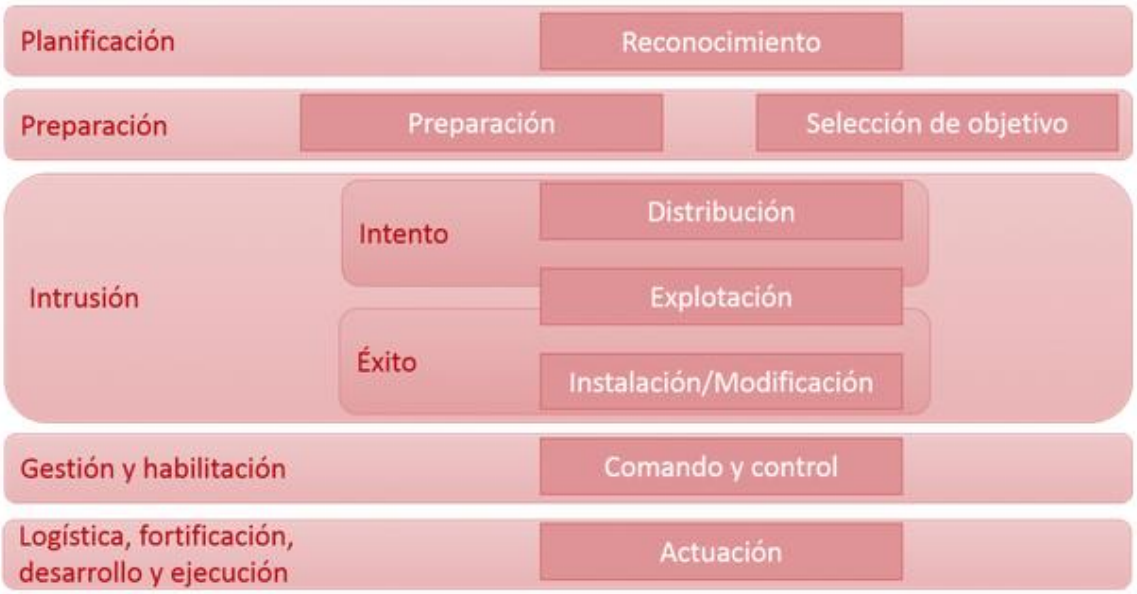

Figura 2: Etapa 1 de Cyber Kill Chain para SCI

Esta primera etapa guarda bastantes similitudes con el modelo original de Cyber Kill Chain. Se asocia sobre todo al mundo TI, ya que inicialmente el modelo planteado por Lockheed Martin, se centraba en entornos corporativos sin tener en cuenta los entornos de TO.

Una vez comprometido el sistema objetivo, en la adaptación realizada a TO por parte de SANS, se pasaría a una segunda etapa. En ocasiones, el compromiso de un sistema puede venir de forma indirecta gracias a la información extraída de un proveedor o colaborador, lo que hace innecesario todo el proceso de la etapa 1.

[6] https://www.incibe-cert.es/blog/cyber-kill-chain-sistemas-control-industrial

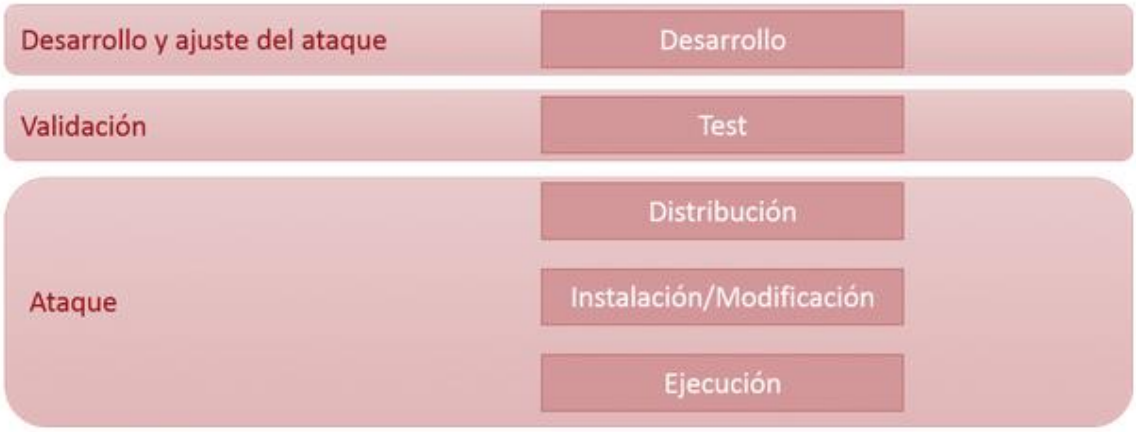

Figura 3: Etapa 2 de Cyber Kill Chain para SCI

Durante la etapa 2, más relacionada con el mundo TO en su parte de *Ataque*, se saca partido al conocimiento recogido en la etapa previa para elaborar un ataque dirigido. No sería necesaria la sucesión inmediata de la segunda etapa tras la primera ya que, algunas fases de esta podrían repetirse según la naturaleza del incidente.

La complejidad para llevar a buen puerto las dos etapas de la adaptación de Cyber Kill Chain al entorno industrial por parte de un atacante, dependerá de las medidas de seguridad que estén aplicadas en el sistema objetivo.

5. Análisis forense en SCI

El uso de técnicas forenses aplicadas a los Sistemas de Control Industrial, dada la situación de estos entornos, puede ser bastante compleja. Lo más común, teniendo en cuenta la pirámide de automatización, es aplicar estas técnicas en los niveles inferiores, 1 y 2, debido a que en ellos se suelen clasificar los activos más atacados dentro de un entorno industrial. En estos niveles encontramos tanto PLC (*Programmable Logic Controller*), RTU (*Remote Terminal Unit*), etc. (nivel 1), como sistemas SCADA (*Supervisory Control And Data Acquisition*), DCS (*Distributed Control System*), etc. (nivel 2).

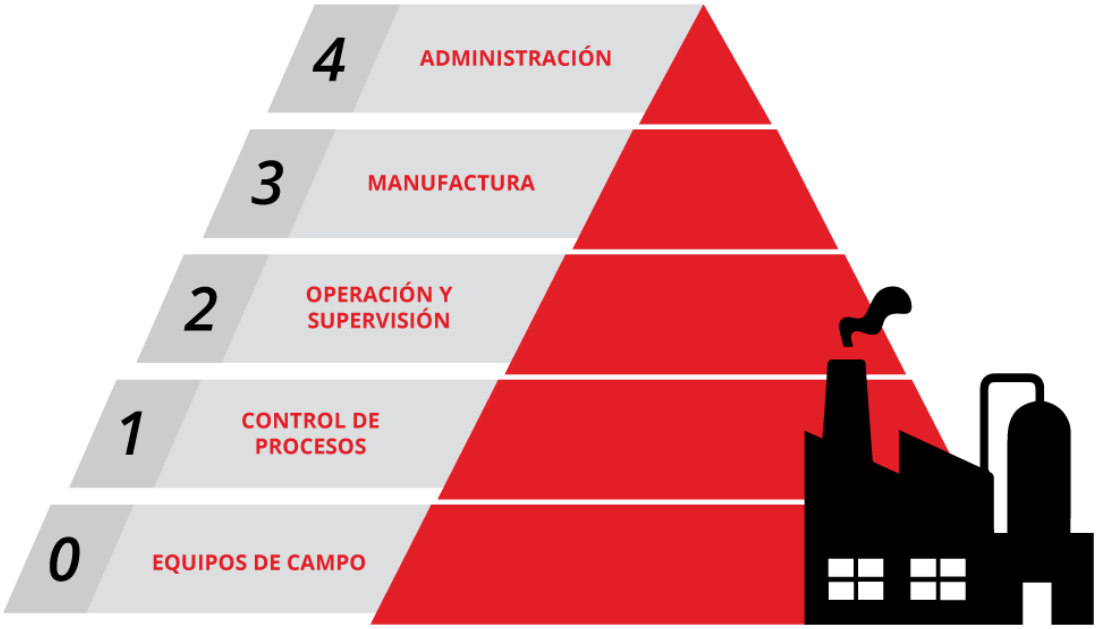

Figura 4.- Pirámide de niveles según ISA-95

Una vez detectados los niveles donde las técnicas forenses pueden ser más útiles dentro de los Sistemas de Control Industrial, también es fundamental tener en cuenta una serie de aspectos, ya que, en este caso y a diferencia de las redes corporativas, lo más importante es la disponibilidad.

5.1. Aspectos a tener en cuenta

El análisis forense en sistemas de control supone una serie de retos que giran en torno a la disponibilidad y a las capacidades que puedan tener los propios dispositivos que se encuentran desplegados en las instalaciones industriales. Para tener una idea más profunda de estos retos, a continuación, se resumirá de forma detallada los aspectos clave a tener en cuenta antes de llevar a cabo un análisis forense [Ref.- 6].

- **Las arquitecturas de los sistemas que utilizan muchos dispositivos a nivel industrial, no son muy conocidas.** Este hecho hace que algunas metodologías o procedimientos a aplicar no sean los más indicados o directamente sea imposible su utilización. A todos estos problemas, le podemos sumar la falta de

documentación, debido a que, en la industria, tanto los protocolos como algunas implementaciones son soluciones propietarias.

- **Gran parte de los dispositivos desplegados en producción no permiten la recopilación de información** que podría utilizarse tras un incidente de ciberseguridad aplicando técnicas forenses. En muchas ocasiones, los dispositivos industriales afectados por un malware, o a los que se les ha detectado algún tipo de anomalía de origen desconocido, no pueden ser apagados o sustituidos con el fin de adquirir datos.
- **Cadena de custodia.** Aunque el análisis del incidente no desemboque en un juicio, algo común en estos entornos dado que no se suele disponer del tiempo suficiente como para iniciar un proceso judicial, será necesario registrar aquellos eventos relacionados con las evidencias extraídas, desde su recolección hasta la presentación de las mismas en el propio juicio, sin dar la posibilidad a que sean modificadas por nada ni por nadie durante el proceso. Este registro busca la extracción adecuada de la prueba, la preservación, la individualización, el transporte apropiado y la entrega controlada.
- Es posible que **algunos análisis forenses requieran de la participación de los proveedores**. Añadir más intermediarios y responsables a un análisis forense puede retrasar de forma considerable los tiempos de entrega y la obtención de evidencias.
- Para la obtención de evidencias, además de la disponibilidad, hay que tener en cuenta otros dos conceptos más. Estos conceptos son el **orden de volatilidad** y la manera de llevar a cabo la **obtención de evidencias**.
    - **Orden de volatilidad:** velocidad con la que la información desaparece de un sistema o de una imagen digital extraída de un dispositivo. En general, el orden de volatilidad de los componentes de un dispositivo informático se clasifica de la siguiente forma (del más al menos volátil):
        - **Contenidos de la memoria caché y registros del sistema**.
        - **Información de red** (tabla de rutas y caché ARP).
        - **Memoria** (contiene gran parte de información, como la de red).
        - **Procesos del sistema**.
        - **Sistema de ficheros temporal**.
        - **Datos del disco duro**.
        - **Datos sobre el registro de acceso remoto**.
        - **Datos de medios extraíbles**.
    - **Obtención de evidencias:** es posible realizar la adquisición de dos formas:
        - **Local:** esta forma de adquirir datos es rápida y suele ser bastante sencilla. Es la opción que debe utilizarse siempre que sea posible. Esta alternativa para obtener datos, ha de testearse previamente en un entorno de preproducción o laboratorio, teniendo en cuenta el orden de volatilidad, obtener *snapshots* del dispositivo o sistema, etc.
        - **Remota:** la forma de adquirir datos, en este modo, es más lenta que en local y es necesario tener diferentes consideraciones antes de realizarse. Se debe disponer de acceso por red al dispositivo afectado desde una red externa, conocer si las comunicaciones suponen un riesgo para la disponibilidad y buen funcionamiento del

dispositivo afectado, si la red soportará el ancho de banda generado en las operaciones, etc.

5.2. Proceso de análisis forense

Dado que en los entornos industriales se intercambian gran cantidad de datos en algunos segmentos de red, como puede ser el segmento dedicado a la monitorización, es importante tener en cuenta qué información será de interés y cuál no a la hora de llevar a cabo un proceso forense.

Actualmente, existen herramientas en el mercado que facilitan este tipo de monitorización y detección de anomalías, ya sea mediante el uso de sondas para realizar capturas en modo pasivo o mediante el despliegue de agentes en dispositivos industriales. Esta última opción permite monitorizar ciertas partes de interés a nivel forense, de tal forma que si ocurre un incidente, la detección sea rápida, pero, como contrapartida, puede originar un consumo elevado de recursos nada beneficioso para los dispositivos industriales.

Para extraer la información importante dentro de un análisis forense, es necesario entender tanto las redes industriales, como los activos que se encuentran en las mismas.

5.2.1. Identificación de activos

A la hora de clasificar los activos que pueden encontrarse en una red industrial para su posterior análisis, hay que tener en cuenta varios aspectos, entre los que destacan:

- **Nivel de la pirámide de automatización al que pertenece.**
  - **Nivel 0:** se corresponde con los equipos de campo englobados en el propio proceso productivo, es decir, se compone de los sensores y actuadores que forman parte del proceso industrial.
  - **Nivel 1:** correspondiente al control de procesos. Es el nivel en el que se produce la interacción entre la parte física (nivel 0) y los sistemas de control más básicos, los PLC, DCS, etc.
  - **Nivel 2:** se corresponde con los equipos de operación y supervisión, ya sea de manera local mediante equipos HMI (*Human Machine Interface*) o de una manera centralizada mediante el uso de SCADA.
  - **Nivel 3:** donde se encuentran los sistemas MES (*Manufacturing Execution System*). Este nivel controla el flujo de la producción y también el almacenamiento de la información del proceso.
  - **Nivel 4:** agrupa las actividades relacionadas con el de ERP (*Enterprise Resource Planning*) o BI (*Business Intelligence*) en una organización industrial, en la red corporativa, etc.
- **Impacto que tendría para la empresa una parada o un funcionamiento incorrecto del activo**. Su alcance se medirá gracias a un análisis de riesgos y un estudio BIA (*Business Impact Analysis* – Análisis de impacto en el negocio), del que saldrá un nivel que puede asemejarse a la siguiente clasificación:
  - **Alto:** cuando suponga una gran pérdida económica para la organización. Además, este funcionamiento erróneo afecta al proceso, que tiene altas probabilidades de pararse o de no ejecutarse de forma correcta.

  - **Medio:** cuando suponga una pérdida leve a nivel económico para la organización. Con respecto al proceso, se tienen pocas probabilidades de que este se pare o se ejecute de forma incorrecta.
  - **Bajo:** cuando no supone una pérdida económica para la empresa o cuando el proceso no se verá alterado en ningún momento.

- **Tareas que realiza dentro del proceso industrial**. Entre las tareas a realizar se propone la siguiente clasificación:

  - **Supervisión:** permite realizar acciones de monitorización con el objetivo de analizar los procesos ejecutados.
  - **Control:** posee capacidades para modificar el proceso ejecutando acciones.
  - **Alimentación:** abastece de energía a otros, tanto en caso de un corte inesperado de suministro, como de un estado normal.
  - **Comunicaciones:** cuyo objetivo es gestionar comunicaciones tanto internas, como externas dentro de un proceso.

- **Ubicación de los activos dentro de la red de tecnologías de operación.**

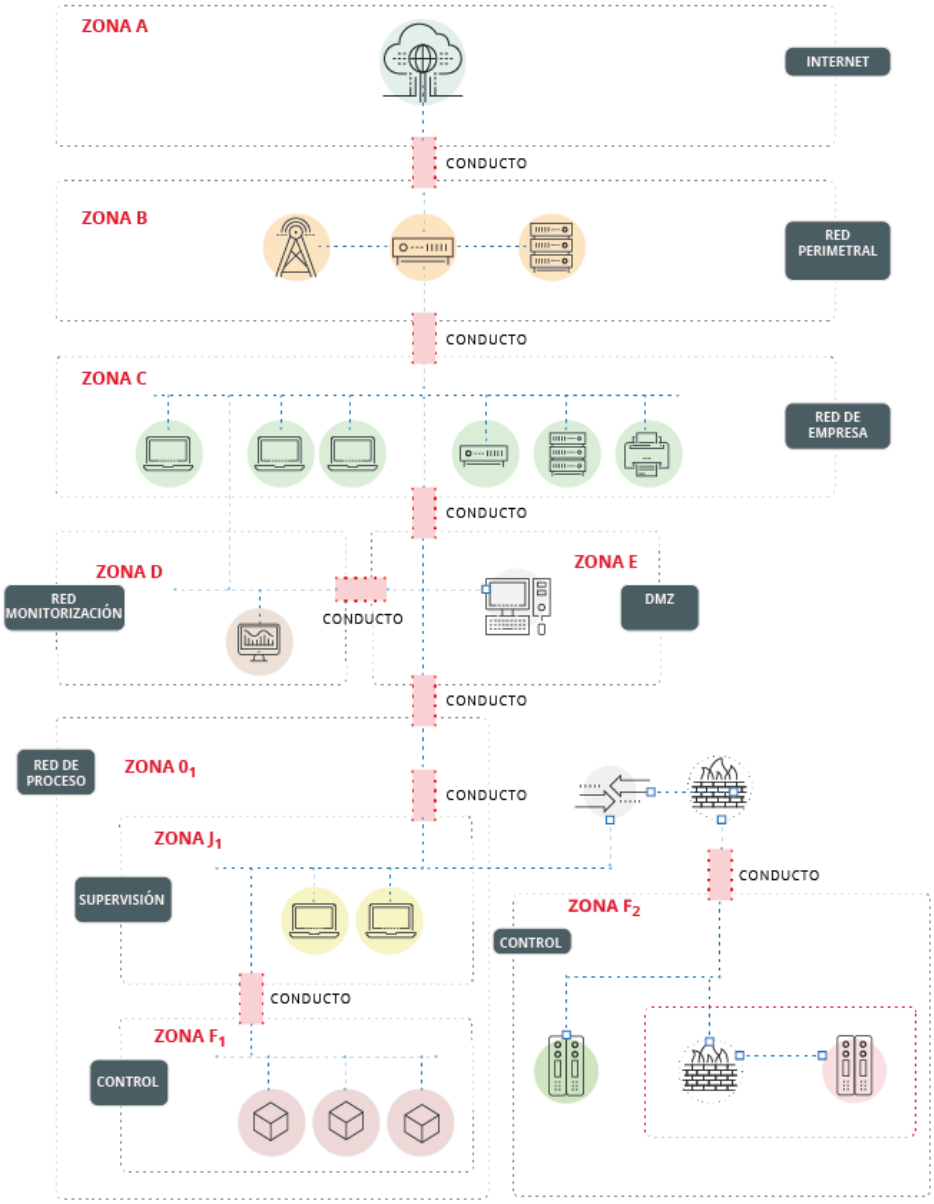

Figura 5.- Ubicación de los dispositivos en una red industrial (zonas y conductos)[7]

Basándose en la ISA-95 ya comentada, el principal objetivo de una organización industrial con respecto a la separación de los diferentes elementos dentro de una red industrial, en zonas y conductos, es crear una arquitectura de red segura. El modelo de arquitectura de red no es único, sino que está basado en guías de buenas prácticas, experiencias y, sobre todo, en las necesidades y limitaciones de cada caso en particular, por lo que pueden coexistir diferentes aproximaciones de arquitecturas seguras, incluso dentro de una misma empresa. Con el fin de catalogar correctamente los activos que existen dentro de cada red, se aconseja a la organización seguir este tipo de modelo:

[7] https://www.incibe-cert.es/blog/zonas-y-conductos-protegiendo-nuestra-red-industrial

- El concepto de “**zonas**” constituye uno de los recursos más importantes, y su definición es uno de los aspectos fundamentales para el éxito de este proceso. Las zonas de seguridad se definen en el estándar IEC-62443 como “agrupaciones de activos físicos o lógicos que comparten requisitos comunes de seguridad, las cuales tienen la frontera (física o lógica) claramente definida”.
- Los “**conductos**” son zonas particulares que se aplican a procesos de comunicación específicos, proporcionando funciones de seguridad que permiten a dos zonas comunicarse de manera segura. Toda comunicación entre diferentes zonas ha de realizarse a través de un conducto.

Las zonas y los conductos han de quedar descritos en diferentes campos dentro de un documento. Ejemplo de esta definición se puede ver en el estándar IEC-62443-3-2, siendo algunos de los campos:

- Nombre o identificador único de la zona/conducto.
- Límites lógicos.
- Límites físicos, si aplica.
- Lista de todos los puntos de acceso y de todos los activos implicados.
- Lista de todos los tipos de flujos de datos / protocolos asociados con cada punto de acceso.
- Zonas y conductos conectados.
- Lista de activos.
- Nivel de seguridad asignado.

5.2.2. Detección de anomalías

La detección de anomalías en los Sistemas de Control Industrial es una de las tareas que más se está potenciando en la actualidad, gracias al despliegue de sondas para la recolección de información de forma pasiva. Entre los mecanismos de detección y gestión a incorporar, no podrían faltar la inclusión de IDS (*Intrusion Detection System*) y de un SIEM (*Security Information and Event Management*), tanto para la generación de alertas en base a patrones ya conocidos a nivel de red, como para la correlación de *logs* y eventos.

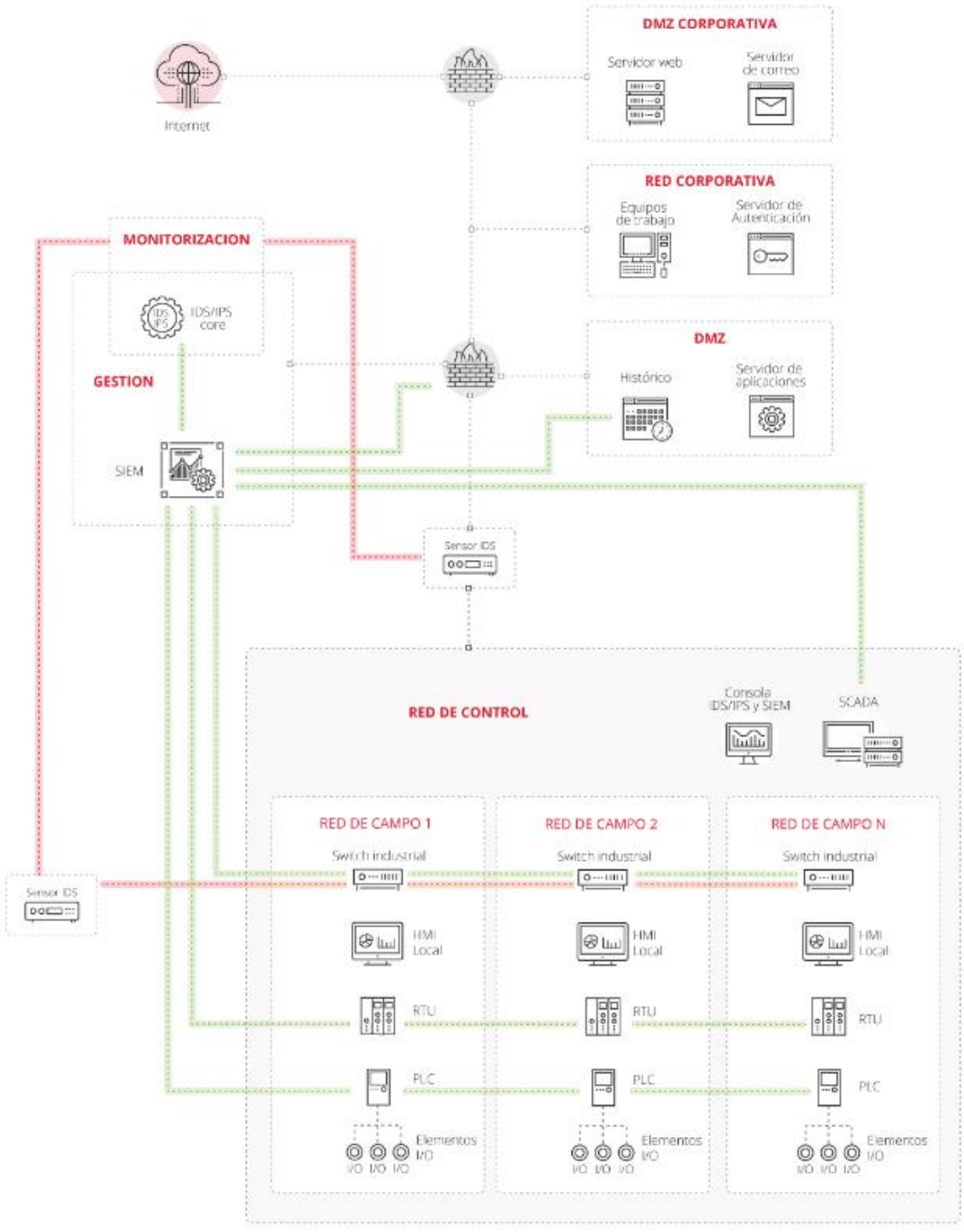

Figura 6.- Arquitectura de red con sondas, IDS y SIEM[8]

[8] https://www.incibe-cert.es/blog/diseno-y-configuracion-ips-ids-y-siem-sistemas-control-industrial

Además de estos mecanismos a nivel de red, existen otros como los HIDS (*Host-based Intrusion Detection Systems*) o los agentes que se ejecutan en los propios activos para la monitorización de ciertas características.

A nivel de activo y, concretamente, teniendo en cuenta las características que aporten información de valor en caso de incidente y su posterior análisis forense, se aconseja contar, entre otros, con:

- **Sistema de hora que posee como referencia el dispositivo.** Uno de los puntos clave a la hora de realizar un análisis forense es establecer el marco temporal en el que un incidente ha sucedido. Por ello, datos como la hora del dispositivo, la franja horaria configurada y el servidor NTP, SNTP, PTP, etc., que utiliza para la sincronización de tiempo, es muy importante. Algunos ataques detectados a nivel industrial modificaban estos parámetros para entorpecer las tareas forenses.
- ***Logs* de actividad.** Otro de los puntos clave a analizar en un dispositivo industrial son los *logs* que ha creado el mismo. Por desgracia, muchos de los dispositivos desplegados en el mundo industrial no disponen de mecanismos o capacidad suficiente para crear *logs* en base a cierta actividad en el dispositivo. Si el sistema de *logs* es inexistente, el análisis forense tendrá que desarrollarse basándose sólo en datos extraídos de las comunicaciones por la red.
- **Fallos del sistema.** Los patrones de fallo pueden evidenciar el objetivo de algunos ataques a nivel industrial. Por ello, y de la misma manera que en el caso de los *log*s, es importante que los dispositivos industriales tengan la capacidad de almacenar ciertos registros de actividad.
- **Integridad de ficheros.** En este caso, ya no solo en el entorno industrial, sino también en la parte corporativa, existen soluciones que permiten monitorizar la integridad de algunos ficheros de importancia, tanto para el sistema que se está ejecutando dentro del dispositivo, como para soluciones software concretas que se apoyen en archivos concretos.
- **Registros de acceso y auditoría de los mismos.** Este tipo de registros permitirá en un análisis forense realizar un mapa de comunicaciones, detectando el origen y destino de las peticiones de registro. El análisis de estos, puede arrojar información de gran valor para la persona responsable del análisis forense, ya que podrían detectar accesos a horas poco comunes, anomalías tras cruzar datos de los equipos origen que intentan conectarse a un dispositivo, etc.
- **Otras fuentes de datos.** Uso de todo tipo de "*artifacts*"[9][10], es decir, datos que podrían o no ser relevantes en una investigación forense o proceso de respuesta a incidentes para la obtención de información. En este aspecto, cabe destacar la diferencia entre "*artifact*" y evidencia, siendo esta un dato ("*artifact*") que es relevante en una investigación forense o en un proceso de respuesta a incidentes porque apoya o refuta una hipótesis.

[9] https://github.com/ForensicArtifacts/artifacts

[10] https://www.blackhat.com/docs/us-14/materials/us-14-Castle-GRR-Find-All-The-Badness-Collect-All-The-Things-WP.pdf

A continuación, se proporciona una lista de enlaces de consulta que guardan relación con la obtención de "*artifacts*" a nivel forense y que pueden ser de utilidad en entornos industriales:

Artifact	Información proporcionada
Proceso de inicio e inicialización	Información sobre horarios y usuarios específicos del programa; puede utilizarse para determinar la actividad del proceso iniciada por usuarios no autorizados.
Uso de memoria residente	A menudo se realiza solo en tiempo real, el uso de la memoria puede proporcionar información sobre programas maliciosos y otras actividades maliciosas.
Alamas (Intentos no autorizados, accesos a ficheros no autorizados)	Historial de intentos de inicio de sesión, acceso a archivos, cambios de estado. Se puede utilizar en tándem con el análisis de archivos de registro de errores.
Paros y reinicios de sistema	Proporciona información sobre la terminación del proceso, el cierre, la interrupción y quién inició la actividad. A menudo se puede proporcionar actividades asociadas con el acceso del atacante a los archivos de inicio / apagado.
Utilización de procesos y recursos	Proporciona información sobre qué procesos se están ejecutando y los recursos afiliados para ejecutar ese proceso. Puede proporcionar información sobre aplicaciones no autorizadas o vectores de ataque concurrentes.
Actividad de CPU	Proporciona actividad de la CPU. Se puede mapear (usando temporización / reloj) para actividades específicas.
Potencial total del disco y uso de capacidad	La revisión directa puede proporcionar información sobre código malicioso o actividad en sectores específicos del disco. También puede proporcionar información sobre cómo se usó el disco

Tabla 1: Artifacts e información que podrían proporcionar en un análisis forense [Ref.- 12]

El uso de estos mecanismos de seguridad no proporcionará una efectividad completa, pero sí permitirá a la organización tener la información suficiente, en caso de incidente, para poder sacar conclusiones de forma rápida y eficiente en un análisis forense.

Además, el uso de herramientas de monitorización ayudaría al mantenimiento de un inventario de activos actualizado, teniendo un mayor control de los dispositivos implicados en cada proceso y evitando la incorporación de dispositivos maliciosos en la red de la empresa.

5.2.3. Estudio de la amenaza

Una vez detectada e identificada la amenaza, se suelen utilizar diferentes técnicas forenses para poder modelar la misma. El comportamiento que tiene dicha amenaza a lo largo de toda la infraestructura y el alcance de la infección, teniendo en cuenta todos los activos infectados, son dos conceptos de gran valor en este tipo de modelados.

Con el objetivo de realizar un modelado de la amenaza, MITRE ha desarrollado una matriz específica para sistemas de control (ICS ATT&CK)[11] que, como ya se ha comentado previamente en este estudio, recopila las diferentes técnicas, tácticas y conocimientos comunes sobre el adversario. En este punto de la investigación, el uso de esta matriz como apoyo al modelado de la amenaza cobra todo el sentido del mundo. La matriz resulta una herramienta bastante interesante si queremos realizar una compartición de información (*Information Sharing*) con otros investigadores o mostrar de una manera "estandarizada" información de la amenaza.

La Tabla 2 muestra un ejemplo de la matriz MITRE modelando la amenaza del malware Stuxnet.

STUXNET			
ID	Táctica	Técnica	Uso
T1050	Nuevo servicio	Persistencia	Instalación de servicio para cargar un driver
T1109	Componentes del firmware	Persistencia	Modificaciones en firmware de los PLC (controladores Siemens S7-315)
T1116	Firmado de código	Evasión del operador y la defensa	Firma de drivers con claves sobadas.
T1055	Inyección DLL	Evasión del operador y la defensa	Inyección de código utilizando técnicas novedosas
T1093	Suspensión de procesos	Evasión del operador y la defensa	Posibilidad de controlar el flujo de los procesos afectando a la disponibilidad de ciertos servicios.
T1014	Rootkit	Evasión del operador y la defensa	Archivos ocultos en dispositivos extraíbles USB gracias al uso de drivers
T1036	Enmascaramiento	Evasión del operador y la defensa	Drivers del sistema, ~W..tmp, copia de atajos de comando, etc.
T1107	Borrado de ficheros	Evasión del operador y la defensa	El malware se eliminaba del dispositivo USB tras 3 infecciones
T1027	Información ofuscada	Evasión del operador y la defensa	Las DLL utilizadas estaban cifradas y ofuscadas con funciones XOR
T1134	Acceso con *tokens*	Escalada de privilegios	Robo y exploración de *tokens* asociados a usuarios
T1068	Explotación de vulnerabilidades	Escalada de privilegios	Objetivo: usuarios del sistema
T1120	Descubrimiento periférico	Descubrimiento	Detección de drivers para USB (detección de PLC vía cable con S7)
T1012	Consulta de registros	Descubrimiento	Consultas para la detección de WinCC/S7

[11] https://www.acsac.org/2017/workshops/icss/Otis-Alexander-ICS,%20Adversarial%20Tactics,%20Techniques.pdf

T1057	Descubrimiento de procesos	Descubrimiento	--
T1087	Cuentas existentes	Descubrimiento	--
T1063	Antivirus existente	Descubrimiento	--
T1135	Compartición de recursos en red	Descubrimiento	--
T1046	Escaneo de servicios en red	Descubrimiento	--
T1053	Tareas programadas	Ejecución	Copia de ficheros compartidos en red cuando empieza la programación de tareas
T1106	Interacción con API	Ejecución	Carga de DLL vía complejas llamadas a una API
T1047	WMI	Ejecución	--
T1068	Explotación de vulnerabilidades	Movimientos laterales	MS08-067, etc.
T1078	Cuentas válidas	Movimientos laterales	Movimientos laterales usando *tokens* de usuarios y contraseñas embebidas en la base de datos de WinCC
T1091	Replicación a través de medios removibles	Movimientos laterales	--
T1077	Compartición de recursos con administradores Windows	Movimientos laterales	--
T1105	Copia de ficheros remotos	Movimientos laterales	--
T1016	Configuración de red del sistema	Recolección	Transmite información sobre IP al C&C
T1082	Datos del sistema local	Recolección	Transmite información del sistema al C&C
T1043	Puertos de uso común	Comando y Control	Stuxnet usaba tráfico bajo el puerto 80
T1092	Comunicación a través de medios extraíbles	Comando y Control	Podía actualizar su configuración gracias a los medios extraíbles
T024	Protocolo de cifrado propietario/personalizado	Comando y Control	Tráfico ofuscado con funciones XOR
T1132	Codificación de datos	Comando y Control	Dentro del parámetro: *?data=…*

Tabla 2: Ejemplo ICS ATT&CK con Stuxnet [Ref.- 8][Ref.- 9][Ref.- 10][Ref.- 11]

Este tipo de matrices se pueden aplicar en diferentes casos de uso, como *Threat Intelligence*, en operaciones o aplicada a un análisis de ciberseguridad para proporcionar una información ordenada y de consumo rápido[12, 13, 14].

- *Threat Intelligence*:
  - Detección de las técnicas específicas utilizadas por el atacante.
  - Compartición y revelación de información sobre el incidente.
- Operaciones:
  - Priorización de tareas.
  - Detección de anomalías.
  - Simulación de los pasos que ejecutó el atacante.
- Análisis de ciberseguridad:
  - Análisis de las brechas de seguridad en las tecnologías de defensa actuales (previas al incidente y que se mantienen sin modificar).
  - Nuevas tecnologías.
  - Investigación.

[12] https://www.acsac.org/2017/workshops/icss/Otis-Alexander-ICS,%20Adversarial%20Tactics,%20Techniques.pdf

[13] https://www.incibe-cert.es/blog/defensa-activa-e-inteligencia-teoria-practica

[14] https://www.incibe-cert.es/blog/defensa-activa-e-inteligencia-threat-intelligence-los-entornos-industriales

6. Herramientas para realizar un análisis forense en redes OT

Dentro de los entornos industriales es importante adquirir evidencias gracias a la creación de *snapshots*, antes de realizar un análisis con cualquier herramienta. En apartados anteriores ya se hizo referencia a este hecho, por ello, ahora se explicarán una serie de herramientas que permiten crear imágenes para utilizarlas posteriormente en la búsqueda de evidencias.

En las redes TI, la creación de imágenes suele ser bastante fácil, dada la gran variedad de herramientas existentes en el mercado y el hecho de que se encuentran preinstaladas, permitiendo la adquisición de evidencias de forma remota y centralizada. En los entornos industriales, por el contrario, este hecho muchas veces no es tan trivial y, por ello, es necesario acceder directamente al dispositivo en busca de un volcado de memoria u otro tipo de evidencias.

También suele ser bastante común la ausencia de puertos USB o lectores de CD, por lo que las distribuciones "*live*" que se cargan gracias a estos medios, no serán utilizables. Una característica más de los entornos industriales que ha de tenerse en cuenta para la obtención de imágenes de forma remota es que las redes, LAN y WAN, pueden tener algún tipo de restricción o harán uso de dispositivos de seguridad. Este hecho se debe a que las redes industriales suelen ser bastante estáticas con respecto a posibles cambios tanto en sus comunicaciones, como en la incorporación de ciertos dispositivos.

	Herramientas								
	dd	EnCase Forensic Imager	FTK Imager	LiME	Lorg	Memoryze	NST	Redline	SIFT
Plataforma									
Windows	✓	✓	✓		✓	✓		✓	✓
Linux	✓		✓	✓	✓		✓		✓
Apple	✓		✓		✓	✓			
Entradas									
Dispositivo Físico	✓	✓	✓	✓	✓	✓	✓	✓	✓
Volumen lógico	✓	✓	✓		✓	✓	✓	✓	✓
Ficheros	✓	✓			✓		✓	✓	✓
Carpetas	✓	✓	✓				✓	✓	✓
Codificación									
Compresión	✓	✓	✓				✓		
Cifrado	✓	✓	✓				✓		
Formato									
Raw	✓	✓	✓	✓	✓	✓	✓	✓	✓
E01		✓	✓				✓		✓
Salidas									
Ex01		✓	✓				✓		
Split	✓	✓	✓				✓		✓

Tabla 3: Herramientas de extracción de información [Ref.- 12]

6.1. Herramientas de extracción de información

- **dd[15]:** comando que viene por defecto en multitud de distribuciones Linux, como Ubuntu o Fedora. Herramienta versátil que permite la generación de copias de seguridad de información a bajo nivel (raw), así como la trasferencia de archivos específicos y conversión de datos.
- **EnCase Forensic Imager[16]:** plataforma de investigación forense que recolecta datos digitales, realiza análisis, informa sobre descubrimientos y los preserva en un formato válido a efectos legales.

[15] https://es.wikipedia.org/wiki/Dd_(Unix)

[16] https://www.guidancesoftware.com/encase-forensic-imager

- **FTK Imager[17]:** herramienta gratuita diseñada para sistemas Windows que permite la creación de imágenes en varios formatos. Su fácil manejo e instalación hacen de ella una opción a tener en cuenta. Con esta herramienta también se puede hacer un volcado de la memoria RAM, para posteriormente analizarla o generar una imagen de dispositivos de almacenamiento.
- **LiME[18]:** herramienta que permite la adquisición de memoria volátil en Linux o dispositivos basados en este sistema operativo.
- **Lorg[19]**: herramienta orientada al análisis de *logs* que permite comprobar los eventos de seguridad en archivos de registro HTTPD. También puede utilizarse como herramienta forense.
- **Memoryze[20]:** esta herramienta forense de Mandiant, ayuda a los investigadores a detectar malware gracias al análisis de imágenes en vivo.
- **NST - Network Security Toolkit[21]:** distribución Linux utilizada por muchos profesionales que incluye una gran colección de herramientas para el análisis de red.
- **Redline[22]:** herramienta desarrollada por *FireEye* que permite recopilar bastantes datos de una memoria volátil. Con esta herramienta es posible auditar a fondo y recoger todos los procesos en ejecución e información de la memoria RAM, como, por ejemplo, metadatos del sistema de archivos, datos de registros, registros de eventos, información de la red, servicios y, tareas e historial web.
- **Sans Investigative Forensic Toolkit (SIFT)[23]:** distribución basada en Ubuntu diseñada para ayudar a equipos de respuesta a incidentes y a equipos que necesiten realizar análisis forenses. Posee gran variedad de herramientas forenses y guías con comandos que pueden facilitar la vida de los analistas.

6.2. Herramientas de análisis

- **Cuckoo Sandbox[24]:** software de código abierto que permite la automatización de análisis de archivos sospechosos que potencialmente pueden contener malware.
- **Logdissect[25]:** herramienta con interfaz gráfica y API escrita en Python para analizar archivos de registro y otros datos.
- **Volatility[26]**: *framework* avanzado para realizar análisis forense de memorias en sistemas Windows, Mac y Linux. Implementado en Python, este *framework* posee

[17] https://accessdata.com/product-download/ftk-imager-version-3.4.3

[18] https://github.com/504ensicsLabs/LiME

[19] https://github.com/jensvoid/lorg

[20] https://www.fireeye.com/services/freeware/memoryze.html

[21] http://www.networksecuritytoolkit.org/nst/index.html

[22] https://www.fireeye.com/services/freeware/redline.html

[23] https://digital-forensics.sans.org/community/downloads

[24] https://cuckoosandbox.org/

[25] https://pypi.org/project/logdissect/1.2/

[26] https://www.volatilityfoundation.org/

diferentes herramientas que permiten realizar diversas tareas para obtener información como:

- Procesos en ejecución.
- Llamadas al sistema en tablas para un mayor entendimiento.
- Análisis en busca de patrones anómalos (bytes, expresiones regulares, cadenas de memoria, etc.).
- Volcado de hashes LM/NTLM, procesos, DLL o módulos del disco.
- Módulo del *kernel* utilizado.
- Mapeo físico a las direcciones virtuales
- Etc.

Algunos comandos interesantes de *Volatility* son:

- **imageinfo:** muestra resumen de alto nivel sobre el sistema al que se le ha realizado un volcado de la memoria.
- **pslit:** identifica los procesos del sistema. Es necesario indicar en la ejecución del comando, que tipo de sistema se está analizando (previamente utilizar imageinfo).
- **maldfind:** ayuda a encontrar códigos DLLs ocultos o inyectados en la memoria del modo de usuario (normalmente maliciosos) y mostrar la información por pantalla.
- **voldiff:** este plugin permite a los investigadores realizar una comparación entre imágenes de las memorias. Ayudando a identificar IOCs y entender el comportamiento avanzado del malware.

- **YARA[27]:** herramienta que permite la detección de malware gracias a una serie de reglas que consisten en un conjunto de cadenas y expresiones booleanas que determinan su lógica. Gracias a esta herramienta, es posible clasificar gran parte de las muestras de malware actuales o buscar comportamientos similares entre ellos.

6.3. Otras herramientas

- **GRR[28]:** GRR Rapid Response es un *framework* de respuesta a incidentes centrado en sistemas forenses "*live*". Basado en el uso de agentes para la obtención de información en los hosts. Esta solución es capaz de centralizar todos los datos recopilados en un servidor remoto y mostrar la información ordenada.
- **Timesketch[29]:** herramienta de código abierto que permite realizar una cronología en un análisis forense y, si se desea, utilizarse de forma conjunta y colaborativa entre más de un analista por medio de un control de usuarios.

[27] https://yara.readthedocs.io/en/v3.5.0/index.html

[28] https://github.com/google/grr

[29] https://github.com/google/timesketch

7. Análisis forense de un malware específico de SCI

Para finalizar este estudio, se presenta un caso práctico de análisis forense del primer malware que afectó a los Sistemas de Control Industrial y supuso el punto de inflexión en la seguridad de estos sistemas: Stuxnet. Para ello, se utilizarán algunas de las herramientas mencionadas en apartado 6.-Herramientas para realizar un análisis forense en redes OT. El objetivo es llevar a la práctica todos los conceptos teóricos comentados, siguiendo el orden de los subapartados desarrollados en el punto 5.2.-Proceso de análisis forense.

7.1. Identificación de activos

La recopilación de evidencias marca el inicio de la investigación forense, en la cual se busca proporcionar información y detectar qué ha pasado exactamente.

A pesar de que hay otros elementos con mayor volatilidad como, por ejemplo, la memoria caché, algunos registros de sistema y determinada información de red (tabla de rutas ARP). En este caso, analizaremos la memoria, de donde podremos obtener información de los procesos, de la red y del sistema.

La muestra pertenece a un entorno de laboratorio y ha sido proporcionada por la empresa que ha sufrido el ataque, que ha logrado mantener el funcionamiento de sus procesos, desinfectar todos los servidores afectados y aislar el malware. De esta forma, ha facilitado las labores de extracción y solo será necesario centrarse en las labores de análisis.

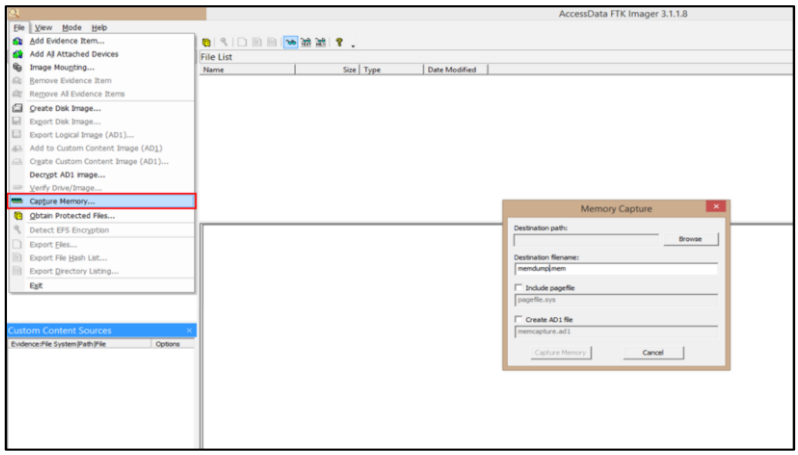

Figura 7.- Volcado de memoria con la herramienta FTK Imager

7.2. Detección de la anomalía

Para empezar con el análisis, se utilizarán una serie de reglas Yara[30] existentes[31] con el objetivo de descartar o detectar una correspondencia con algún tipo de malware del que ya se posee información en Sistemas de Control Industrial.

```
root@remnux:/home/remnux/Desktop# yara -s stuxnet.yara /home/remnux/Desktop/memdump.vmem
StuxNet_Malware_1 /home/remnux/Desktop/memdump.vmem
0x4b54723:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x4bd8bd5:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x4c10723:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x4f3fbd5:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x5198e3d:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x6b10723:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x6ba8b63:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x7ec5071:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x87c4723:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x8801723:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x11c83bd5:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x11f71071:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x139e3071:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x164d7538:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x17fad938:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
0x4b54eb5:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x4c10eb5:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x4c32659:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x4ee72f5:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x4fb6863:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x6a7e659:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x6b10eb5:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x6e45370:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x7ec59c5:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x87a1370:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x87c4eb5:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x8801eb5:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x11f719c5:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x139e39c5:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x164d7e05:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x18819791:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x18b6e205:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x1a8d7659:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x1c214370:$op2: 74 36 8B 7F 08 83 FF 00 74 2E 0F B7 1F 8B 7F 04
0x4b54f3f:$op3: 74 70 81 78 05 8D 54 24 04 75 1B 81 78 08 04 CD
0x4c10f3f:$op3: 74 70 81 78 05 8D 54 24 04 75 1B 81 78 08 04 CD
0x4c326e3:$op3: 74 70 81 78 05 8D 54 24 04 75 1B 81 78 08 04 CD
0x4ee737f:$op3: 74 70 81 78 05 8D 54 24 04 75 1B 81 78 08 04 CD
0x4fb68ed:$op3: 74 70 81 78 05 8D 54 24 04 75 1B 81 78 08 04 CD
0x6a7e6e3:$op3: 74 70 81 78 05 8D 54 24 04 75 1B 81 78 08 04 CD
0x6b10f3f:$op3: 74 70 81 78 05 8D 54 24 04 75 1B 81 78 08 04 CD
0x6e453fa:$op3: 74 70 81 78 05 8D 54 24 04 75 1B 81 78 08 04 CD
```

Figura 8.- Uso de reglas Yara para la detección de strings relacionados con el malware Stuxnet

Tras ver el resultado Figura 8, se puede apreciar cómo varios indicadores de compromiso[32], confirman las sospechas de que el sistema analizado, contenía malware. En este caso, se trata de Stuxnet, considerado como la primera "ciberarma" utilizada contra sistemas industriales.

[30] https://www.incibe-cert.es/blog/el-valor-los-indicadores-compromiso-industria

[31] https://github.com/Yara-Rules/rules/blob/master/malware/APT_Stuxnet.yar

[32] https://www.incibe-cert.es/blog/el-valor-los-indicadores-compromiso-industria

Dentro del fichero APT_Stuxnet.yar[33], se encuentran las reglas (utilizadas anteriormente) que permiten la detección de cadenas en base a los Indicadores de Compromiso extraídos con la inteligencia generada tras el análisis, en su momento, de Stuxnet. Por ejemplo, la Regla 1 (StuxNet_Malware_1), dentro del set de reglas disponible en el fichero, detecta unas instrucciones en lenguaje ensamblador propias de Stuxnet.

```
rule StuxNet_Malware_1
{

    meta:
        description = "Stuxnet Sample - file malware.exe"
        author = "Florian Roth"
        reference = "Internal Research"
        date = "2016-07-09"
        hash1 = "9c891edb5da763398969b6aaa86a5d46971bd28a455b20c2067cb512c9f9a0f8"

    strings:
         // 0x10001778 8b 45 08  mov     eax, dword ptr [ebp + 8]
         // 0x1000177b 35 dd 79 19 ae    xor     eax, 0xae1979dd
         // 0x10001780 33 c9     xor     ecx, ecx
         // 0x10001782 8b 55 08  mov     edx, dword ptr [ebp + 8]
         // 0x10001785 89 02     mov     dword ptr [edx], eax
         // 0x10001787 89 ?? ??  mov     dword ptr [edx + 4], ecx
         $op1 = { 8b 45 08 35 dd 79 19 ae 33 c9 8b 55 08 89 02 89 }
```

Figura 9.- Estructura de la regla y búsqueda de la cadena de caracteres concreta ($op1)

```
StuxNet_Malware_1 /home/remnux/Desktop/memdump.vmem
0x4b54723:$op1: 8B 45 08 35 DD 79 19 AE 33 C9 8B 55 08 89 02 89
```

Figura 10.- Salida por pantalla de la detección (Regla 1)

Otro ejemplo claro puede verse en la regla 3 (StuxNet_Malware_3) de la Figura 11. La cadena buscada se corresponde con una de las DLL cargada como biblioteca dinámica en la inyección de procesos que realiza el malware Stuxnet.

[33] https://github.com/Yara-Rules/rules/blob/master/malware/APT_Stuxnet.yar

```
rule Stuxnet_Malware_3
{

    meta:
        description = "Stuxnet Sample - file ~WTR4141.tmp"
        author = "Florian Roth"
        reference = "Internal Research"
        date = "2016-07-09"
        hash1 = "6bcf88251c876ef00b2f32cf97456a3e306c2a263d487b0a50216c6e3cc07c6a"
        hash2 = "70f8789b03e38d07584f57581363afa848dd5c3a197f2483c6dfa4f3e7f78b9b"

    strings:
        $x1 = "SHELL32.DLL.ASLR." fullword wide
```

Figura 11.- Estructura de la regla y búsqueda de la cadena de caracteres concreta ($x1)

```
Stuxnet_Malware_3 /home/remnux/Desktop/memdump.vmem
0x6b963f8:$x1: S\x00H\x00E\x00L\x00L\x003\x002\x00.\x00D\x00L\x00L\x00.\x00A\x00S\x00L\x00R\x00.\x00
```

Figura 12.- Salida por pantalla de la detección (Regla 3)

7.3. Estudio de la amenaza

Una vez detectado el malware, se revisarán las características principales del mismo y se extraerán una serie de evidencias "artifacts" para poder finalmente confirmar, sin duda alguna, que se trata de Stuxnet, y que además no ha sufrido ningún tipo de variación (comportamiento del malware) o modificación (implementación del malware) por parte de los posibles atacantes.40

Con el objetivo de ejecutar ciertas acciones en el fichero proporcionado por la empresa afectada que verifiquen la infección por parte del malware Stuxnet, el siguiente paso consiste en utilizar la herramienta *Volatility*, que permitirá la ejecución de algunas acciones y tratamientos en el fichero proporcionado (.vmem). Se va a utilizar *Volatility* por tratarse de un *framework* con diferentes herramientas que permitirá a los analistas ahorrar tiempo, ya que dispone de diferentes opciones a la hora de llevar a cabo un análisis forense.

Se empezará ejecutando el comando ***imageinfo*** para obtener información del sistema al que corresponde la memoria analizada.

```
root@remnux:/home/remnux/Desktop# vol.py imageinfo -f memdump.vmem
Volatility Foundation Volatility Framework 2.4
Determining profile based on KDBG search...

          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/remnux/Desktop/memdump.vmem)
                      PAE type : PAE
                           DTB : 0x319000L
                          KDBG : 0x80545ae0
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000
             KUSER_SHARED_DATA : 0xffdf0000
           Image date and time : 2018-06-08 04:31:36 UTC+0000
     Image local date and time : 2018-06-08 00:31:36 -0400
```

Figura 13.- Información del sistema analizado a través del comando imageinfo

Gracias a este comando, sabremos los procesos que se estaban ejecutando, así como también el tipo de sistema utilizado, en este caso Windows XP, que podremos utilizar como argumento del comando **pslist**, como se muestra a continuación.

```
root@remnux:/home/remnux/Desktop# vol.py pslist --profile=WinXPSP2x86 -f memdump.vmem
Volatility Foundation Volatility Framework 2.4
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x823c8830 System                    4      0     59      403 ------      0
0x820df020 smss.exe                376      4      3       19 ------      0 2018-10-29 17:08:53 UTC+0000
0x821a2da0 csrss.exe               600    376     11      395      0      0 2018-10-29 17:08:54 UTC+0000
0x81da5650 winlogon.exe            624    376     19      570      0      0 2018-10-29 17:08:54 UTC+0000
0x82073020 services.exe            668    624     21      431      0      0 2018-10-29 17:08:54 UTC+0000
0x81e70020 lsass.exe               680    624     19      342      0      0 2018-10-29 17:08:54 UTC+0000
0x823315d8 vmacthlp.exe            844    668      1       25      0      0 2018-10-29 17:08:55 UTC+0000
0x81db8da0 svchost.exe             856    668     17      193      0      0 2018-10-29 17:08:55 UTC+0000
0x81e61da0 svchost.exe             940    668     13      312      0      0 2018-10-29 17:08:55 UTC+0000
0x822843e8 svchost.exe            1032    668     61     1169      0      0 2018-10-29 17:08:55 UTC+0000
0x81e18b28 svchost.exe            1080    668      5       80      0      0 2018-10-29 17:08:55 UTC+0000
0x81ff7020 svchost.exe            1200    668     14      197      0      0 2018-10-29 17:08:55 UTC+0000
0x81fee8b0 spoolsv.exe            1412    668     10      118      0      0 2018-10-29 17:08:56 UTC+0000
0x81e0eda0 jqs.exe                1580    668      5      148      0      0 2018-10-29 17:09:05 UTC+0000
0x81fe52d0 vmtoolsd.exe           1664    668      5      284      0      0 2018-10-29 17:09:05 UTC+0000
0x821a0568 VMUpgradeHelper        1816    668      3       96      0      0 2018-10-29 17:09:08 UTC+0000
0x8205ada0 alg.exe                 188    668      6      107      0      0 2018-10-29 17:09:09 UTC+0000
0x820ec7e8 explorer.exe           1196   1728     16      582      0      0 2018-10-29 17:11:49 UTC+0000
0x820ecc10 wscntfy.exe            2040   1032      1       28      0      0 2018-10-29 17:11:49 UTC+0000
0x81e86978 TSVNCache.exe           324   1196      7       54      0      0 2018-10-29 17:11:49 UTC+0000
0x81fc5da0 VMwareTray.exe         1912   1196      1       50      0      0 2018-10-29 17:11:50 UTC+0000
0x81e6b660 VMwareUser.exe         1356   1196      9      251      0      0 2018-10-29 17:11:50 UTC+0000
0x8210d478 jusched.exe            1712   1196      1       26      0      0 2018-10-29 17:11:50 UTC+0000
0x82279998 imapi.exe               756    668      4      116      0      0 2018-10-29 17:11:54 UTC+0000
0x822b9a10 wuauclt.exe             976   1032      3      133      0      0 2018-10-29 17:12:03 UTC+0000
0x81c543a0 Procmon.exe             660   1196     13      189      0      0 2019-06-03 04:25:56 UTC+0000
0x81fa5390 wmiprvse.exe           1872    856      5      134      0      0 2019-06-03 04:25:58 UTC+0000
0x81c498c8 lsass.exe               868    668      2       23      0      0 2019-06-03 04:26:55 UTC+0000
0x81c47c00 lsass.exe              1928    668      4       65      0      0 2019-06-03 04:26:55 UTC+0000
0x81c0cda0 cmd.exe                 968   1664      0 --------      0      0 2019-06-03 04:31:35 UTC+0000   2019-06-03 04:31:36 UTC+0000
0x81f14938 ipconfig.exe            304    968      0 --------      0      0 2019-06-03 04:31:35 UTC+0000   2019-06-03 04:31:36 UTC+0000
```

Figura 14.- Lista de procesos obtenida a través del comando pslist

En la salida del comando pueden observarse varios procesos relacionados con el ejecutable *lsass.exe*. El proceso del sistema creado por *lsass.exe* gestiona la autenticación de usuarios a nivel local mediante un paquete especificado en *HKLM\SYSTEM\CurrentControlSet\Control\Lsa*. Generalmente, esta autenticación se realiza mediante Kerberos[34] para cuentas de dominio o MSV1_0 en cuentas locales.

Además de permitir la autenticación de usuarios, *lsass.exe* también es responsable de implementar la política de seguridad local (políticas de contraseñas y auditoría) y de escribir eventos en el registro de eventos de seguridad. Trataremos estos procesos como evidencia, ya que *lsass.exe* suele ejecutarse en una única instancia sin tener procesos secundarios. Para filtrar de forma clara las instancias asociadas al ejecutable utilizaremos el comando ***getsids*** donde se realizará un filtro por PID:

```
root@remnux:/home/remnux/Desktop# vol.py getsids -p 680,868,1928 -f memdump.vmem
Volatility Foundation Volatility Framework 2.4
lsass.exe (680): S-1-5-18 (Local System)
lsass.exe (680): S-1-5-32-544 (Administrators)
lsass.exe (680): S-1-1-0 (Everyone)
lsass.exe (680): S-1-5-11 (Authenticated Users)
lsass.exe (868): S-1-5-18 (Local System)
lsass.exe (868): S-1-5-32-544 (Administrators)
lsass.exe (868): S-1-1-0 (Everyone)
lsass.exe (868): S-1-5-11 (Authenticated Users)
lsass.exe (1928): S-1-5-18 (Local System)
lsass.exe (1928): S-1-5-32-544 (Administrators)
lsass.exe (1928): S-1-1-0 (Everyone)
lsass.exe (1928): S-1-5-11 (Authenticated Users)
```

Figura 15.- Opción 1 para obtener las instancias asociadas al ejecutable (lsass.exe)

[34] https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-kerberos.html

Como alternativa también podría utilizarse ***memdump***:

```
root@remnux:/home/remnux/Desktop# vol.py -f memdump.vmem pstree | egrep '(lsass.exe)'
Volatility Foundation Volatility Framework 2.4
.... 0x81c47c00:lsass.exe                      1928    668      4       65 2019-06-03 04:26:55 UTC+0000
.... 0x81c498c8:lsass.exe                       868    668      2       23 2019-06-03 04:26:55 UTC+0000
... 0x81e70020:lsass.exe                        680    624     19      342 2018-10-29 17:08:54 UTC+0000
```

Figura 16.- Opción 2 para obtener las instancias asociadas al ejecutable (lsass.exe)

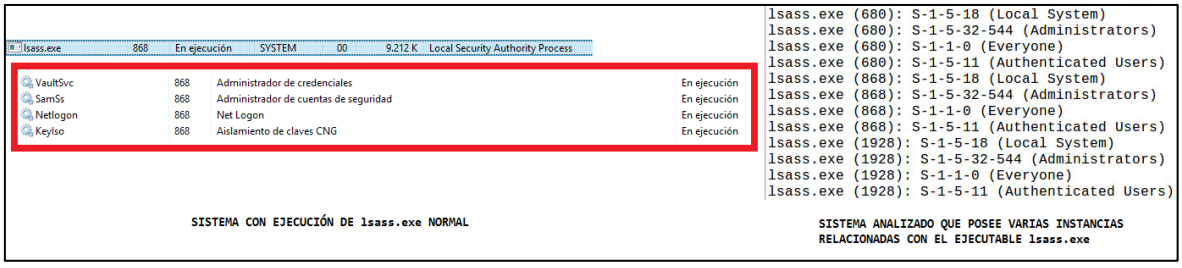

Figura 17.- Sistema sin infección vs. sistema infectado (lsass.exe)

Dado que una de las características ya detectadas de Stuxnet era la inyección DLL sobre los procesos *lsass*, *winlogon* y *svchost*, el siguiente paso es verificar que no ha afectado a otros procesos (el malware sigue el mismo patrón de comportamiento esperado). Con este objetivo, se utilizará el comando ***malfind***. Este comando permite la detección de inyecciones DLL o código oculto dentro del fichero de memoria analizado.

Gracias a la opción *–D* se guardarán las secciones de memoria sospechosas en el directorio que se indique. La opción *--plugins* permite el uso de directorios adicionales de plugins.

```
root@remnux:/home/remnux/Desktop# vol.py --plugins=PluginDirectory malfind -f memdump.vmem --profile=WinXPSP3x86 -D /home/remnux/Desktop/
Volatility Foundation Volatility Framework 2.4
Process: lsass.exe Pid: 1928 Address: 0x6f0000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: Protection: 6

0x006f0000  29 87 7f ae 00 00 00 00 ff ff ff ff 77 35 00 01   ).........w5..
0x006f0010  4b 00 45 00 52 00 4e 00 45 00 4c 00 33 00 32 00   K.E.R.N.E.L.3.2.
0x006f0020  2e 00 44 00 4c 00 4c 00 2e 00 41 00 53 00 4c 00   ..D.L.L...A.S.L.
0x006f0030  52 00 2e 00 30 00 33 00 36 00 30 00 62 00 37 00   R...0.3.6.0.b.7.

0x6f0000 29877fae0000     SUB [EDI+0xae7f], EAX
0x6f0006 0000             ADD [EAX], AL
0x6f0008 ff               DB 0xff
0x6f0009 ff               DB 0xff
0x6f000a ff               DB 0xff
0x6f000b ff7735           PUSH DWORD [EDI+0x35]
0x6f000e 0001             ADD [ECX], AL
0x6f0010 4b               DEC EBX
0x6f0011 004500           ADD [EBP+0x0], AL
0x6f0014 52               PUSH EDX
0x6f0015 004e00           ADD [ESI+0x0], CL
0x6f0018 45               INC EBP
0x6f0019 004c0033         ADD [EAX+EAX+0x33], CL
0x6f001d 0032             ADD [EDX], DH
0x6f001f 002e             ADD [ESI], CH
0x6f0021 0044004c         ADD [EAX+EAX+0x4c], AL
0x6f0025 004c002e         ADD [EAX+EAX+0x2e], CL
0x6f0029 004100           ADD [ECX+0x0], AL
0x6f002c 53               PUSH EBX
0x6f002d 004c0052         ADD [EAX+EAX+0x52], CL
0x6f0031 002e             ADD [ESI], CH
0x6f0033 0030             ADD [EAX], DH
0x6f0035 0033             ADD [EBX], DH
0x6f0037 0036             ADD [ESI], DH
0x6f0039 0030             ADD [EAX], DH
0x6f003b 006200           ADD [EDX+0x0], AH
0x6f003e 37               AAA
0x6f003f 00               DB 0x0

Process: lsass.exe Pid: 1928 Address: 0x680000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: Protection: 6
```

Figura 18.- Detección de inyección en proceso lsass.exe

```
Process: svchost.exe Pid: 940 Address: 0xbf0000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: Protection: 6

0x00bf0000  90 06 bf 00 c6 07 bf 00 24 00 bf 00 a5 04 00 00   ........$.......
0x00bf0010  f2 04 bf 00 48 06 00 00 c9 04 bf 00 29 00 00 00   ....H.........)...
0x00bf0020  00 00 b7 00 e8 13 00 00 00 5a 77 4d 61 70 56 69   .........ZwMapVi
0x00bf0030  65 77 4f 66 53 65 63 74 69 6f 6e 00 5a 51 81 c1   ewOfSection.ZQ..

0xbf0000 90              NOP
0xbf0001 06              PUSH ES
0xbf0002 bf00c607bf      MOV EDI, 0xbf07c600
0xbf0007 002400          ADD [EAX+EAX], AH
0xbf000a bf00a50400      MOV EDI, 0x4a500
0xbf000f 00f2            ADD DL, DH
0xbf0011 04bf            ADD AL, 0xbf
0xbf0013 004806          ADD [EAX+0x6], CL
0xbf0016 0000            ADD [EAX], AL
0xbf0018 c9              LEAVE
```

Figura 19.- Detección de inyección en proceso svchost.exe

Para buscar más evidencias sobre las DLL se utilizará el comando ***dlllist***, que nos proporcionará una lista de las DLL detectadas en el fichero de memoria analizado y, además, el plugin *ldrmodules*, que nos permitirá ver los tres casos de inyección de código utilizados por Stuxnet.

```
root@remnux:/home/remnux/Desktop# vol.py dlllist -f memdump.vmem --profile=WinXPSP3x86 | grep ASLR
Volatility Foundation Volatility Framework 2.4
0x013f0000    0x138000         0x1 C:\WINDOWS\system32\KERNEL32.DLL.ASLR.0360c5e2
0x00d00000    0x138000         0x1 C:\WINDOWS\system32\KERNEL32.DLL.ASLR.0360c8ee
0x00870000    0x138000         0x1 C:\WINDOWS\system32\KERNEL32.DLL.ASLR.0360b7ab
```

Figura 20.- Inyecciones DLL detectadas

Cada vez que Stuxnet necesitaba cargar una DLL, utilizaba un método especialmente diseñado para eludir el bloqueo y las tecnologías basadas en HIDS que monitorizaban las llamadas de LoadLibrary. Stuxnet llamaba a LoadLibrary con un nombre de archivo especialmente diseñado que no existe en el disco y que normalmente provoca que LoadLibrary falle. Con estos fallos, el malware conseguía que se realizasen llamadas a direcciones de memoria donde se ubicaban los ficheros maliciosos.

```
root@remnux:/home/remnux/Desktop# vol.py --plugins=PluginDirectory malfind -f memdump.vmem --profile=WinXPSP3x86 -D /home/remnux/Desktop/
Volatility Foundation Volatility Framework 2.4
Pid      Process            Base       InLoad InInit InMem MappedPath
---------- ---------- ---------- ---------- ---------- ---------- ----------
   1928 lsass.exe          0x00080000 False  False  False
   1928 lsass.exe          0x773d0000 True   True   True   \WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-ww_35d4ce83\comctl32.dll
   1928 lsass.exe          0x77f60000 True   True   True   \WINDOWS\system32\shlwapi.dll
   1928 lsass.exe          0x771b0000 True   True   True   \WINDOWS\system32\wininet.dll
   1928 lsass.exe          0x77a80000 True   True   True   \WINDOWS\system32\crypt32.dll
   1928 lsass.exe          0x77fe0000 True   True   True   \WINDOWS\system32\secur32.dll
   1928 lsass.exe          0x77c00000 True   True   True   \WINDOWS\system32\version.dll
   1928 lsass.exe          0x01000000 True   False  True
   1928 lsass.exe          0x59000000 True   True   True   \WINDOWS\system32\netapi32.dll
   1928 lsass.exe          0x77e70000 True   True   True   \WINDOWS\system32\rpcrt4.dll
   1928 lsass.exe          0x71ab0000 True   True   True   \WINDOWS\system32\ws2_32.dll
   1928 lsass.exe          0x71ad0000 True   True   True   \WINDOWS\system32\wsock32.dll
   1928 lsass.exe          0x774e0000 True   True   True   \WINDOWS\system32\ole32.dll
   1928 lsass.exe          0x7e410000 True   True   True   \WINDOWS\system32\user32.dll
   1928 lsass.exe          0x77f10000 True   True   True   \WINDOWS\system32\gdi32.dll
   1928 lsass.exe          0x77120000 True   True   True   \WINDOWS\system32\oleaut32.dll
   1928 lsass.exe          0x76d60000 True   True   True   \WINDOWS\system32\iphlpapi.dll
   1928 lsass.exe          0x769c0000 True   True   True   \WINDOWS\system32\userenv.dll
   1928 lsass.exe          0x7c800000 True   True   True   \WINDOWS\system32\kernel32.dll
   1928 lsass.exe          0x76bf0000 True   True   True   \WINDOWS\system32\psapi.dll
   1928 lsass.exe          0x77c10000 True   True   True   \WINDOWS\system32\msvcrt.dll
   1928 lsass.exe          0x77dd0000 True   True   True   \WINDOWS\system32\advapi32.dll
   1928 lsass.exe          0x7c9c0000 True   True   True   \WINDOWS\system32\shell32.dll
   1928 lsass.exe          0x00870000 True   True   True
   1928 lsass.exe          0x76f20000 True   True   True   \WINDOWS\system32\dnsapi.dll
   1928 lsass.exe          0x5d090000 True   True   True   \WINDOWS\system32\comctl32.dll
   1928 lsass.exe          0x71aa0000 True   True   True   \WINDOWS\system32\ws2help.dll
   1928 lsass.exe          0x77b20000 True   True   True   \WINDOWS\system32\msasn1.dll
```

Figura 21.- Nombres de ruta en blanco, PE en 0x80000 (ya se sabe inyectado) y falta de carga con llamadas LoadLibrary

Todas estas evidencias corroboran que el código ha sido inyectado a través de funciones *VirtualAlloc*[35], *VirtualAllocEx*[36] o *WriteProcessMemory*[37].

Estas tres funciones juegan un papel importante en la gestión de la memoria dentro de los sistemas operativos Windows, ya que permiten realizar cambios, reservar espacio, escribir, etc., en áreas de memoria ocupados por ciertos procesos. En este caso, parece que las escrituras y gestiones de la memoria se han salido de las regiones previamente delimitadas, de ahí que exista una inyección. Este hecho, se ha visto reflejado en imágenes como la Figura 16 o la Figura 17 donde se han detectado varias instancias asociadas al mismo ejecutable.

Para finalizar la investigación, se realizará un análisis de las conexiones de red detectadas en busca de comunicaciones con los C&C. Gracias al comando ***connscan***, el *framework* de *Volatility* nos permite encontrar las direcciones IP locales y remotas pertenecientes a los procesos generados por las conexiones detectadas en el fichero de memoria analizado.

```
root@remnux:/home/remnux/Desktop# vol.py connscan -f memdump.vmem
Volatile Systems Volatility Framework 2.4
Offset     Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ------
0x01da9e68 192.168.16.129:1311       128.61.111.9:51442          1280
0x01e4fe68 192.168.16.129:1233       128.61.111.9:21             1280
0x01eeebf0 172.16.237.145:1170       72.167.202.5:80              528
0x020bf4e0 172.16.237.145:1045       96.17.106.99:80             1648
0x0242ec28 172.16.237.145:1048       96.17.106.99:80             1708
0x025069e8 172.16.237.145:1090       137.254.16.78:80             152
```

Figura 22.- Comunicaciones detectadas a través del comando connscan

Existen diferentes IP externas a las que se realizaban comunicaciones de navegación web. Tras revisar dichas direcciones, comprobamos que al menos una de ellas pertenecía a las IP detectadas como C&C en la campaña de Stuxnet.

TYPE	INDICATOR	TITLE	ACTIVE	RELATED PULSES
domain	units.cw			0
domain	variant.you			0
hostname	www.mypremierfutbol.com			1
hostname	www.todaysfutbol.com			2

Figura 23.- Hostnames relacionados con los C&C de Stuxnet[38]

Como se ha podido observar, es importante realizar la recopilación de evidencias y seguir ciertos pasos para ahorrar tiempo de análisis forense en los entornos industriales.

[35] https://msdn.microsoft.com/en-us/library/windows/desktop/aa366887(v=vs.85).aspx

[36] https://msdn.microsoft.com/en-us/library/windows/desktop/aa366890(v=vs.85).aspx

[37] https://msdn.microsoft.com/es-es/library/windows/desktop/ms681674(v=vs.85).aspx

[38] https://otx.alienvault.com/pulse/57e8a5f53f5d465dafc74968/

En este caso, se optó por mantener una investigación que requería de alguna tarea manual para verificar una hipótesis planteada al inicio de la misma, que el malware era "Stuxnet". Gracias a las evidencias recopiladas, se ha podido verificar que la hipótesis inicial era correcta.

El último paso a realizar, en el caso de detectar alguna anomalía no detectada previamente en otras investigaciones, es el de crear los diferentes IoC asociados. Estos Indicadores de Compromiso pueden ser: registros detectados, *hostnames*, archivos, procesos, etc.

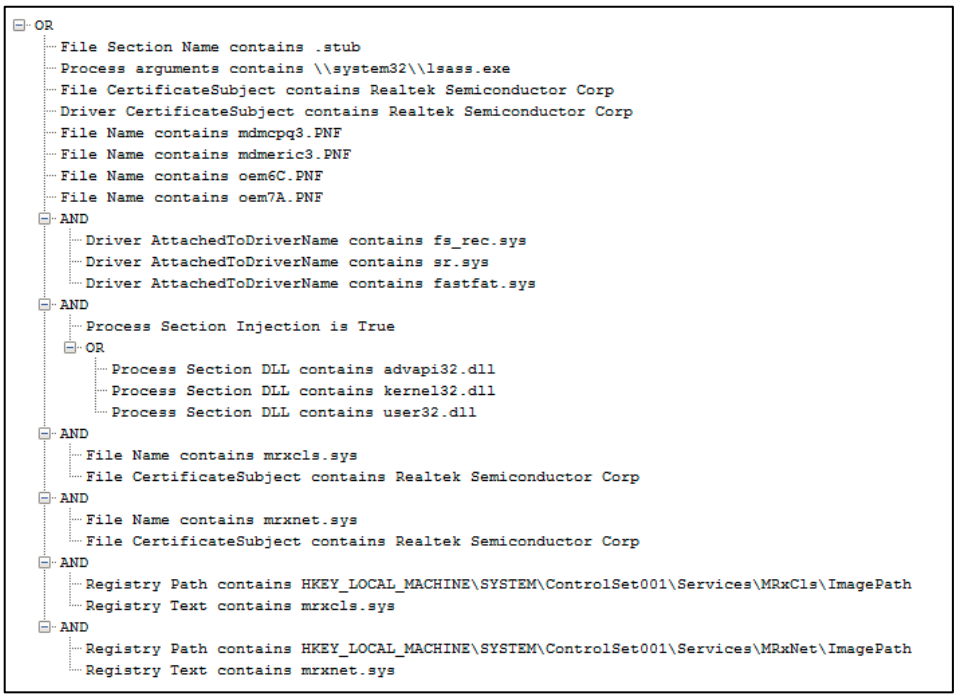

Figura 24.- Ejemplo de Indicadores de Compromiso para Stuxnet

Para finalizar, se ofrece un listado, a modo de resumen, que recoge cada uno de los puntos obtenidos fruto de la investigación:

- Procesos en ejecución maliciosos.
- Evidencias de inyecciones DLL.
- Funciones a través de las cuales se ha podido inyectar el código.
- Comunicaciones con los C&C.
- Dominios maliciosos que aparecieron en listas negras.
- Verificación con reglas YARA del malware analizado (Stuxnet).
- Posterior creación y uso de IoC para localizar infecciones en otros dispositivos de la empresa afectada.

8. Conclusiones

Las técnicas de análisis forense se están aplicando ya sobre los Sistemas de Control Industrial para investigar la aparición de malware. Como se ha mostrado en este estudio, dichos análisis requieren de conocimientos que, en ocasiones, están íntimamente ligados a dispositivos industriales.

Gracias a herramientas de apoyo, como las que se han presentado en el documento, utilizadas de manera correcta, el análisis puede dar buenos resultados y permitir identificar todas las características de un malware causante de un incidente industrial. Como se ha mostrado en el caso práctico, la elección de las herramientas y la utilización de las técnicas adecuadas permiten identificar un malware concreto.

Este tipo de actuaciones, que ya se han utilizado durante los últimos años sobre los entornos corporativos, actualmente se están poniendo en valor en los entornos industriales, pero aún es manifiesta la carencia de perfiles especializados en este campo. Por ello, este estudio pretende proporcionar los conocimientos básicos que sirvan de punto de partida, para despertar la curiosidad y el interés, ofreciendo unas nociones básicas de cómo abordar ciertos problemas.

9. Glosario de términos y acrónimos

ARP: Address Resolution Protocol

DCS: Distributed Control System

HIDS: Host-based Intrusion Detection Systems

HMI: Human Machine Interface

IDS: Intrusion Detection System

LAN: Local Area Network

NTP: Network Time Protocol

OT: Operation Technology

PLC: Programmable Logic Controller

PTP: Precision Time Protocol

RAM: Random Access Memory

RTU: Remote Terminal Unit

SCADA: Supervisory Control And Data Acquisition

SIEM: Security Information and Event Management

SNTP: Simple Network Time Protocol

SOC: Security Operations Center

TTP: Técnicas, Tácticas y Procedimientos

WAN: Wide Area Network

10. Referencias

Referencia	Título, autor y enlace web
[Ref.- 1]	Guía para la Construcción de un Centro de Operaciones y Respuesta de Ciberseguridad Industrial. https://www.cci-es.org/web/cci/detalle-actividad/-/journal_content/56/10694/753289
[Ref.- 2]	Development of a tailored methodology and forensic toolkit for industrial control systems incident response. https://calhoun.nps.edu/handle/10945/42595
[Ref.- 3]	Stuxnet Malware Analysis Paper. https://scadahacker.com/files/duqu/stuxnet-malware-analysis-paper.pdf
[Ref.- 4]	Mitre ATT&CK. https://attack.mitre.org/
[Ref.- 5]	The Industrial Control System Cyber Kill Chain. https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297
[Ref.- 6]	Buenas prácticas en el análisis forense de sistemas de automatización y control industrial. https://www.cci-es.org/documents/10694/267017/4.+Buenas+Pr%C3%A1cticas+An%C3%A1lisis+Forense+ICS_CCI.pdf/44c9ecb3-2740-4880-848d-5151a384ff9d;jsessionid=490EEB78806815DE732C728E6B6087BF?version=1.0
[Ref.- 7]	Forensic Plans for Control Systems. https://www.sans.org/reading-room/whitepapers/forensics/forensic-analysis-industrial-control-systems-36277
[Ref.- 8]	Recommended Practice: Creating Cyber Forensics Plans for Control Systems. https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/Forensics_RP.pdf
[Ref.- 9]	Stuxnet Under the Microscope. https://www.esetnod32.ru/company/viruslab/analytics/doc/Stuxnet_Under_the_Microscope.pdf
[Ref.- 10]	ICS ATT&CK Stuxnet http://hugoideler.com/wp-content/uploads/2017/10/ATTCK-Stuxnet.pdf
[Ref.- 11]	Forensic Images: For Your Viewing Pleasure. https://www.sans.org/reading-room/whitepapers/forensics/forensic-images-viewing-pleasure-35447
[Ref.- 12]	The Industrial Control System Cyber Kill Chain. https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297
[Ref.- 13]	Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. https://nostarch.com/malware
[Ref.- 14]	The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory. http://shop.oreilly.com/product/9781118825099.do
[Ref.- 15]	Malware Forensics: Investigating and Analyzing Malicious Code. https://www.researchgate.net/publication/291140179_Malware_Forensics_Investigating_and_Analyzing_Malicious_Code
[Ref.- 16]	Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems. https://www.elsevier.com/books/industrial-network-security/knapp/978-0-12-420114-9
[Ref.- 17]	Hacking Exposed Industrial Control Systems: ICS and SCADA Security Secrets & Solutions. https://www.mhprofessional.com/9781259589713-usa-hacking-exposed-industrial-control-systems-ics-and-scada-security-secrets-solutions-group
[Ref.- 18]	Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS. https://www.researchgate.net/publication/327527699_Cybersecurity_for_industrial_control_systems_SCADA_DCS_PLC_HMI_and_SIS

Referencia	Título, autor y enlace web
[Ref.- 19]	Anatomy of an Attack: Detecting and Defeating CRASHOVERRIDE. https://www.virusbulletin.com/conference/vb2018/abstracts/anatomy-attack-detecting-and-defeating-crashoverride
[Ref.- 20]	Developing Cyber Forensics for SCADA Industrial Control Systems. https://www.researchgate.net/publication/266477470_Developing_Cyber_Forensics_for_SCADA_Industrial_Control_Systems
[Ref.- 21]	GREYENERGY a successor to BlackEnergy. https://www.eset.com/ca/business/resources/white-papers/greyenergy-a-successor-to-blackenergy/
[Ref.- 22]	W32.Stuxnet Dossier. https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

incibe
cert_