MÁSTER UNIVERSITARIO EN SEGURIDAD DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

TRABAJO FIN DE MÁSTER

EDR para PYME:
P-EDR Arch

Autor

Francisco Javier Pérez Sánchez

Director del Trabajo Fin de Máster

Alejandro Ramos Fraile

CURSO 2021-2022

DECLARACIÓN PERSONAL DE AUTORÍA

Francisco Javier Pérez Sánchez con DNI 45380753-M, estudiante del Máster Universitario en Seguridad de las Tecnologías de la Información y las Comunicaciones en la Universidad Europea de Madrid, como autor de este documento académico titulado "EDR para PYMES: P-EDR Arch" y presentado como Trabajo Final de Máster

DECLARO QUE

Es un trabajo original, que no copio ni utilizo parte de obra alguna sin mencionar de forma clara y precisa su origen tanto en el cuerpo del texto como en su bibliografía y que no empleo datos de terceros sin la debida autorización, de acuerdo con la legislación vigente.

Asimismo, declaro que soy plenamente consciente de que no respetar esta obligación podrá implicar la aplicación de sanciones académicas, sin perjuicio de otras actuaciones que pudieran iniciarse.

En Alcobendas, a 11 de septiembre de 2022.

Fdo: Francisco Javier Pérez Sánchez

AGRADECIMIENTOS

A mis padres, Antonio y Ana, y a mi hermano, Antonio; por todo el apoyo y cariño que me han dado no solamente durante esta etapa, sino a lo largo de toda mi vida. Sin sus valores y paciencia mostrados, no habría sido posible llegar adonde estoy.

A todos los profesores del máster de ciberseguridad de la Universidad Europea de Madrid; por su amabilidad, enseñanzas y buena orientación proporcionadas a lo largo de todo el curso.

A mis amigos, tanto a los que ya tenía como a los que me llevo tras esta experiencia. Por todo el apoyo que me han brindado de manera desinteresada cada vez que lo he necesitado en momentos difíciles.

RESUMEN: Aunque la ciberseguridad es una rama que cada vez cuenta con más tecnologías en el mercado disponibles para las organizaciones, aún hay franjas dentro de este sector que no cuentan con la oportunidad de acceder al amplio abanico de posibilidades a nivel defensivo que les permitan bastionar de manera adecuada sus sistemas y/o arquitecturas frente a una cantidad de amenazas que va en aumento. Este riesgo supone fugas de información, pérdidas reputacionales y daños económicos considerables que ponen en peligro la continuidad de negocio de estas empresas. Esta falta de acceso a los medios viene dada principalmente por la carencia monetaria por parte de las corporaciones de obtener las herramientas que necesitan incorporar para cubrir sus necesidades.

Así pues, en este trabajo se ha desarrollado una arquitectura que permite realizar la detección, monitorización, análisis y respuesta de los ciberataques dados en unas máquinas de las que el propio usuario dispone, en este caso, un conjunto de ordenadores que componen la red de una supuesta pequeña empresa. De estos datos, tomados en tiempo real, se recaban: alerta generada por el ciberataque, técnica utilizada, identificador referente en MITRE ATT&CK, táctica donde se encasilla, *hash* y nombre del fichero, línea de comandos del sistema relacionado, fecha y hora de cuando se ejecutó; entre otros. Para ello se han utilizado diversas tecnologías, entre las que se destacan la implementación de un *script* que coopera con la herramienta *Sysmon* del paquete de *Sysinternals* además de con un agente de recopilación de *logs* del sistema conocida como *Winlogbeat*, y de una plataforma de respuesta incidentes de seguridad escalable conocida como *TheHive* (Adouani, y otros, 2022). Específicamente, la cooperación de las tres primeras herramientas constituye las bases de un sistema de *Endpoint Detection and Response*, donde *Sysmon* proporciona la detección, *Winlogbeat* la monitorización y el *script* la respuesta necesaria, todo de manera continuada. Por otro lado, *TheHive* es una plataforma de *tres en uno*; donde tres tecnologías independientes colaboran entre sí con el objetivo de recabar y analizar la mayor información posible de un nuevo ciberataque, reportarlo y preparar una contra respuesta de la manera más efectiva y rápida, en el caso de que sea necesario.

Los resultados obtenidos demuestran que se trata de un proyecto de ciberseguridad que supone una mejora notable frente al seguimiento, análisis y defensa continuada de amenazas que pongan en peligro a aquellas organizaciones que no pueden permitirse una tecnología tan costosa; ya que proporciona el establecimiento de una arquitectura potente, fiable, escalable, fácil de mantener y al alcance de cualquier persona con conocimientos en el campo.

Palabras clave: Sysmon, EDR, alertas, ciberdefensa, MITRE ATT&CK

ABSTRACT: Although cybersecurity is a branch that increasingly has more technologies on the market available to organizations, there are still fringes within this sector that do not have the opportunity to access the wide range of possibilities at a defensive level that allow them to adequately bastion their systems and/or architectures against an increasing number of threats. This risk involves information leaks, reputational losses and considerable economic damage that endangers the business continuity of these companies. This lack of access to the media is mainly due to the lack of money on the part of corporations to obtain the tools which they need to incorporate in order to cover their needs.

Thus, in this work an architecture has been developed that allows the detection, monitoring, analysis and response of cyberattacks given in some machines that the user himself has, in this case, a set of computers that make up the network of a supposed small company. From this data, taken in real time, the following are collected: alert generated by the cyberattack, technique used, reference identifier in MITRE ATT&CK, tactic where it is typecasted, hash and name of the file, related system command line, date and time when it was done; among others. Various technologies have been used for this, among which the implementation of a *script* that cooperates with the *Sysmon* tool of the *Sysinternals* package as well as with a system *log* collection agent known as *Winlogbeat*, and an incident security and scalable response platform solution known as *TheHive*. Specifically, the cooperation of the first three tools forms the basis of an *Endpoint Detection and Response* system, where *Sysmon* provides the detection, *Winlogbeat* the monitoring, and the developed *script* gives the necessary response, all on an ongoing basis. On the other hand, *TheHive* is a three-in-one platform; where three independent technologies collaborate with each other with the aim of gathering as much information as possible about a new cyberattack, reporting it and preparing a counter-response effectively and quickly, if necessary.

The results obtained show that it is a cybersecurity project that represents a notable improvement against the monitoring, analysis and continuous defense of threats that endanger those organizations that cannot afford such an expensive technology; since it provides the establishment of a powerful, reliable, scalable architecture, easy to maintain and within the reach of anyone with knowledge in the field.

Keywords: Sysmon, EDR, alerts, cyber defense, MITRE ATT&CK

Índice general

Índice de figuras

Índice de tablas

Siglas

ADMIT *Arquitecture Design (or Development) Methodology for Information Technology*

ANF *Appliication Normative Framework*

API *Application Programming Interface*

AV *Antivirus*

CCN Centro Criptológico Nacional

CISO *Chief Information Security Officer*

DNS *Domain Name System*

DSA *Digital Signature Algorithm*

ECC *Elliptic Curve Cryptography*

EDR *Endpoint Detection and Response*

ELK Stack *Elasticsearch, Logstash and Kibana Stack*

EPP *Endpoint Protection Platform*

GDPR *General Data Protection Regulation*

HTTP *Hyper Text Transfer Protocol*

ID *Identifier*

IOC *Indicator of Compromise*

IP *Internet Protocol*

ISO *International Standards Organization*

JSON *Javascript Object Notation*

MISP *Malware Information Sharing Platform*

MITRE ATT&CK *MITRE Adversarial Tactics, Techniques and Common Knowledge*

NGAV *Next Generation Antivirus*

NIST *National Institute of Standards and Technology*

ONF *Organization Normative Framework*

P-EDR Arch *Personal-Endpoint Detection and Response Architecture*

PIB Producto Interior Bruto

PID *Process Identifier*

PYME Pequeña y Mediana Empresa

RACI *Responsible, Accountable, Consulted and Informed*

RAG *Red, Ambar, Green (metric)*

RAM *Random Access Memory*

RSA *Rivest-Shamir-Adleman (algorithm)*

SDCL *Systems Development Life Cycle*

SGSI Sistema de Gestión de la Seguridad de la Información

SIRP *Security Incident Response Platform*

SOC *Security Operations Center*

SSL *Secure Sockets Layer*

TFM Trabajo Fin de Máster

TLS *Transport Layer Security*

TLT *Targeted Level of Trust*

URL *Uniform Resource Locator*

VBA *Visual Basic for Applications*

WMI *Windows Management Instrumentation*

XDR *eXtended Detection Response*

1. INTRODUCCIÓN

En este primer capítulo se presentan diferentes aspectos generales de este Trabajo de Fin de Máster (TFM), específicamente, aquellos vinculados con la motivación que ha llevado a la elaboración de éste, los objetivos que desean cumplirse, el tema que comprende, los resultados obtenidos; entre otros.

1.1. CONTEXTO Y JUSTIFICACIÓN

Actualmente, todo está ampliamente informatizado. Con la aparición de tecnologías cada vez más innovadoras, surgen métodos de vulneración más metódicos contra los sistemas de las organizaciones; lo que dificulta ampliamente el bastionado y en consecuencia, la protección de información sensible que pone en peligro la reputación de las empresas. Aunque las herramientas que utilizan bases de datos basados en análisis de firma o basados en heurísticas, como los antivirus, suponen una primera línea de defensa contra estos ciberataques; pueden ser mecanismos de ciberdefensa insuficientes para el ingente número de amenazas que existen hoy en día. Es por ello, que se utilizan sistemas de protección conocidos como *Endpoint Detection and Response* (EDR) que incluyen la funcionalidad de estos antivirus tradicionales junto a herramientas de monitorización, y en algunos casos; inteligencia artificial para ofrecer una respuesta rápida y eficaz ante los riesgos y las amenazas más complejas que afectan a las corporaciones.

No obstante, estos sistemas de protección son altamente costosos para las Pequeñas y Medianas Empresas (PYME); lo que hace prácticamente inalcanzable que puedan incorporar dicha tecnología a su arquitectura.

Por ende, la motivación principal que lleva a la realización de este proyecto, y que por lo tanto, lo justifica; es el aprendizaje de una serie de tecnologías que van en auge y que suponen un pilar fundamental en la protección de los activos de información de cualquier empresa que tenga una ventana abierta a Internet. La adquisición de estos conocimientos, llevarán a la realización de una arquitectura basada en un sistema de protección EDR, que podrá ser empleada por cualquier usuario o empresa que así lo requiera.

Este TFM, abarca un prototipo inicial (y por supuesto, escalable), que permite realizar todas y cada una de estas tareas, mediante la utilización de tecnologías ya existentes en cooperación con algunas personalmente desarrolladas e integradas.

1.2. PLANTEAMIENTO DEL PROBLEMA

La adquisición de estos sistemas de protección EDR son altamente costosos para las PYME o individuos particulares; lo que impide la inclusión de dicha tecnología en sus equipos e infraestructuras. Aunque existen proyectos de código abierto que intentan poner al alcance las técnicas que utilizan los sistemas EDR, son proyectos que no llegan a asimilarlos en su completitud; bien porque carecen de una parte de monitorización centralizada, no cuentan con unos niveles de detección mínimos que aseguren robustez, no proporcionan herramientas de contrarrespuesta adecuados, o son fácilmente evadibles. Por lo tanto puede resumirse en que, son soluciones que aunque están disponibles públicamente, no ofrecen unos niveles de protección idóneos para lo que se espera.

Este TFM, abarca un prototipo inicial (y por supuesto, escalable y altamente configurable), que permite realizar todas y cada una de estas tareas, mediante la utilización de tecnologías ya existentes en cooperación con algunas personalmente desarrolladas e integradas; lo que supone una innovación para un problema tan distendido como es la adquisición de un sistema de protección EDR por parte de la PYME o usuario que lo necesite.

1.3. OBJETIVOS DEL PROYECTO

El objetivo principal de este TFM es el desarrollo de una arquitectura EDR completamente gratuita y *open-source* que permita a individuos de casa o PYME, el establecimiento de una herramienta de alta utilidad que permita bastionar de manera adecuada sus equipos y/o dominios. Para ello, se hará uso de mecanismos e implementaciones ya existentes así como de las nuevas que se requieran para alcanzar el objetivo global. Este EDR debe ser adaptable a nivel de las necesidades del usuario que quiera utilizarlo.

Para la obtención de dicho objetivo, se plantean los siguientes objetivos específicos:

- Configuración adecuada de la herramienta *Sysmon* del paquete *Sysinternals*, para monitorización de los eventos del sistema (Russinovich & Garnier, 2022). Implantación de un *script* para medidas pertinentes frente a los distintos eventos.
- Realización de un fichero de configuración de reglas que permita establecer flexibilidad a la hora de imponer las medidas que ejercerá el EDR en cuestión (Perez, SysmonCommunityGuide - Configuration, 2021) (Catch All (MS Windows Event Logging XML - Sysmon), 2022).

- Instalación y configuración de motor de búsqueda y analítica *Elasticsearch* (Elastic, 2022), para envío de la información relevante al servicio de monitorización centralizado a disposición del equipo de *Security Operations Center* (SOC).
- Configuración e implementación de agentes en los que equipos e infraestructuras requeridos, para recopilación de la información que será enviada al motor de búsqueda.
- Adecuación de escenario de herramientas a ser usadas por parte de un posible equipo de SOC, para realizar una contrarrespuesta de la manera más rápida y eficaz posible frente a amenazas complejas detectadas.
- Conexión segura entre todos los elementos anteriormente citados.
- La arquitectura debe ser totalmente gratuita a nivel de *software*.

1.4. RESULTADOS OBTENIDOS

Tras la realización de las múltiples pruebas realizadas con la arquitectura implementada, los resultados obtenidos son altamente positivos. Se ha desarrollado un sistema de protección EDR completamente gratuito y *open-source* que actúa en base a comportamientos de los eventos registrados por *Sysmon* en el equipo de interés, altamente configurable en función de estos eventos, con una robustez lo suficientemente exigente para que sea dificultoso de evadir y con capacidad de ser reutilizable al albergar la alternativa de añadir nuevas funcionalidades que se estimen necesarias. En base a la batería de pruebas utilizada a modo de posible *malware* en las pruebas, y a las funciones que trae de manera genérica; este sistema de protección EDR ha mostrado en sus resultados que es capaz de:

- Capacidad de dictaminar las reglas de inclusión o exclusión en base a los *Identifiers* (ID) de eventos de *Sysmon.*
- Recoger las alertas que se consideren necesarias en el gestor de eventos de *Windows.*
- Detener un proceso dictaminado como malintencionado en base a su *Process Identifier* (PID).
- Detener a los hijos que invoca un proceso dictaminado como malintencionado en base al PID original.
- Detener a los procesos padre de un proceso dictaminado como malintencionado en base al PID original.
- Detener una serie de conexiones remotas inusuales con el equipo; en base a puerto, *Internet Protocol* (IP) (Kaspersky, 2022) o *Uniform Resource Locator* (URL).
- Detener un hilo que inyecta un proceso en el sistema.

- Añadir automáticamente reglas de *firewall* para bloquear conexiones entrantes/salientes de un proceso.
- Realizar escaneos mediante reglas *Yara* (Arntz, 2017) (GoDaddy, 2017) (Analyzing Document Macros with Yara, 2019) (Academy, 2022).
- Realizar eliminación de ficheros en base a detecciones realizadas por reglas *Yara.*
- Eliminar y almacenar ficheros potencialmente peligrosos en base al *hash* que lo define en una carpeta oculta y únicamente inaccesible a nivel de *NT-System*. Está carpeta está protegida *System Access Control List* (ACL) (Hartong, Sysmon 11 — DNS improvements and FileDelete events, 2020) (Perez, SysmonCommunityGuide: File Delete, 2021) .
- Restaurar ficheros eliminados como falsos positivos.
- Realizar un volcado de *Random Access Memory* (RAM) de un proceso concreto (Stackoverflow, Take a user dump using powershell, 2013) (Liang, y otros, 2022).
- Aislar un equipo infectado con el objetivo de evitar, por ejemplo, que un *malware* se extienda por el dominio al que dicho equipo pertenece.

Además, todos estos eventos del sistema recogidos por *Sysmon* son elevados como alertas a un equipo externo que ejerce una función de sistema de monitorización centralizado mediante un *software* conocido como *TheHive*; el cual se encuentra completamente supervisado por un equipo de SOC que podrá ejercer de la manera más rápida y eficaz posible una contrarrespuesta a la amenaza encontrada. Esta información podrá ser embebida mediante una pieza fundamental de la arquitectura conocida como *Cortex*, la cual analizará la información recogida en los observables del evento; añadiendo datos adicionales que serán de gran utilidad al equipo de SOC. También es posible compartir dicha información mediante una plataforma de compartición de información de *malware*, conocida en inglés como *Malware Information Sharing Platform* (MISP), en el caso de que así sea considerado.

Como información extra, aparte de la que aporta el propio *Sysmon* en base a la regla que encasille, se incluye de manera genérica (aunque extensible):

- Tipo de alerta generada.
- Técnica empleada.
- ID de referencia en el *MITRE Adversarial Tactics, Techniques, and Common Knowledge* (*MITRE ATT&CK*).
- Táctica empleada según *MITRE ATT&CK.*
- Banderas que indican la clase de medida tomada en función de la alerta generada.

Adicionalmente, y como ya se ha indicado, se recoge información muy útil en función de la regla generada en el fichero de configuración de *Sysmon*, como puede ser *hash* de un fichero en concreto, nombre de fichero original (Hartong, Sysmon 10.0 - New features and changes, 2019), usuario, línea de comandos ejecutada, fecha y hora del ataque; entre otros.

1.5. ESTRUCTURA DE LA MEMORIA

Este documento se dividirá en varias secciones las cuales amasan la información más importante relacionada con el proyecto en cuestión. Dichas secciones son:

- **Capítulo 2: Estado de la cuestión.** En este capítulo se recogerán aquellas tecnologías similares así como estudios que han intentado realizar unas bases generales o un enfoque parecido a lo que abarca este proyecto.
- **Capítulo 3: Descripción del problema.** En este capítulo se formaliza una necesidad en base al estado de la cuestión planteado y los objetivos que se quieren establecer.
- **Capítulo 4: Solución propuesta.** En este capítulo se indican los objetivos que se quieren establecer, los logros a alcanzar, la metodología planteada, las herramientas usadas, la implementación dada, el presupuesto del proyecto y la planificación que se ha seguido para llegar a la solución final.
- **Capítulo 5: Pruebas y validación.** En este capítulo se realiza un análisis textual de las pruebas a realizar, la batería de pruebas a utilizar y que se espera obtener.
- **Capítulo 6: Resultados.** En este capítulo se recoge el fruto de la realización de las pruebas formuladas en el punto anterior; y se hace una comparativa textual entre lo que se estimaba y lo que se ha conseguido.
- **Capítulo 7: Conclusiones.** En este capítulo se incluyen las ideas obtenidas tras la realización de todo este proyecto.
- **Capítulo 8: Trabajos futuros.** En este capítulo se concluye la memoria a modo de desenlace con las líneas de trabajo futuro y el fruto que puede realizarse a partir de este TFM.
- **Apéndices.** En este capítulo, se recapitulan una serie de subapartados que sirven para suplir el proyecto en su completitud, como son referencias bibliográficas, diario de investigación, manual de instalación, manual de usuario y guía de solucionado de errores.

2. ESTADO DE LA CUESTION

En esta sección se recoge un breve resumen respectivo al ámbito establecido sobre el tema a tratar, así como tecnologías y arquitecturas ya existentes, principales amenazas que se encuentran y justificación del proyecto a realizar en este trabajo.

2.1. ANTECEDENTES

Las soluciones EDR surgen como herramientas capaces de cubrir brechas de seguridad que otras tecnologías, como los antivirus o las *Endpoint Protection Platforms* (EPP) de última generación, no eran capaces de tratar (Wikipedia, 2022). Esta necesidad surge en primera instancia alrededor del año 2010, con los primeros ataques de *phishing,* impulsados mayoritariamente a través de la ejecución de *macros* almacenadas en documentos de la suite de *Microsoft Office*, como pueden ser *Microsoft Word, Excel* o *Powerpoint* (Cloutier, 2021). La amenaza que esto suponía era considerable, ya que la utilización de estos programas en las compañías estaba ampliamente generalizada, con la compartición de estos ficheros entre departamentos y organizaciones de manera continuada; todo esto teniendo en cuenta que la extensión de los archivos no era maliciosa como tal, sino en su lugar, el contenido que albergaba el propio documento; lo que lo hacía indetectable para los sistemas de protección del momento (Maayan, 2020).

Incluso a fecha de hoy, los propios antivirus tradicionales usan mecanismos de firma y heurísticas para detectar *malware* y procesos maliciosos; lo que los limita exclusivamente a aquella información que almacenan en sus bases de datos (N-able, 2020). En la práctica, estos datos suelen abarcar únicamente el 57% de los ataques totales que se realizan. De manera adicional, únicamente pueden detectar estos ficheros potencialmente maliciosos una vez se encuentran alojados en el sistema, lo que abre una pequeña ventana para que un actor malintencionado pueda actuar.

Bajo estas premisas, la primera aparición de un sistema EDR viene dada a manos de Anton Chuvakin; el cual lo describe como una herramienta capaz de prevenir, detectar y monitorizar de manera centralizada los equipos y estructuras que componen las redes y dominios de una corporación. Este avance, que surge en el año 2013 y se mantiene en la actualidad; supone un enorme hallazgo a nivel de ciberdefensa, ya que permite entre todas sus funcionalidades, analizar los eventos de los *endpoints* de los sistemas en búsqueda de comportamientos inusuales y actuar en base a ellos. Volviendo al ejemplo anterior, si un fichero de la suite de *Microsoft Office* genera a través de una *macro* un proceso hijo con una consola de comandos

con privilegios de administrador; sería detectado por la solución EDR como un comportamiento fuera de lo común y bloqueado previo a su ejecución; impidiendo que el código malicioso se efectuase. Este sencillo ejemplo es trasladable a otros *malware* más sofisticados que han ido surgiendo estos años atrás, y que son material de vanguardia para los actores maliciosos como son *WannaCry, Onyx, CryptoLocker, Ryuk, Teslacrypt, Cerber, SamSam, Cryptowall, NotPetya, Bad Rabbit, Locky, Jigsaw, Black Basta, Mindware* o *Retefe* (Grossman, 2017) (Cyfirma, 2022) (Recovery, 2022) (Burdova, 2022) (Wang, 2022).

De hecho, y tal como se constata en la enumeración de *malware* anterior, así como en estudios estadísticos provistos por compañías como *Symantec* o *Heimdal Security*; la vertiente más empleada en el actual año 2022 es el *ransomware*, afectando globalmente a 1 de cada 40 empresas, un incremento total del 59% comparado a las 1 de cada 60 empresas que se veían afectadas en el pasado año. Concretamente, se trata de un tipo de amenaza que secuestra los equipos y encripta la información dentro de éstos a cambio de una serie de beneficios para el ciberdelincuente. La mayor parte de antivirus (AV) y *Next Generation Antivirus* (NGAV) existentes no son capaces de detectar esta clase de vulnerabilidades, las cuales suponen pérdidas anuales por daños según *Dataprot*, de hasta 20 billones de doláres (Vojinovic, 2022). Además, se destaca un trabajo de investigación realizado por el instituto de investigación independiente en materia de seguridad informática en Alemania, también conocido como AV-TEST, donde se llega a la conclusión de que la mayor cantidad de ataques de esta índole son realizados en sistemas operativos *Windows*, una plataforma ampliamente utilizada en el día a día de la mayoría de las corporaciones.

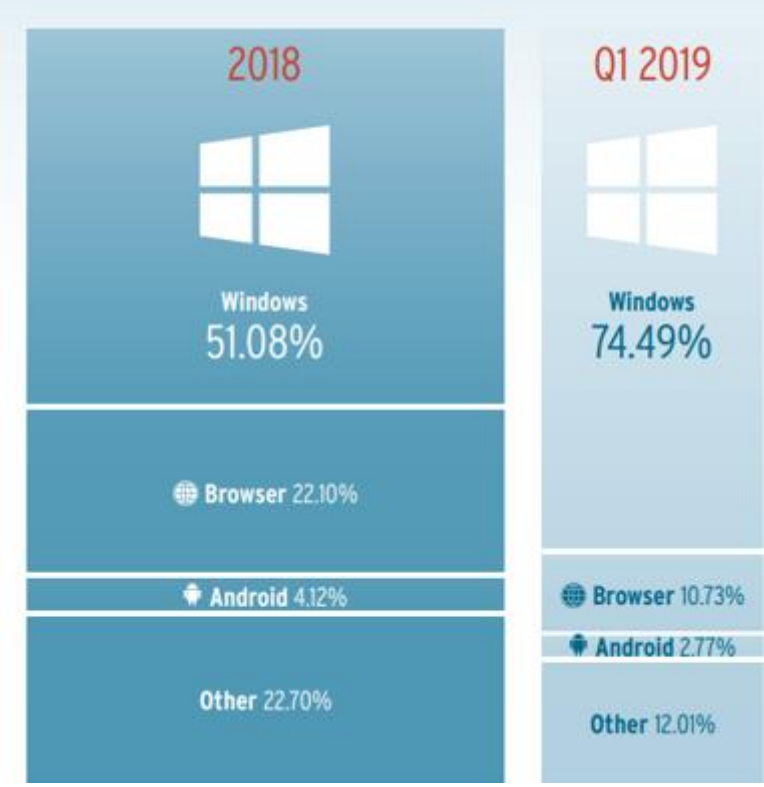

Ilustración 1. Sistemas operativos más empleados según AV-TEST

Siguiendo un estudio titulado *"Ransomware in 2022: Evolving Threats, slow progress"* realizado por Alexander Culafi para *TechTarget*, la mayoría de las organizaciones que se ven afectadas por un ataque de este tipo; recuperan únicamente una porción de los datos que tenían previo a la afectación de sus sistemas después de abonar la cantidad solicitada por los ciberdelincuentes; o utilizar copias de seguridad realizadas por la propia compañía y almacenadas con anterioridad. En otras palabras, cumplir con las propias exigencias, tampoco asegura la total recuperación de la información pérdida, es más, ni siquiera asegura una parte de ella o su distribución. Adicionalmente, Alexander Culafi indica en su artículo, que la mayor parte de estas amenazas tuvieron éxito debido a que las barreras implementadas por las industrias afectadas eran insuficientes (Culafi, 2022).

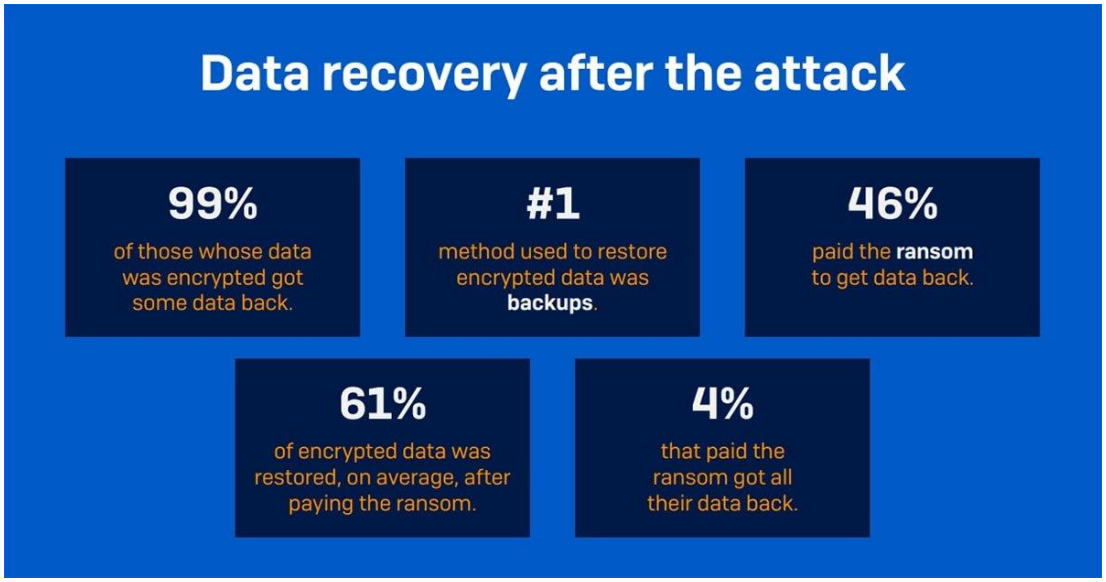

Ilustración 2. Porcentajes relativos a las compañías afectadas por ransomware (2022)

No es sorprendente por lo tanto, que el papel de un sistema de protección EDR sea tan sumamente importante en la defensa para los equipos y dominios de las redes de una compañía; ya que son capaces de aplicar medidas de prevención, detección y respuesta ante las amenazas anteriormente citadas.

Con todo esto, y tras la propuesta de Chuvakin, no han sido pocos los negocios que comenzaron el desarrollo de sus propias soluciones EDR para colocarlos a la venta en su posteridad, y así, proporcionar una oportunidad para aportar unos niveles de seguridad sustanciales a los equipos y sistemas de las corporaciones; modelo que en este momento se sigue manteniendo.

2.2. CONTEXTO Y JUSTIFICACIÓN

Cabe esperar que un sistema de protección que ofrece unas medidas tan innovadoras; suponga una adquisición únicamente al alcance de aquellas organizaciones que tengan un nivel económico lo bastante elevado para incluirla en su arquitectura. Todo esto sin contemplar los costes de mantenimiento y persistencia, lo que hace que sea un recurso aún más encarecido. Muchos de estos casos serían los EDR de marcas reconocidas, como *Symantec, CrowdStrike, FireEye, Carbon Black* o *Cynet*.

En base a esto, se han ido desarrollando prototipos en distintos repositorios por parte de usuarios que intentan simular el comportamiento de un sistema de esta índole, utilizando herramientas del propio equipo en conjunto con algunas desarrolladas por sí mismos, como es el caso del proyecto *whids* (whids, 2022) o *sysmon-edr* (Sysmon EDR Active Response Features,

2021) (del cual se ha basado este proyecto). Del mismo modo, hay arquitecturas EDR ya formalizadas que son gratuitas como es el caso de *OpenEDR* de *ComodoSecurity.* No obstante, estos prototipos y arquitecturas ya existentes tienen diversas problemáticas, ya que, o no cuentan con alguna de las características que definen a un EDR, son proyectos inacabados o difíciles de implementar en los sistemas de una organización; o por otra parte, no son plenamente de código abierto como prometen. Por otro lado, no se han encontrado entre estos proyectos gratuitos; contemplaciones reales de realizar conexiones con *software* especialmente diseñado para que los equipos de SOC puedan tratar las alertas con la suficiente importancia que se considera que merecen; ofreciéndoles a estos grupos las utilidades necesarias en materia de ciberseguridad para profundizar de la manera óptima en las amenazas encontradas. Lo más cercano que incorporan algunas soluciones son análisis de los datos recogidos de manera genérica mediante la *Elastic, Logstash and Kibana* (ELK) *Stack*.

La finalidad de este TFM es, por lo tanto, la realización de una solución EDR que contenga todas las características propias que definen a un recurso de este tipo, y que asimismo, sea de fácil implementación, gratuita y completamente *open-source*. Además, se ha procurado en todo momento que sea fácilmente mantenible, configurable y escalable, a fin de que las empresas o individuos que decidan incorporar esta tecnología en sus sistemas; no tengan que preocuparse por la vida útil de este producto.

2.3. PROYECTOS SIMILARES Y APORTE DEL PROYECTO

En este apartado, se mencionan los proyectos similares a éste, lo obtenido tras las pruebas realizadas por parte de sus autores, una evaluación personal de que se considera que necesita para considerarse completo como solución EDR, y por último, que proporciona adicionalmente la arquitectura presentada en este TFM frente a dichos proyectos.

2.3.1. *WINDOWS HOST IDS* (WHIDS)

Presentado como un proyecto de código abierto el cual combina características de un sistema de detección de intrusos con detección basada en capacidades de respuesta a incidentes. Utiliza *Sysmon* como pilar de su arquitectura, y realiza gran parte de sus funcionalidades conforme a reglas *gene*, las cuales detectan patrones dados en los eventos de los *logs* de *Windows*, y dado que dichos eventos pueden considerarse un indicador de compromiso, dichas reglas *gene* pueden ser usadas, por ejemplo, para la configuración de las reglas de filtración de eventos de *Sysmon* sin necesidad de un propio fichero de configuración de *Sysmon*. Esto significa, entre otras cosas, que la correlación de eventos se hace a nivel de host, al utilizar esta tecnología como

base. Estos eventos serán enviados posteriormente a una plataforma centralizada donde podrá ser tratada de manera adecuada por entidades pertenecientes a un equipo de respuesta a incidentes.

Además de ser fácilmente integrable con otras herramientas de código abierto, como ELK o MISP, permite una flexibilidad muy grande en lo relativo a su motor de detección, es decir, las reglas *gene* en las que se basa el sistema son muy configurables. Asimismo, estas reglas se basan en el marco de ATT&CK, luego están respaldadas por una entidad que avala la veracidad de dichas reglas.

Permite la ejecución de *scripts* o ejecutables en ficheros que no se encuentran en el propio *endpoint*, siendo utilizables por parte de equipos de respuesta a incidentes mediante *hooks*.

Por último, la arquitectura presentada puede cooperar con cualquier producto de antivirus existente, aunque se recomienda encarecidamente su uso con *Windows Defender.*

Tras comprobar las pruebas realizadas con este EDR, frente a algunas amenazas, se llega a que se trata de una solución EDR con ciertas carencias:

- No aporta información de una amenaza detectada a nivel de *host*, lo cual influye en que si un usuario se ve afectado por una amenaza, únicamente el equipo de respuesta a incidentes tenga constancia de ello.
- La respuesta que ofrece frente a eventos por parte de la propia solución únicamente plantea volcados en memoria de ficheros, procesos y registros. El resto de posibles respuestas reactivas las delega enteramente en un antivirus. Esta solución además de plantear el riesgo de que un equipo se vea vulnerado por ataques más minuciosos (como *ransomware*), presenta un problema de poca flexibilidad a la hora de configurar la respuesta exacta que quiere darse como medida de prevención por parte de un equipo de respuesta a incidentes.
- Las reglas *gene* únicamente buscan patrones en los ficheros de eventos de *Sysmon*. Esto significa que, basa su totalidad de detección frente a comportamientos reflejados en los eventos, y no en el propio contenido de ficheros, ejecutables o *scripts*; como podrían hacer por ejemplo, la utilización de reglas *YARA*.
- Aunque la solución incluye campos de información extra en los eventos de *Sysmon*, como por ejemplo, tamaño de fichero; no incluye campos realmente importantes para equipos de respuesta a incidentes como identificador de MITRE y técnicas, táctica y alerta asociadas a éste.

- Utiliza los proveedores de *Event Tracing for Windows* (ETW) para las reglas *gene*, que aunque ofrecen una segmentación más específica de los componentes de un evento de *Windows*, y por lo tanto más flexibilidad, complica enormemente la tarea de configuración de la arquitectura. Se recuerda que se busca un balance entre flexibilidad y facilidad de uso.

2.3.2. SYSMON-EDR

Basado en la utilización de *Sysmon*, junto a los comandos proporcionados por la propia *Powershell* de *Windows*, presenta una solución EDR basada en el uso de un *script* que se encuentra constantemente en uso en segundo plano en el sistema, el cual monitoriza constantemente todos los eventos que llegan a *Sysmon* a partir de un registro *WMI* y ofrece respuesta en función de éstos.

Para ello, este *script* presenta una serie de banderas que son incluidas en las distintas reglas de *Sysmon*, proporcionando la funcionalidad deseada en base a estos. Dichas banderas vienen acompañadas al mismo tiempo, de información referente al marco de MITRE ATT&CK, lo que avala la solución planteada. Además, utiliza reglas *YARA* para el análisis del contenido de ficheros y ejecutables, permitiendo detectar archivos maliciosos independientemente de su nombre (o incluso ofuscación). Todos los eventos nuevos son elevados en forma de ventana a modo de alerta para el usuario que utilice esta herramienta.

Todas las reglas de *Sysmon* son enteramente modificables, al igual que el fichero de respuesta proporcionado, al ser un *script* visible (*open-source*). Únicamente esta posibilidad, junto al propio fichero de configuración de *Sysmon*, proporcionada una flexibilidad extremadamente alta la hora de implementar una configuración determinada en una arquitectura. Del mismo modo, debido a su simplicidad de implementación, es fácilmente integrable junto a otras herramientas de código abierto y gratuitas.

Tras las pruebas realizadas, se encuentra que es una base muy potente (y de hecho, utilizada para este proyecto), pero con una diversidad de carencias:

- Los eventos creados por *Sysmon* son controlados a nivel de *host*, pero no se elevan a un sistema aparte que sirva para un equipo de respuesta a incidentes de ciberseguridad.
- No se incluye una base de reglas *YARA* que pueda ser útil para detectar las amenazas actuales.
- Únicamente es capaz de detectar 5 de los 26 eventos posibles que puede generar *Sysmon*.

- Aunque tiene diversas funcionalidades como respuesta, no incluye algunas realmente importantes como volcado de memoria de ficheros y registros, o aislamiento de un equipo infectado.
- No ofrece control para cierre del propio *Sysmon* o del *script* que proporciona respuesta.

2.3.3. COMODOSECURITY – OPENEDR

Una arquitectura altamente sotifiscada, gratuita y de código abierto, compuesta por diversos componentes de detección en *endpoint* y una interfaz para el tratamiento de alertas generadas por parte del equipo de respuesta a incidentes.

Utiliza una gran cantidad de librerías y métodos de detección para proveer de la mayor cantidad de información posible al equipo de respuesta a incidentes. Entre las librerías y módulos que maneja, se presenta una en concreto que proporciona a la interfaz web utilizada una segmentación de una amenaza en concreto en sus distintas fases a modo de 'árbol de procesos', facilitando la tare de búsqueda y análisis.

Tras su utilización, se llega a las siguientes carencias:

- No presenta ningún mecanismo de protección frente a amenazas dadas en un equipo, es decir, no hay parte de respuesta automatizada ante amenazas.
- Realmente, el proyecto es *open-source* en una parte. No permite una gran flexibilidad a la hora de configurar elementos como alertas o reglas.
- Complicado de instalar. Reducida facilidad de uso.
- Difícil de integrar con otras herramientas.
- Realmente, su funcionalidad de *response*, está limitada a lo que realice el equipo de respuesta a incidentes una vez el equipo se ha visto vulnerado por un ataque.

2.3.4. APORTE DEL PROYECTO

En lo referente a que aporta la solución EDR planteada (en conjunto) respecto al resto de proyectos anteriores, se incluye:

- Inclusión de una arquitectura completamente gratuita y *open-source* integrada por todos los componentes que definen a una solución EDR. Esta arquitectura aparece segregada en cuestión del objetivo que tengan.
- Respuesta automatizada a determinados eventos de *Sysmon* en base a las reglas. Entre estas respuestas se incluyen detener un proceso, detener un proceso padre de otro proceso, establecer reglas *firewall*, detener un hilo inyectado en un proceso, apagado

del sistema, detener conexión no autorizada, almacenamiento de ficheros potencialmente peligrosos en una carpeta de cuarentena, análisis exhaustivo mediante un set definido de reglas *YARA*, restauración de un fichero, aislamiento de equipo infectado y volcado de memoria.

- Análisis forense de ficheros potencialmente peligrosos, o del propio sistema.
- Elevación de alertas múltiples: una a nivel de host para el *endpoint* específico, y otra para la plataforma de respuesta a incidentes de ciberseguridad.
- Alta flexibilidad a la hora de establecer nuevas reglas de detección o medidas de actuación frente a las reglas elevadas.
- Análisis de contenidos de ficheros mediante reglas *YARA*, y eliminar en detección.
- Fácil capacidad de uso e integridad con otras herramientas.
- Aporte de herramientas al equipo de respuesta a incidentes para generar más información a partir de un dato proporcionado en un evento generado.
- Capacidad de enviar los reportes generados por el equipo de respuesta a incidentes a una plataforma de compartición de información de *malware*.
- Arquitectura con una vida útil extremadamente prolongada, al utilizar componentes de organismos conocidos o en constante actualización, como *Sysmon* (*Microsoft)* o *Cortex* (*TheHive Project).*

3. DESCRIPCIÓN DEL PROBLEMA

A continuación, se recoge en este apartado un inciso de la problemática encontrada tras la información aportada en la sección anterior, la necesidad que surge a raíz de dicha problemática, así como la idea resolutiva que pretende ofrecerse mediante el desarrollo de este TFM.

La principal necesidad que se produce, en base a las secciones anteriores, es proporcionar oportunidad a las organizaciones que no tienen la potencia fiscal de adquirir un producto EDR ya desarrollado por otra compañía, y que puedan incorporar esta arquitectura en sus sistemas de manera que tengan la protección necesaria frente a las amenazas empleadas por los ciberdelincuentes en la actualidad. Dicho de otra manera, la problemática que se plantea es que existe una tecnología capaz de hacer frente, de manera prácticamente inigualable respecto a otros recursos del mercado, a la mayor parte de *malware* utilizado o que se pueda desarrollar; pero que debido a las características tan exigentes que ofrece, no está al alcance de todos; y por otra parte al mismo tiempo, existen soluciones gratuitas que podrían adoptar estas PYME pero que carecen de la naturaleza que define a una solución EDR, están incompletas, o prometen funcionalidades que luego no se cumplen.

Esta necesidad principal establece un vínculo de conexión con el objetivo global de este TFM, que expone el diseño y desarrollo de una arquitectura EDR completamente gratuita y *open-source*, que facilita a individuos de casa y PYME el establecimiento de una herramienta de calidad que permita bastionar de manera robusta sus equipos y dominios. Análogamente, el cumplimiento del objetivo global es el resultado de cumplir los objetivos específicos marcados, entre los cuales se recogían la configuración y utilización de productos adquiribles por cualquier sujeto, ya que se recurre al uso de tecnologías completamente gratuitas, ya existentes o incorporadas por este mismo proyecto; tal y como se citó en el primer capítulo de este mismo escrito.

Por lo tanto, se presentará a lo largo de lo que resta del documento, una arquitectura compuesta por una serie de elementos que cohesionando entre sí, tienen la capacidad de abarcar de una manera bastante aceptable todos los problemas que vienen siendo fruto de la casuística generada en el desarrollo de soluciones EDR en los últimos años; y que se verá solventada mediante la recopilación de metodologías y mecanismos ya existentes en este ámbito; implementando en su finalidad, una arquitectura EDR gratuita, que cumple las funcionalidades esperadas para un sistema de este tipo, de código abierto y con una vida útil

extremadamente prolongada, a la que se le dará el nombre de *Personal-Endpoint Detection and Response Architecture* (o en su defecto, P-EDR Arch).

4. SOLUCIÓN PROPUESTA

Primeramente, se resumen los objetivos generales y específicos que pretenden cumplirse mediante la completitud del proyecto. Además, se mostrarán las distintas metodologías seguidas para la formalización de éste. A continuación, se especificará textualmente la planificación seguida en las distintas fases de la realización del TFM, acompañado de una tabla que relaciona la tarea a los días aplicados. Por último, se presentará de manera concisa los costes económicos que suponen el despliegue de la arquitectura.

4.1. OBJETIVOS

Como fue descrito en el apartado introductorio; se considera un único objetivo general; siendo este el diseño y desarrollo de una arquitectura de protección de equipos y sistemas para PYME y usuarios de casa, que cuente con una solución EDR incorporada y comunicación con una plataforma utilizable por un equipo de SOC con las suficientes utilidades para poder monitorizar, prevenir, detectar y actuar frente a las amenazas actuales y futuras que puedan surgir, todo esto de manera gratuita, *open-source*, escalable, adaptable, configurable y con una larga vida útil. En pocas palabras, el objetivo general constituye de manera global la elaboración de todo el proyecto en cuestión.

En cuanto a los objetivos específicos que han de cumplirse para llegar al resultado esperado, se tienen:

- Configuración adecuada de la herramienta *Sysmon* del paquete *Sysinternals*, para monitorización de los eventos del sistema. Implantación de un *script* para medidas pertinentes frente a los distintos eventos.
- Realización de un fichero de configuración de reglas que permita establecer flexibilidad a la hora de imponer las medidas que ejercerá el EDR en cuestión.
- Instalación y configuración de motor de búsqueda y analítica *Elasticsearch*, para envío de la información relevante al servicio de monitorización centralizado a disposición del equipo de *Security Operations Center* (SOC).
- Configuración e implementación de agentes en los que equipos e infraestructuras requeridos, para recopilación de la información que será enviada al motor de búsqueda.
- Adecuación de escenario de herramientas a ser usadas por parte de un posible equipo de SOC, para realizar una contrarrespuesta de la manera más rápida y eficaz posible frente a amenazas complejas detectadas.

- Conexión segura entre todos los elementos anteriormente citados.
- La arquitectura debe ser totalmente gratuita a nivel de *software*.

4.2. METODOLOGÍA

Se han considerado dos metodologías, primeramente una metodología a nivel de investigación de los recursos, y una segunda metodología oficial basada en la ISO 27034-3:2018, la cual describe el proceso general para administrar la seguridad de cada aplicación específica utilizada por una organización (ISO, 2018) (CEC, 2020).

4.2.1. METODOLOGÍA DE INVESTIGACIÓN

Debido a que se trata de un proyecto de investigación de tecnologías e integración entre ellas; puede decirse que ésta se divide en etapas con diferentes metas; donde se investiga inicialmente el concepto a implementar y como llevarlo a cabo. Esta metodología consiste, por lo tanto, en probar que la tecnología a implementar dé como resultado lo que se espera, y que lo haga de una manera eficiente y sin problemas. Por lo tanto, la recopilación de información para establecer el prototipo que conforma este proyecto puede decirse que ha seguido una metodología por capas o escalonada; ya que el hallazgo de la información para las tecnologías usadas se ha ido insertando en el producto final cuando existía una confirmación de que funcionaban correctamente con el grueso del proyecto y que, además, aportaban lo que se buscaba de ésta. En la **Ilustración 3** se muestra una imagen del esquema metodológico seguido:

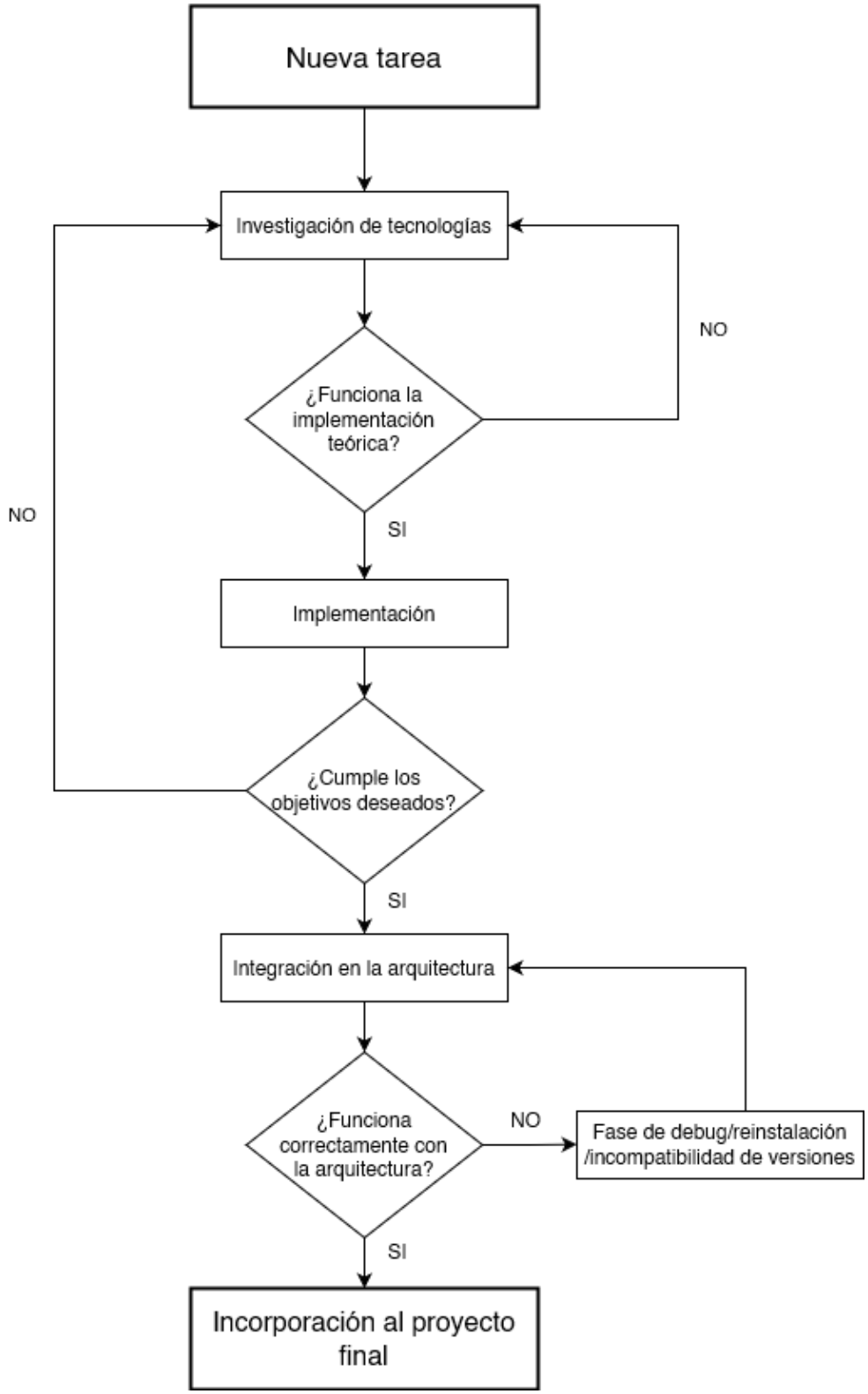

Ilustración 3. Metodología de investigación del proyecto

4.2.2. ISO 27034-3:2018

Desde otra perspectiva, el desarrollo de toda la arquitectura se ha realizado siguiendo la ISO 27034-3:2018; norma que proporciona orientación para ayudar a las empresas a instaurar la seguridad en los procesos utilizados para gestionar sus aplicaciones y es aplicable a aplicaciones desarrolladas de manera interna, aplicaciones adquiridas de terceros y donde el desarrollo o la operación de la aplicación se subcontrata. Principalmente la orientación que ofrece está enfocada en el diseño, selección, especificación y aplicación de los controles de seguridad de la información mediante un conjunto de procesos que están integrados a través del Desarrollo de Sistemas de Ciclo o *Systems Development Life Cycle* (SDLC) de una organización.

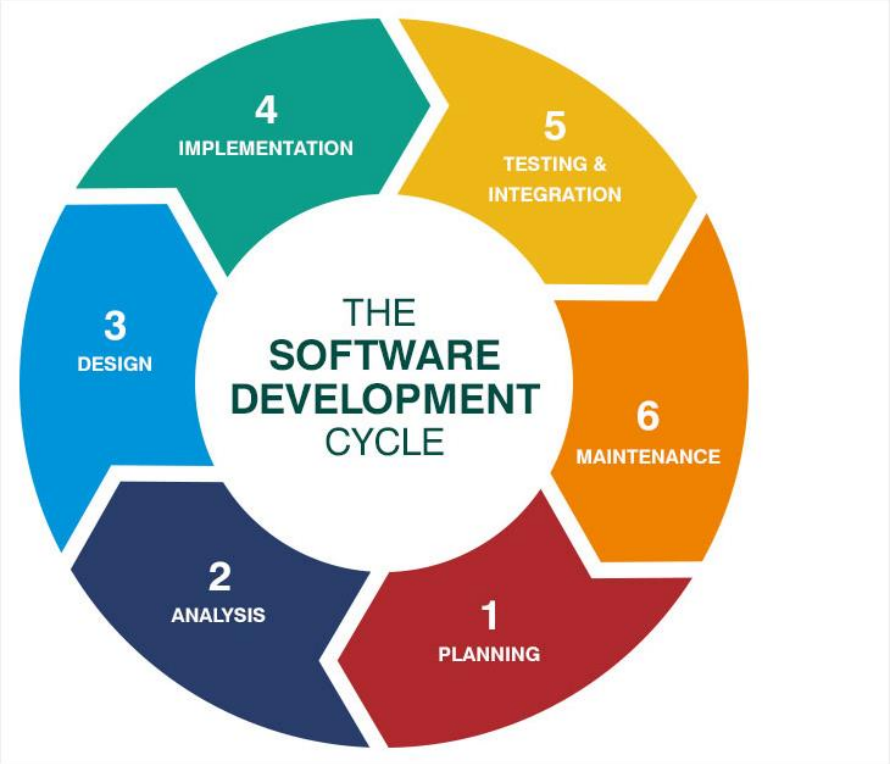

Ilustración 4. Esquema de Desarrollo de Sistemas de Ciclo de una organización

Además, dicha norma trata todos los aspectos de la determinación de los requisitos de seguridad de la información, así como de la prevención del uso o accidentes de una aplicación no autorizada.

La ISO 27034 se base en los siguientes principios fundamentales:

- La seguridad se observa como un requisito
- La seguridad de aplicaciones depende del contexto
- Inversión correcta para la seguridad de las aplicaciones

- La seguridad de las aplicaciones tiene que ser demostrada

Debido a que lo que se pretende con este TFM es la formalización de una arquitectura mediante la cooperación entre aplicaciones de terceros así como desarrolladas personalmente (interna) de la manera lo más segura posible, este estándar encaja perfectamente en la metodología a seguir. **Es por ello que, aunque se trata de una metodología genérica, ha sido adaptada al contexto del proyecto en cuestión.**

4.2.2.1. REVISIÓN

La ISO 27034-3:2018, incluye una serie de requisitos a modo de cláusulas que deben cumplirse:

1) **Alcance. –** Esta primera cláusula va alineada con la política propuesta de la organización, y es donde se incluye que va a implementarse, mejorarse o asegurarse.
2) **Referencias normativas. –** En esta segunda cláusula se recopilan todos aquellos estándares o normativas publicados de manera oficial, y que servirán de guía para el documento a tratar.
3) **Términos y definiciones. –** En esta tercera cláusula se recoge toda aquella terminología importante aplicada en el contexto y que hay que referenciar para el correcto entendimiento por parte del lector.
4) **Términos abreviados. –** En esta cuarta cláusula se indican las siglas que referencian los términos incluidos en la cláusula tres.
5) **Proceso de gestión de la seguridad de las aplicaciones. –** En esta quinta cláusula se especifica todo el proceso de manejo de seguridad de cada aplicación específica usada o desarrollada por la organización.
6) **Pasos en el proceso de gestión de la seguridad de las aplicaciones. –** En esta sexta cláusula se adoptan todos los conceptos de la cláusula anterior de manera práctica para la aplicación específica en concreto.
7) **Marco normativo de aplicación. –** En la última cláusula, se señala la información detallada para que una aplicación alcance el nivel objetivo de confianza o *Targeted Level of Trust* (TLT). Para ello, se utiliza un recurso autoritario que indica cómo hacerlo, y que se conoce como marco normativo de aplicación; el cual recoge los elementos, decisiones y resultados acumulados durante el ciclo de vida de la aplicación específica.

4.2.2.2. APLICACIÓN

Se incluye la arquitectura desarrollada en base a la información recopilada brevemente de las cláusulas de la normativa ISO 27034-3:2018:

1) **Alcance. –** Este trabajo de investigación se centra en definir en líneas generales una guía o procedimiento para implementar una arquitectura con una solución EDR que sirva para defender los equipos y sistemas de una organización. El alcance que se tiene en cuenta es el siguiente:
    - La guía o procedimiento será únicamente trasladable a equipos con sistemas operativos *Windows 10* (preferiblemente en su versión 20H2 o superior).
    - La guía o procedimiento tendrá en cuenta las soluciones a nivel local y la conexión con un equipo remoto centralizado que será tratado por el equipo de SOC.
    - La guía o procedimiento utilizará una batería de pruebas para probar sus funcionalidades definidas.
2) **Referencias normativas. –** Se detallan las referencias normativas que han sido empleadas para el desarrollo de la aplicación:
    - **MITRE ATT&CK. –** MITRE ATT&CK es un marco oficial que sirve como modelo y base para reconocer el comportamiento de amenazas existentes de una manera generalizada, mostrando las diferentes fases en el ciclo de vida y los recursos o plataformas que la amenazas suele atacar. Este marco proporciona además tácticas y técnicas que son de especial utilidad desde un punto ofensivo o defensivo (Trellix, 2022). Actualmente MITRE ATT&CK cuenta con tres vertientes: *ATT&CK For Enterprise, ATT&CK For Mobile* y *ATT&CK For ICS*; siendo la primera la utilizada en el desarrollo de esta guía.
    - **Arquitecture Design (or Development) Methodology for Information Technology (ADMIT). –** ADMIT es un marco normativo utilizado para el desarrollo de ciclos de vida de arquitecturas, sus fases y manejo de procesos de la arquitectura desarrollada. Puede ser usada junto a otros marcos normativos.
3) **Términos y definiciones. –** Seguidamente, se señalan los términos y definiciones necesarios para poder entender la solución planteada:
    - ***Organization Normative Framework,*** son esencialmente repositorios, matrices, guías o procedimientos de la compañía sobre los controles y procesos de seguridad de las aplicaciones.

- ***Application Normative Framework,*** es un subconjunto de la *Organization Normative Framework* que contiene información específica de una aplicación.
- ***Endpoint Detection and Response,*** se trata de una solución que funciona como detección y respuesta a amenazas dadas en equipos y sistemas. Integra recursos que combinan respuestas automatizadas basadas en reglas con capacidades analíticas.
- ***Security Operations Center,*** es una función centralizada perteneciente a una organización que combina procesos, tecnología y especialistas en el campo de la ciberseguridad defensiva para monitorizar y ofrecer mejoras de manera continua a la organización, y al mismo tiempo, realizar maniobras de prevención, detección, análisis y contrarrespuesta a amenazas emergentes de la manera más rápida posible.
- ***National Institute of Standards and Technology,*** es una organización fundada en 1901 que forma parte del departamento de comercio de los Estados Unidos. Actualmente, el *National Institute of Standards and Technology* se encarga de realizar trabajo desde tecnologías básicas hasta las más innovadoras; promoviendo desde innovación hasta métricas de distinto tipo. Podría decirse por lo tanto, que es uno de los organismos de referencia en los ámbitos que constituyen el ámbito abarcado (NIST, 2022).
- ***General Data Protection Regulation,*** es una regulación de derechos europeos sobre protección de datos y privacidad en la Unión Europea y el área económica europea. Incluye un conjunto de estándares, desarrollados para proporcionar a los ciudadanos europeos control sobre sus datos. Una organización que prometa el establecimiento de esta regulación en sus sistemas asegura el manejo legal y la protección adecuada de los datos sensibles, evitando uso malintencionado de éstos (StorMagic, 2021).
- ***Sistema de Gestión de la Seguridad de la Información,*** como su nombre indica, es un sistema de gestión documentado, consistente en un conjunto de controles de seguridad que protegen la confidencialidad, integridad y disponibilidad de sus activos frente a amenazas y vulnerabilidades.
- ***International Standards Organization,*** es una organización de desarrollos de estándares internacionales compuesta por representativos de organizaciones de estandarización nacional de los estados miembros (ISO, 2022).

- ***Tabla Responsible, Accountable, Consulted and Informed***, es un tipo de métrica empleada para asignar responsabilidades a los roles definidos en una actividad específica dentro de una organización.
- ***Elliptic Curve Cryptography,*** tipo de criptografía basada en el uso de algoritmos mediante curvas elípticas. Utiliza un par de claves pública y privada para cifrar o descifrar datos, y destaca frente a otros algoritmos de cifrado en su balance de rapidez y nivel de seguridad aportada (AVINetworks, 2022).
- ***Algoritmo de Rivest-Shamir-Adleman***, uno de los sistemas de cifrado para distendidos, el cual emplea cifrado asimétrico mediante el uso de un par de claves pública y privada pertenecientes a cada equipo. Dicha clave pública es accesible para el resto de los usuarios, y sirven para identificar de manera inequívoca a ese usuario. La clave privada es única de cada usuario. Se basa en el uso de números primos y el algoritmo Euclidiano (Encryption Consulting, 2022).
- ***Digital Signature Algorithm,*** sistema de cifrado el cual se basa en el concepto matemático de exponenciación modular y el problema del algoritmo discreto. Fue propuesto por el *National Institute Standards and Technology* para su uso en 1991, y se sigue empleando en la actualidad. Usa un par de claves pública y privada; y provee autenticación, integridad y no repudio de la información cifrada (Simplilearn, 2022).
- ***Secure Sockets Layer,*** es la tecnología estándar para establecer conexiones de Internet securizadas, y salvaguardar información sensible que está siendo enviada entre dos sistemas; previniendo que los ciberdelincuentes lean o modifiquen dicha información.
- ***Transport Layer Protocol,*** se trata de una versión actualizada de *Secure Sockets Layer*, la cual ofrece cifrado mediante *Elliptic Curve Cryptography,* el algoritmo de *Rivest-Shamir-Adleman* o *Digital Signature Algorithm*, tipos de cifrado más potentes que los que ofrece la tecnología anterior.
- ***Hypertext Transfer Protocol,*** protocolo estándar a nivel de aplicación usado para intercambiar información entre dos equipos a través de Internet (Britannica, 2022).

- ***Internet Protocol,*** protocolo o conjunto de reglas para enrutar y direccionar paquetes de datos para que puedan viajar a través de las redes y llegar al destino correcto.
- ***Windows Management Instrumentation,*** infraestructura de manejo de datos y operaciones basada en sistemas operativos *Windows.* Proporciona la capacidad de automatizar tareas y eventos mediante el uso de *scripts* y aplicaciones en equipos remotos (Windows Management Instrumentation, 2021).
- ***Domain Name System,*** servicio encargado de transformar los nombres de dominio en direcciones de *Internet Protocol*.
- ***Endpoint*,** es un equipo, sistema, servicio o recurso que se comunica y recibe datos a través de la red en la que se encuentra conectada. Ejemplos de esta definición son ordenadores, dispositivos móviles, *sockets* o servidores.
- ***Reglas YARA,*** son utilizadas como piezas de lenguaje de programación empleadas para encontrar ejemplos de *malware* en contenido de ficheros. Se basan en patrones y condiciones, de manera que si se cumplen; se devuelve un valor verdadero que identifica a dicho fichero como potencialmente peligroso (Rules, 2022).
- ***Sysmon,*** herramienta del paquete *Sysinternals*, es un servicio para sistemas operativos *Windows* que, una vez instalado en el equipo, permanece funcionando continuamente en segundo plano, recopilando información de los eventos que ocurren en el sistema en base a un fichero de configuración dado. *Sysmon* no ofrece protección o análisis, ni genera alertas o contramedidas por sí mismo; sino que más bien, se presenta en este proyecto como la herramienta encargada de la monitorización y recopilación en los *endpoints* de las arquitecturas.
- ***TheHive,*** plataforma de respuesta a incidentes de seguridad de código abierto diseñada para facilitar la labor de los grupos de SOC o cualquier otro practicante en seguridad defensiva; encargándose de los incidentes de seguridad que necesitan ser investigados y tener una contrarrespuesta rápidamente.
- ***Cortex*,** herramienta empleada para el análisis de observables que aparecen en *TheHive*, como son IP, *hashes*, nombres de dominio o URLs. Las funciones de análisis pueden ser automatizadas. El empleo de *Cortex* en conjunto con

TheHive, facilita enormemente las fases de contrarrespuesta, gracias a las funcionalidades de respuesta a eventos que da *Cortex* (TheHive-Project, 2022).

- ***Malware Information Sharing Platform,*** es una solución de código abierto que permita coleccionar, almacenar, distribuir y compartir indicadores y amenazas de incidentes de ciberseguridad y análisis de *malware* con otras organizaciones. Esta colaboración sirve de soporte en el día a día para la aportación de información entre especialistas de ciberseguridad de manera eficiente (Project, 2022).
- ***Security Incident Response Platform,*** plataforma de respuesta a incidentes de seguridad de código abierto y gratuita, constituida por *TheHive, Cortex y MISP*. Su diseño está enfocado en tratar con incidentes de seguridad que necesitan ser investigados y actuados con la mayor velocidad posible.
- ***Elasticsearch,*** es un motor de búsqueda y analítica distribuido, gratuito y de código abierto para todos los tipos de datos. Supone la base de la *ELK Stack*, y es ampliamente conocida por sus *Application Programming Interface* (API) sencillas, su velocidad mediante la utilización de índices invertidos, ingesta de datos, análisis; entre otros. Los ficheros que proporciona vienen dados en formato *Javascript Object Notation* (JSON). Incluye la capacidad de cooperar con una serie de agentes capaz de recopilar información concreta de los sistemas.
- ***Kibana,*** se trata de un *frontend* gratuito y de código abierto que es utilizado para visualizar los datos que son indexados en *Elasticsearch*.
- ***Beats,*** es una plataforma con agentes de datos con un propósito definido, consistente en el envío de datos de cientos o miles de datos a *Elasticsearch*. Concretamente en este proyecto, se utilizará *winlogbeats*, utilizado para monitorizar eventos del sistema de *Windows* (Burnham, 2018).
- ***ElastAlert,*** es un sistema simple para generar alertas de anomalías, picos u otros patrones de interés desde *Elasticsearch* (Yelp, 2020).
- ***Berkeley BD,*** se describe como una base de datos capaz de almacenar información en pares clave y valor. Se trata de un tipo de base de datos no relacional, extremadamente escalable, con un coste de despliegue muy bajo y sin pérdidas de datos o corrupción en sus bases. Además, es una base de datos especializada en sistemas de tiempo real, y en el manejo de una inmensa cantidad de hilos que controlen procesos que se encargan de sobrescribir su

base de datos. Esta base de datos se emplea en *TheHive* para el almacenamiento de las alertas que van llegando.

- ***MySQL,*** es un sistema de gestión de bases de datos de código abierto ampliamente utilizado actualmente. Maneja una base de datos relacional, y es muy rápida, escalable, confiable y fácil de usar. Será la base de datos empleada por *MISP* para el almacenamiento de los datos que vaya recibiendo.
- ***Lucene,*** es un motor de búsqueda de bases de datos, escrito enteramente en *Java*. Su tecnología es perfecta para aplicaciones que requieren búsqueda de texto completa, y especialmente para aquellas que son multiplataforma. Permite multi-indexado, distintos tipos de *queries*, ordenación por campos, es rápido y eficiente desde un punto de vista de memoria, entre otros. Por ello, es la herramienta perfecta para *TheHive*, donde llegarán continuamente enormes cantidades de datos que deben ser indexadas y ordenadas de la manera más eficiente posible.
- ***Docker,*** es una plataforma de código abierto que permite el despliegue de aplicaciones dentro de contenedores de *software*. La ventaja que tiene *Docker* a la instalación manual es que permite este despliegue separando la infraestructura desde la que se despliega respecto al contenedor; de manera que habilita, por ejemplo, obtener *software* funcional que por problemas de versiones de compatibilidad respecto al sistema operativo desde el que se trabaja, no podría ser posible. Supone una enorme ventaja a la hora de instalar arquitecturas completas, ya que ahorra los procesos de instalación individualmente donde todos los componentes; limitándolo únicamente a la configuración de un fichero *.yaml* de manera adecuada, y automatizando el proceso, desplegando dicha arquitectura en cuestión de minutos, sin problemas de compatibilidad o datos corruptos.
- ***Dockerstation,*** es una aplicación para desarrolladores con el objetivo de gestionar proyectos basados en *Docker* de la forma más simple posible. En lugar de realizar todo el proceso en *Docker* mediante la introducción de líneas de comando; *Dockstation* permite monitorizar, configurar y gestionar los contenedores dentro de una interfaz gráfica. Permite comprobar cuantos recursos de la máquina utilizan los contenedores por separado, que interfaces de red usan; entre otra información básica. Tiene compatibilidad con versiones anteriores de *Dockstation*, lo que permite seguir reutilizando proyectos en

versiones desfasadas. Además, permite el control de los puertos abiertos y cerrados en los distintos contenedores, lo cual puede ser de tremenda utilidad a la hora de configurar aplicaciones que interaccionan entre sí.

- ***Git,*** es un sistema de control de versiones distribuido gratuito y de código abierto diseñado para el manejo de proyecto pequeños o grandes. Se caracteriza por su rapidez y eficiencia.
- ***Github,*** servicio de *hosting* en Internet para subida de desarrollo de *software* y control de versiones mediante *Git*. Está provista de control de acceso, *bug tracking,* peticiones de funcionalidades para el *software*, manejo de tareas, integración continua y desarrollo de documentación para cada proyecto.
- ***P-EDR Arch,*** conjunto de tecnologías gratuitas y de *open-source* que conforman una arquitectura de ciberdefensa contra amenazas actuales. Para su funcionamiento principal, utiliza una solución EDR constituida principalmente por *Sysmon*, el fichero de configuración que produce el funcionamiento de éste, un script en *PowerShell* que opera mediante los manejos del procesos del sistema, una carpeta de cuarentena oculta a nivel de privilegios de *NT-System* y protegida por *System ACL*, reglas *YARA* y un agente de *Beats* conocido como *winlogbeats* que comparte la información generada por las alertas a una aplicación de *Elasticsearch* que se aloja en un equipo remoto. Por lo tanto, *P-EDR Arch* se compone de esta solución EDR, una implementación de *Elasticsearch, Kibana* y *Elasticalert* desplegados en un segundo equipo remoto, y la ejecución del conjunto de herramienta de *TheHive Project* que se encuentran en un tercer equipo.

4) **Términos abreviados. –** Se incluye a continuación una lista de los terminados abreviados; en el caso de que sean necesarios:

- EDR: *Endpoint Detection and Response*
- SOC: *Security Operations Center*
- MISP: *Malware Information Sharing Platform*
- RACI: *Responsible, Accountable, Consulted and Informed*
- NIST: *National Institute of Standards and Technology*
- TLP: *Traffic Light Protocol*
- SSL: *Secure Sockets Layer*
- TLS: *Transport Layer Protocol*

- HTTP: *Hypertext Transfer Protocol*
- IP: *Internet Protocol*
- WMI: *Windows Management Instrumentation*
- ECC: *Elliptic Curve Cryptography*
- RSA: *Rivest-Shamir-Adleman*
- DSA: *Digital Signature Algorithm*
- P-EDR Arch: *Personal-Endpoint Detection and Response Architecture*
- DNS: *Domain Name System*
- SIRP: *Security Incident Response Platform*

5) Proceso de gestión de la seguridad de las aplicaciones. – Los conceptos teóricos que definen el proceso de gestión de la seguridad están basados en las siguientes secciones:
   - Identificar los requerimientos y ámbito de la aplicación
   - Indicar los riesgos de seguridad de la aplicación
   - Crear y mantener el marco normativo de la aplicación
   - Aprovisionar y manejar la aplicación
   - Auditar la seguridad de la aplicación

   Las primeras tres secciones están enfocadas en identificar y seleccionar controles de seguridad adecuados para la aplicación. Teniendo en cuenta que la seguridad es una característica fundamental para una aplicación de este tipo; definirla de manera adecuada en los primeros pasos de planteamiento de adquisición o desarrollo de ésta es crucial. Esta primera definición les permitirá a los equipos la identificación de piezas claves o deseables, y del mismo modo, permite la integración de la seguridad minimizando la disrupción de planes de la organización.

   Las dos últimas secciones se enfocan en la implementación y verificación de los controles de seguridad elegidos.

   Para cumplir todo lo marcado anteriormente, una aplicación segura debería en términos generales:

   - Tener asignada un nivel de confianza objetivo
   - Cualquier componente de seguridad o proceso usado en la aplicación debería ser seleccionado siguiendo el *ONF* de la compañía
   - Todos los controles de seguridad de la aplicación seleccionados dentro del nivel de confianza objetivo deberían ser implementados, verificados y auditadosç

Además de estos términos generales, es necesario que para la seguridad de la organización, la aplicación:

- Debe tener asignados unos roles y responsables que se encarguen del manejo y control de los procesos que la propia aplicación emplea
- Debe señalar la relación existente entre el proceso de gestión de seguridad de la aplicación con el *ONF* de la organización, con el objetivo principal de crear y mantener el *ANF* de la aplicación, y en consecuencia, el esquema general de la compañía
- Debe emplear herramientas aprobadas por la organización
- Debe tener marcado un nivel de confianza objetivo; donde se señale adicionalmente, el nivel de confianza actual de la aplicación
- Debe incluir un documento que indique el impacto de la incorporación de éste (y por lo tanto de la aplicación) en un nuevo ámbito

Se incluye una ilustración que generaliza todo lo detallado textualmente.

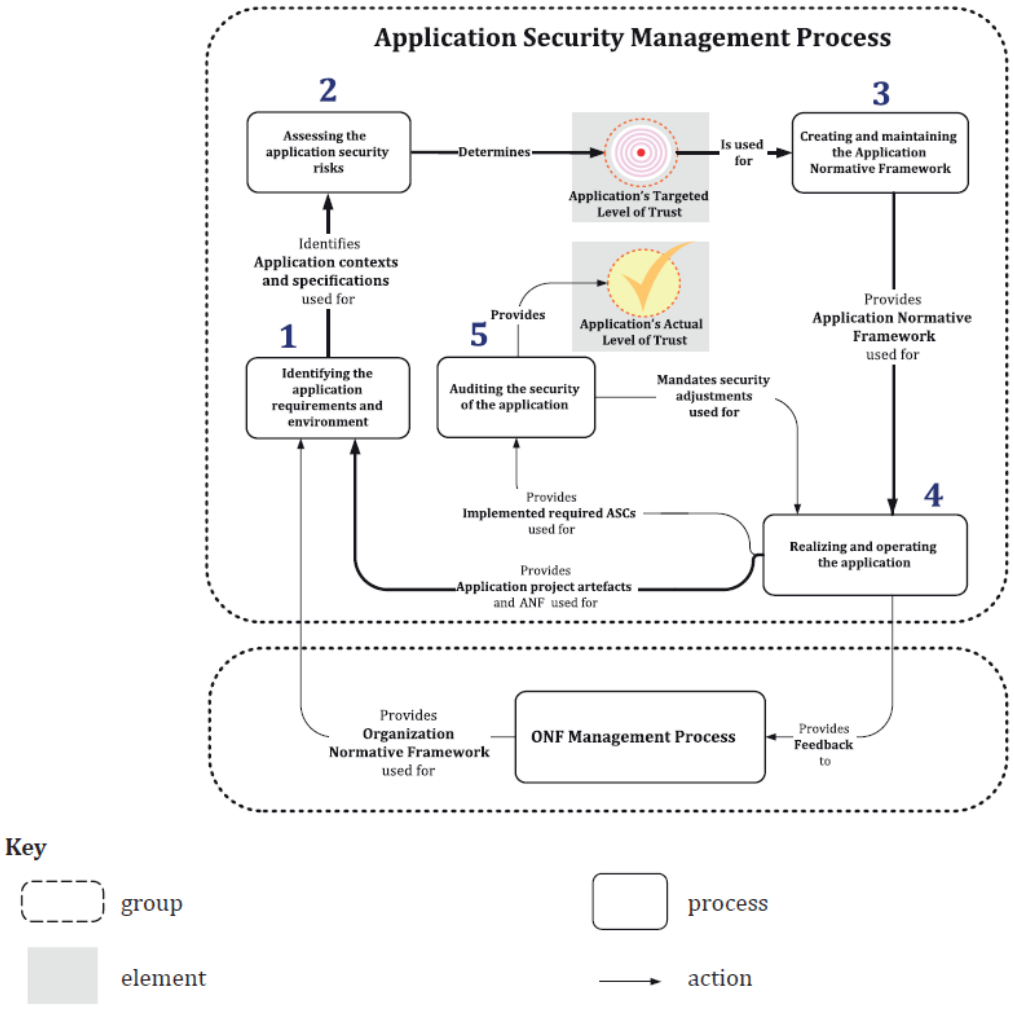

Ilustración 5. Esquema de pasos a seguir en la implementación del proceso de gestión de seguridad de la aplicación

6) **Pasos en el proceso de gestión de la seguridad de las aplicaciones. –** Teniendo en cuenta el marco teórico definido en el punto anterior, se procede a la cumplimentación de estos en base a la aplicación específica.

   Primeramente, para **la identificación de los requerimientos y ámbito de la aplicación**, se debe incluir:

   - Actores, es decir, definición de roles, responsabilidades y cualificaciones.
   - Identificación de las especificaciones de seguridad de la organización para la aplicación, es decir; especificación de los requerimientos para el *software* envuelto, políticas como requerimiento mínimo para contraseñas, documentos normativos y de cumplimiento, objetivos organizacionales y comerciales, o diagramas de la arquitectura.

- Los flujos que presenta la aplicación, es decir, como se transmite la información dentro de la propia aplicación o con componentes externos.
- Establecimiento del contexto para el ámbito tecnológico, de negocio y regulatorio.

Los roles considerados para la consecuente aplicación son:

- **Comité en base al *ONF***, encargados de asegurarse que los controles que se establecen en base a esta aplicación son los adecuados. Esto incluye seguimiento de guías y procedimientos, asignación de roles si es necesario; entre otros.
- **Gerente de proyectos,** individuo encargado de la configurabilidad en relación con la aplicación (añadir o quitar nuevas funcionalidades, establecer los distintos elementos de una determinada manera; entre otros).
- **Equipo de *SOC*,** grupo constituido por una serie de especialistas en ciberseguridad, concretamente en ciberdefensa, encargados de analizar y ofrecer una contrarrespuesta en base a una amenaza dada.
- **Equipo de forense,** grupo determinado por una serie de especialistas en ciberseguridad, concretamente en análisis forense, encargados de analizar y desarrollar una documentación en base al comportamiento de una amenaza. Esta fase se dará post contrarrespuesta.
- **Responsable principal en el equipo de *SOC*,** será la persona que liderará las decisiones finales en el equipo de *SOC*.
- **Auditores,** personas encargadas de comprobar que la seguridad acerca de la aplicación específica es adecuada para satisfacer la totalidad de las actividades proporcionada por la organización.
- **Usuario final**, son todos aquellos empleados de una organización que utilizan equipos o sistemas donde la aplicación se encuentra en uso.

En base a estos roles, se desarrolla una tabla de realización de procesos; la cual asignará responsable en función de la tabla *RACI*:

Código	Responsabilidad
R	*Responsible* o responsable parcial sobre la realización de la actividad

A	*Accountable* o responsable total sobre la realización de la actividad
C	*Consulted* o consultado sobre la realización de la actividad
I	*Informed* o informado sobre la realización de la actividad

Tabla 1. Métrica RACI

Realización de actividad	Comité ONF	Gerente de proyectos	Equipo de SOC	Equipo de forense	Responsable principal SOC	Auditores	Usuario final
1) Identificación de responsable en el equipo de SOC	A/R		I				
2) Implementación del proceso de gestión de seguridad de la aplicación		R	R		A		
3) Identificación de las necesidades de la organización en base a las características de la aplicación	A	I	C		A/R		
4) Identificación de los requerimientos de la aplicación	C	I	R		A/R		
5) Identificación del contexto de negocio en base a la aplicación, incluyendo procesos, actores y requerimientos de negocio requeridos o afectados por la aplicación	A	I			R		
6) Identificar el contexto regulatorio de la aplicación	A	C/I			R		
7) Identificar el contexto	C	I	R		A/R		

tecnológico de la aplicación							
8) Validar, verificar e integrar los resultados de toda estas actividades en el *ANF* preliminar	C	C/I	R		A/R		

Tabla 2. Realización de actividades en relación con la implementación de la aplicación

Verificación de la actividad	Comité ONF	Gerente de proyectos	Equipo de SOC	Equipo de forense	Responsable principal SOC	Auditores	Usuario final
1) Verificar que existe un responsable asignado y ejerce sus funciones correctamente	A		C		I	R	
2) Recopilar contextos de seguridad de la aplicación (de negocio, regulatorio y tecnológico)		C	C	I	C	A/R	
3) Recopilar especificaciones y descripciones de procesos referidos a la aplicación		C	C	I	C	A/R	
4) Recopilar el inventario de los procesos que definen la aplicación y los diagramas de flujo de la información		C	C	I	C	A/R	
5) Recopilar evidencias de que los roles y responsabilidades fueron detallados para cada actor		C	C	I	C	A/R	

6) Recopilar especificaciones de la aplicación, documentos referidos y diagramas de la arquitectura		C	C	I	C	A/R	

Tabla 3. Verificación de actividades en relación con la implementación de la aplicación

En lo referente a las especificaciones organizacionales de seguridad de la aplicación se tendrá en cuenta:

- Especificaciones de requerimiento para el *software* que unifica la aplicación.
- Requerimiento que engloban a las políticas de seguridad, como mínimo requisito para contraseñas.
- Documentos regulatorios y de cumplimiento.
- Objetivos organizacionales y de negocio respecto a la aplicación.
- Diagrama/s de la arquitectura.

Se incluyen a continuación las especificaciones de requerimiento para el *software*:

- La solución EDR que forma parte de la aplicación, está destinada para funcionar en equipos con sistema operativo *Windows 10*, exclusivamente.
- La habilitación del protocolo HTTP así como del *Native Elasticsearch binary protocol*; es fundamental para el acceso y compartición de la información entre *endpoint* y motor de búsqueda e indexación, y por transitividad, con el equipo de SOC. El uso de SSL/TLS aunque no es obligatorio, es recomendado para mayor securización (Digicert, 2022).
- La utilización de *Traffic Light Protocol* (TLP) es obligatoria para el uso de los observables recogidos en *TheHive*.
- La recogida de ficheros maliciosos se realiza almacenando el *hash* que identifica a dicho fichero en una carpeta oculta incluso a nivel de administrador; únicamente es visible con el usuario *NT-System*. La habilitación de dicho usuario es obligatoria si se quiere acceder a esta información.
- Uso del protocolo *SSH* para configuración de los *endpoints* por parte de los equipos especializados de la manera óptima.

Entre las políticas y procedimientos a utilizar por la organización se obliga a establecer lo siguiente:

- **Política de manejo e implantación de contraseñas seguras actualizada**. Es necesario usuario y contraseña para acceder a *TheHive, Cortex y MISP*. Del mismo modo, se recomienda establecer contraseñas seguras en los *endpoints*, para evitar acceso no autorizado a los equipos físicamente.
- **Procedimiento de lista blanca para conexiones remotas con los *endpoints***. De esta manera, se evitan descargas de sitios web inusuales, o inyecciones en memoria dinámica de procesos inseguros hacia estos equipos. Este procedimiento puede ser trasladado a las propias reglas de inclusión de la solución EDR, facilitando la tarea de trasladarlo de un ámbito teórico a práctico.
- **Política de roles y responsabilidades respecto a la aplicación anterior**, en base a las tablas anteriormente citadas.
- **Política de auditorías internas**, o en su defecto, una **política sobre prestación de servicios por el auditor externo**; para mantener un seguimiento correcto del estado y seguridad de la aplicación en todo momento.
- **Política de clasificación y manejo de la información**, en base a los ficheros que son monitorizados y controlados por parte de la solución EDR en los *endpoints* específicos.
- **Política de gestión de *logs* o registros,** con el objetivo de comprobar en todo momento la información que está siendo monitorizad, y poder seguir en todo momento las trazas de posibles acciones inusuales ejecutadas en los *endpoints*. Sirve como mecanismo de análisis para el equipo de SOC y el equipo forense.

Respecto al cumplimiento normativo y de regulación se tiene en cuenta:

- ***Data compliance* o cumplimiento de datos**; cumplimiento de regulación que garantiza que los datos que posee una organización están ordenados, almacenados y administrados de manera correcta. El objetivo principal es que estos datos no se pierdan o sean robados y empleados de manera inadecuada. Un buen ejemplo es la GDPR.
- ***Cybersecurity compliance*** o cumplimiento de ciberseguridad; el cual implica el cumplimiento de normas para proteger la seguridad informática de las organizaciones. Un buen ejemplo es la *International Standards Organization* (ISO) 27001, que describe los requisitos necesarios para establecer, implantar, mantener y mejorar un sistema de gestión de la seguridad de la información (SGSI).

Los objetivos organizacionales y de negocios posibles para la implantación y desarrollo de esta aplicación son:

- Mejora en la protección de los sistemas y equipos de la organización. Esta mejora también se aplica a los activos e información tratada dentro de éstos.
- Ofrecer niveles de seguridad más robustos para los clientes que soliciten servicios a la organización.
- Desarrollo de una arquitectura mejor provista en materia de ciberdefensa.
- Implantación de tecnologías gratuitas y de código abierto para la defensa de los sistemas.
- Análisis mejorado y más detallado de las amenazas que afectan a la organización.
- Compartición *online* de los documentos desarrollados por los equipos de SOC y forense en base a vulnerabilidades localizadas en los sistemas de la organización.
- Monitorización avanzada de los equipos finales usados por los empleados de la organización.
- Establecimiento de una solución EDR basada enteramente en la ISO 27034-3:2018, lo que otorga un respaldo adicional en procesos de auditoría.

En cuanto a los diagramas que representan el conjunto de la aplicación, son divisibles en dos arquitecturas: una arquitectura referente a la solución EDR establecida en los *endpoints* o equipos finales utilizados por los empleados de la empresa, y una arquitectura que engloba esta solución EDR junto al resto de elementos que la componen. Ambas serán presentadas apropiadamente en la solución.

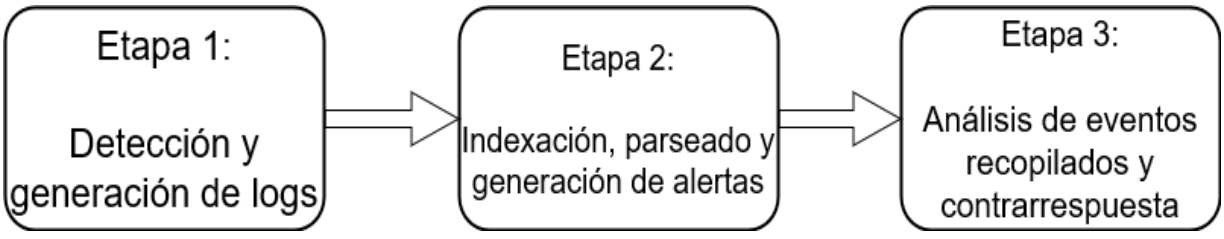

Ilustración 6. Flujo de información genérico realizado

Sobre los **detalles del flujo de información** relacionados con la aplicación; pueden tomarse los diagramas de las arquitecturas presentados anteriormente como referencia para entender las etapas que van a ser descritas y lo que ocurre:

- **Etapa 1. –** Una vez instalada la arquitectura en su completitud, el objetivo es que la información de los eventos producidos por *Sysmon* de los *endpoints* que tienen instaladas las dependencias que forman parte de la solución EDR, lleguen correctamente al equipo de SOC. En una primera instancia, la información que se va a recoger viene en base del fichero de configuración de reglas de *Sysmon*. Concretamente, en la versión de *Sysmon* empleada; existen un total de 27 eventos diferentes que pueden registrarse, todos identificados por un ID que los hace inequívocos (Malware Archaeology, 2019) (Drysdale, A Sysmon Event ID Breakdown, 2021). Los eventos en concreto junto a su ID, vienen registrados en la siguiente tabla:

ID	Nombre del evento	Descripción breve
1	*Process Creation*	Proporciona información extendida sobre un proceso recién creado.
2	*A process changed a file creation time*	Proporciona información cuando un proceso cambiar el tiempo de creación de un fichero.
3	*Network connection*	Proporciona información de registros de conexiones TCP/UDP en máquina.
4	*Sysmon service state changed*	Proporciona información del estado del servicio *Sysmon* (iniciado o detenido).
5	*Process terminated*	Proporciona información sobre cuando un evento termina.
6	*Driver loaded*	Proporciona información cuando un *driver* es cargado en el sistema.
7	*Image loaded*	Proporciona información cuando un módulo es cargado en un proceso en específico.
8	*CreateRemoteThread*	Proporciona información cuando un proceso crea un hilo en otro proceso.
9	*RawAccessThread*	Proporciona información cuando un proceso realiza operaciones de lectura de disco mediante acceso a carpeta padre (‘\\.\’).

10	*ProcessAccess*	Proporciona información cuando un proceso abre otro proceso en funcionamiento, y solicita información de éste o lectura y escritura del espacio de direcciones del proceso objetivo.
11	*FileCreate*	Proporciona información cuando un fichero es creado, descargado o sobreescrito.
12	*RegistryEvent (Object create and delete)*	Proporciona información de un registro del sistema cuando es creado o eliminado.
13	*RegistryEvent (Value Set)*	Proporciona información de un registro del sistema cuando se modifican los valores de dicho registro.
14	*RegistryEvent (Key and Value Rename)*	Proporciona información de un registro del sistema cuando se modifica el nombre de la clave o el nombre del valor que lo identifica.
15	*FileCreateStreamHash*	Proporciona información de un fichero antes de que sea descargado, comprobando el *hash* de los contenidos de ese fichero cuando se está descargando en el sistema.
16	*ServiceConfigurationChange*	Proporciona información cuando la configuración del fichero *Sysmon* es cambiada.
17	*PipeEvent (Pipe Created)*	Proporciona información cuando una conexión vía *pipe* es creada.
18	*PipeEvent (Pipe Connected)*	Proporciona información cuando una conexión vía *pipe* es realizada entre cliente y servidor.
19	*WmiEvent (WmiEventFilter activity detected)*	Proporciona información cuando se detecta el registro de un evento *Windows Management Instrumentation* (WMI) en el sistema.
20	*WmiEvent (WmiEventConsumer activity detected)*	Proporciona información cuando se detecta el registro de un consumidor WMI en el sistema.
21	*WmiEvent (WmiEventConsumerToFilter activity detected)*	Proporciona información cuando un consumidor WMI se enlaza a un filtro específico.

22	*DNSEvent (DNSQuery)*	Proporciona información cuando un proceso ejecuta una consulta DNS, independientemente de su resultado.
23	*FileDelete (File Delete archived)*	Proporciona información cuando se elimina un fichero, y es guardado en una carpeta de cuarentena oculta solo visible y bajo privilegios de *NT-System*. Está carpeta está protegida por *System ACL*.
24	*ClipboardChange (New content in the clipboard)*	Proporciona información cuando el contenido del portapapeles cambia.
25	*ProcessTampering (Process image change)*	Proporciona información cuando se detectan técnicas de ocultación inusuales (Drysdale, A Sysmon Event ID Breakdown – Now with Event ID 25!!, 2021).
26	*FileDeleteDetected (File Delete logged)*	Proporciona información cuando se elimina un fichero, y no se guarda.
255	*Error*	Proporciona información cuando se produce un error en el propio servicio de *Sysmon.*

Tabla 4. Eventos registrados por Sysmon

Para la recogida de la datos de manera adecuada, *Sysmon* hace distinciones sobre qué clase de información recoge en base al ID del evento que está tratando. Estos eventos se verán en la solución al problema.

Junto a toda esta información se ha incluido para cada evento: ID de técnica que lo referencia según MITRE ATT&CK, nombre de la técnica, táctica, alerta generada y banderas que señalan la respuesta que se realiza; todo esto en un campo estructurado denominado '*Message'*. Toda esta información almacenada en *logs* es registrada en formato *xml*.

Este flujo de información está en constante monitorización por el *script* en *Powershell* desarrollado que proporciona respuesta, y por el agente *winlogbeat* que transportará la información de las alertas al equipo de SOC.

Se destaca además, que de manera obligada el *script* generará en el *frontend* del usuario final, una alerta visual que incluirá parte de la información recopilada; siendo concretamente esta información el campo '*Message'* sin las banderas, *'UtcTime'* modificada al tiempo local del sistema (Stackoverflow, How to convert powershell UTC

datetime object to EST, 2019) (Wheeler, Vasin, & Wilson, 2022), *'ProcessId', 'User'* e información mínima extra en función del evento.

- **Etapa 2.** – La información transmitida desde *winlogbeat* hasta *Elasticsearch* lo hace utilizando el protocolo TCP. Una vez la información llega al motor de búsqueda, es indexada en formato *JSON*, formato empleado por *Elasticsearch* para el almacenamiento de información, donde puede ser visualizada desde un navegador web mediante *Kibana*, el cual está monitorizando continuamente *Elasticsearch*. Al mismo tiempo, existe una monitorización continua por parte de *Elastalert* sobre *Elasticsearch*, el cual generará una alerta cada vez que existan actualizaciones de uno de los indexadores asociados a *Elasticsearch*, siendo en este caso *winlogbeat.**- (OpenSecure, 2021).

  Antes de ser enviada a *TheHive*, nuevamente, mediante el protocolo TCP, la información puede ser *parseada* en dos fases previo a la generación de la alerta. Primeramente, mediante el uso de las *Ingest pipelines* las cuales permiten segmentar información que se encuentra comprimida en un solo campo, como es el caso del campo '*Message*' anteriormente mencionado (Official Elastic Community, 2021). La segunda fase de *parseo* viene dada por el propio *Elastalert*, el cual permite elegir la información exacta que quiere enviarse a *TheHive*, añadir etiquetas en base a esta información y además, convertir parte de esta información a tipo de observables que se desee.
- **Etapa 3. –** En lo referente al propio sistema que compone *TheHive Project*, la comunicación entre las tres herramientas es realizada mediante el uso de los protocolos TCP y HTTP (o HTTPS, en su defecto). El flujo de información llegará completamente preparado desde *Elasticsearch*, para ser tratado de manera adecuada por el equipo de SOC. Además; como la información se tratará a modo de observables, podrán lanzarse las utilidades propias de *Cortex* o *MISP* para generar más información de la que ya se aporta de base; profundizando más en una amenaza y generando una contrarrespuesta óptima. Hay que tener en cuenta que, alguna de las utilidades empleadas por *Cortex* hace uso de herramientas de terceros, como *Virustotal* para el análisis de *hashes*, lo que significa que el uso de estas utilidades provoca que parte de la información sea compartida. Del mismo modo, si se desea generar una compartición de la

información mediante *MISP*, cualquier organización o individuo que se suscriba, tendrá acceso a la información que se haya decidido divulgar.

Por último, se establece el ámbito de la aplicación en función de los contextos tecnológico, de negocio y regulatorio. Se considera que, la explicación realizada a lo largo de los puntos anteriores muestra de una manera bastante detallada estos contextos; así que se procede a un resumen breve para uso genérico:

- **Contexto tecnológico:** En lo referente al contexto tecnológico, la aplicación aplica una función de monitorización, prevención, detección y contrarrespuesta a amenazas que se producen en los distintos equipos que forman la organización. Supone un incremento para la ciberdefensa de los sistemas, así como una mejoría en las herramientas proporcionadas a los empleados que actúan en equipos de usuarios finales y para el equipo de SOC, frente a las amenazas y vulnerabilidades encontradas.
- **Contexto de negocio:** En lo referente al contexto de negocio, la aplicación proporciona una mayor seguridad para la información de los clientes, suponiendo una reducción de los riesgos producidos en materia de ciberseguridad, lo que supone un decremento dado por impactos económicos posibles en lo relativo a la protección de los datos de los clientes. Consecuentemente, en un tiempo extendido, supone un incremento reputacional para la compañía en el ámbito de la ciberseguridad.
- **Contexto regulatorio:** En lo referente al contexto regulatorio, la incorporación de la aplicación proporciona unas bases más sólidas frente al seguimiento y utilización de marcos normativos en ámbito de ciberseguridad, al basarse la implementación de ésta en las cláusulas especificadas en la ISO-27034-3:2018. Como consecuencia, también se sigue el cumplimiento normativo de datos (*data* compliance) y el cumplimiento normativo de ciberseguridad (*cybersecurity* compliance); así como las guías, políticas y procedimientos seguidos por la organización, y que pueden ser avalados en un proceso de auditoría.

En la siguiente fase, indicar los riesgos de seguridad en la aplicación, se tendrán en cuenta:

- Roles (y responsabilidades) que participan en la fase de realización y verificación de actividades que indiquen los riesgos de seguridad.
- Identificación de riesgos generales y específicos
- Evaluación de métrica para riesgos.

Al igual que en la etapa previa, se usará RACI para indicar el tipo de responsabilidad por rol definido:

Realización de actividad	Comité ONF	Gerente de proyectos	Equipo de SOC	Equipo de forense	Responsable principal SOC	Auditores	Usuario final
1) Identificar y evaluar riesgos de seguridad llevados por la aplicación	C		C	C	A/R		
2) Identificar y evaluar la medida en la que la aplicación ha resuelto los riesgos de seguridad previamente identificados.	C		R	I	A		
3) Identificar los requerimientos de seguridad para garantizar la seguridad mínima para la aplicación	C		C	C	A/R		
4) Determinar el nivel de confianza objetivo para la aplicación, una vez identificados todos los requerimientos de seguridad	C		C	C	A/R		
5) Validar y aprobar el nivel de seguridad objetivo establecido	I		R	I	A		
6) Recopilar la información producida por el análisis de riesgos	C		C	A	A/R		

7) Actualizar los contenidos del ANF	A		R	I	I		
8) Actualizar los contenidos del ONF	A/R		C	I	C		
9) Hacer que la información sea accesible para los involucrados	I		I	A/R	R		

Tabla 5. Realización de actividades en relación con los riesgos de la aplicación

Verificación de la actividad	Comité ONF	Gerente de proyectos	Equipo de SOC	Equipo de forense	Responsable principal SOC	Auditores	Usuario final
1) Recopilar los análisis de riesgos de seguridad de la aplicación en función de la entrada y los resultados	I	C	R	C	A/R		
2) Verificar que los riesgos de seguridad de la aplicación fueron analizados	I	C	R	C	A/R		
3) Verificar que los riesgos de seguridad de la aplicación fueron evaluados adecuadamente según el conjunto de controles de seguridad de la aplicación	I	C	R	C	A/R		
4) Verificar que el nivel de confianza objetivo para la aplicación fue definido y aprobado	R	I	C	I	A/R		
5) Verificar que el responsable principal ha aceptado los riesgos residuales	A	I	C	I	R		

asociados con la aplicación							

Tabla 6. Verificación de actividades de los riesgos de la aplicación

Los riesgos que se han identificado respecto a la aplicación se dividen en dos: generales y específicos. Los generales son aquellos que se identifican mayoritariamente en la fase de implementación o desarrollo de la aplicación mientras que, los específicos, son aquellos detectados en la fase de realización de las actividades en relación con los riesgos de la aplicación.

- **Riesgos generales:**
  - La aplicación se subdivide según sus secciones en aplicación de escritorio o aplicación web. Esto supone que hay que abarcar un abanico de vulnerabilidades bastante ingente en comparación de si su implementación fuera exclusivamente unilateral a una de las dos partes.
  - La aplicación es utilizada únicamente por usuarios internos de la organización.
  - La aplicación trata con cualquier clase de información que se recoja en los equipos de los usuarios finales donde se implementa la solución EDR (con la posibilidad de incluir datos personales, de clientes; entre otros).
  - La aplicación no está unificada en un único equipo, sino que es producto de la interacción de varios sistemas independientes entre sí.
- **Riesgos específicos:**
  - La aplicación cuenta con métodos de autenticación y protección de credenciales.
  - La autenticación por parte de la aplicación se realiza consultando en una base de datos.
  - La aplicación permite la división en roles, dando flexibilidad en cuanto tareas autoritativas.
  - La aplicación no está protegida contra ataques de enumeración de usuarios.
  - La aplicación no cuenta con funcionalidad de gestión de tamaño de *logs* o estructuras de datos de almacenamiento similares.
  - La aplicación es segura frente a modificaciones de ficheros de configuración clave de los elementos que componen la solución en los equipos de usuario final.

Los niveles de impacto, junto a los niveles de probabilidad, servirán de métrica para medir el nivel de riesgo. Estos niveles de impacto se han ajustado al esquema seguido por el *Centro Criptológico Nacional* (CCN), y que implementan en las CCN-STIC-817 (gestión de ciberincidentes) (CCN, 2020); pero por supuesto, es una métrica que puede variar en función de las necesidades y localización de la organización.

ID de impacto	Nivel de impacto	Descripción del impacto
1	BAJO	-Afecta a los sistemas de la organización. -Interrupción de la prestación del servicio. -El ciberincidente precisa para resolverse menos de 1 jornada-persona. -Impacto económico entre el 0,0001% y el 0,001% del Producto Interior Bruto (PIB) actual. -Extensión geográfica superior a 1 Comunidad Autónoma de España (CC.AA). -Daños reputacionales puntuales, sin eco mediático.
2	MEDIO	-Afecta a más del 20% de los sistemas de la organización. -Interrupción en la prestación del servicio superior al 5% de usuarios. -El ciberincidente precisa para resolverse entre 1 y 5 jornadas-persona. -Impacto económico entre el 0,001% y el 0,003% del PIB actual. -Extensión geográfica superior a 2 CC.AA. -Daños reputacionales apreciables, con eco mediático.
3	ALTO	-Afecta a más del 50% de los sistemas de la organización. -Interrupción en la prestación del servicio superior a 1 hora y superior al 10% de los usuarios. -El ciberincidente precisa para resolverse entre 5 y 30 jornadas-persona. -Impacto económico entre el 0,03% y el 0,07% del PIB actual. -Extensión geográfica superior a 3 CC.AA. -Daños reputacionales de difícil reparación, con eco mediático y afectando a la reputación de terceros.

4	MUY ALTO	-Afecta a la seguridad ciudadana con potencial peligro para bienes materiales. -Afecta apreciablemente a actividades oficiales o misiones en el extranjero. -Afecta a un servicio esencial. -Afecta a sistemas clasificados como reservado. -Afecta a más del 75% de los sistemas de la organización. -Interrupción en la prestación del servicio superior a 8 horas y superior al 35% de los usuarios. -El ciberincidente precisa para resolverse entre 30 y 100 jornadas-persona. -Impacto económico entre el 0,07% y el 0,1% del PIB actual. -Extensión geográfica superior a 4 CC.AA. -Daños reputacionales a la imagen del país. -Daños reputacionales elevados y cobertura continua en medios de comunicación nacionales.
5	CRÍTICO	-Afecta apreciablemente a la Seguridad Nacional. -Afecta a la seguridad ciudadana, con potencial peligro de la vida de las personas. -Afecta a una infraestructura crítica. -Afecta a sistemas clasificados como secreto. -Afecta a más del 90% de los sistemas de la organización. -Interrupción en la prestación del servicio superior a 24 horas y superior al 50% de los usuarios. -El ciberincidente precisa para resolverse más de 100 jornadas-persona. -Impacto económico superior al 0,1% del PIB actual. -Extensión geográfica supranacional. -Daños reputacionales muy elevados y cobertura continua en medios de comunicación internacionales.

Tabla 7. Niveles de impacto para la aplicación

Los niveles de probabilidad se marcan tomando como referencia la CCN-STIC-470F1, es decir aquella métrica utilizada para definir la probabilidad en la herramienta PILAR (CCN, 2013), pero nuevamente este sistema de metraje es flexible en función de las necesidades y territorio de la organización que desee emplear esta aplicación:

ID de probabilidad	Nivel de probabilidad	Frecuencia
1	Muy rara	Su aparición es un hito inusual (0,01)
2	Poco probable	Puede presentarse una vez anualmente (0,1)
3	Posible	Se presenta de manera trimestral, al menos una vez (1)
4	Muy alta	Se presenta mensualmente, al menos una vez (10)
5	Casi seguro	Se presenta mensualmente, al menos dos veces (100)

Tabla 8. Niveles de probabilidad para la aplicación

Por lo tanto, con la utilización de estas dos tablas puede obtenerse el nivel de riesgo asociado y representarse mediante la matriz de riesgo. La fórmula que se sigue para este cálculo es:

Riesgo = Nivel de impacto x Nivel de probabilidad

El valor numérico asociado a los niveles de impacto debe ser categorizado por la propia organización, e incluido en la evaluación del plan de gestión de riesgos de la misma. Así mismo, se presenta un ejemplo a modo de guía de una supuesta matriz de riesgos (Praxis, 2019) parametrizada siguiendo la métrica *Red, Ambar, Green* (RAG) (Praxis, 2019) obtenida mediante el cálculo anterior:

		Amenazas					Oportunidades				
Probabilidad	0.90	0.05	0.09	0.18	0.36	0.72	0.72	0.36	0.18	0.09	0.05
	0.70	0.04	0.07	0.14	0.28	0.56	0.56	0.28	0.14	0.07	0.04
	0.50	0.03	0.05	0.10	0.20	0.40	0.40	0.20	0.10	0.05	0.03
	0.30	0.02	0.03	0.06	0.12	0.24	0.24	0.12	0.06	0.03	0.02
	0.10	0.01	0.01	0.02	0.04	0.08	0.08	0.04	0.02	0.01	0.01
		0.05	0.10	0.20	0.40	0.80	0.80	0.40	0.20	0.10	0.05
		Impacto									

Ilustración 7. Matriz de riesgos (impacto x probabilidad)

Se incluyen una serie de requerimientos de seguridad aplicados en torno al uso de la aplicación que favorecerán en la disminución de la probabilidad y el impacto que pueda producirse en una corporación que emplee esta arquitectura:

- Segmentación de la arquitectura tal y como se representa en el diagrama. Esto es debido a que, si se instalan todas las dependencias en una misma máquina, puede dar lugar a errores graves de configuración. La arquitectura está especialmente diseñada para que los componentes genéricos sean establecidos de esta manera. Si quiere habilitarse este EDR en equipos personales (de casa); se debería comprar al menos un par de sistemas adicionales o establecer máquinas virtuales con conexión en modo "*bridge*" con el *host*. En su defecto, el uso de contenedores de *Docker* también es buena opción.
- Proporcionar suficiente espacio de almacenamiento, y/o realizar *backups* frecuentes (al menos, una copia completa y dos copias incrementales cada mes) en los sistemas donde se implemente el ELK reducido y *TheHive Project*. Esto es debido a que el tamaño de los ficheros *log* y las bases de datos de éstos puede incrementar exponencialmente en función del número de equipos *endpoint* que tenga instalada la solución EDR y transitivamente, el envío de datos a *Elasticsearch* y de alertas a *TheHive*. Ignorar este requerimiento puede resultar en llenado de los sistemas y pérdida de datos de amenazas producidas en los límites de la organización.
- Se recomienda la habilitación del protocolo HTTPS en el ELK reducido, así como en *TheHive Project*, y a ser posible, el acceso con usuario y contraseña en todas las dependencias relacionadas. Del mismo modo, se recomienda securizar todas las conexiones con el protocolo SSL/TLS. Este paso ayuda a reducir la cantidad de accesos no autenticados y no autorizados por parte de actores maliciosos.
- Se recomienda utilizar contraseñas robustas en los paneles de acceso de las distintas dependencias. Una contraseña robusta debería incluir entre 8 y 24 caracteres, minúsculas, mayúsculas, números y caracteres especiales. Además, debería cambiarse con frecuencia (al menos, mensualmente) y no hacer referencia a información personal o relacionada con el usuario que la utiliza, ya que son principales vectores de ataque utilizados por los ciberdelincuentes.
- El acceso al sistema donde se encuentre alojado *TheHive Project,* debería ser único y exclusivo al equipo de SOC. Para ello, se recomienda crear usuarios y contraseñas específicos para cada uno de los integrantes,

establecer direcciones IPs estáticas en los equipos de dicha subred e implantar una lista blanca de acceso a las dependencias que solo incluya dichas direcciones IPs. Establecer esta medida para el servidor de ELK reducido no es obligatorio, pero sí recomendado.

- El sistema que aloje la información del ELK reducido debe estar vigilada en todo momento de manera física. El acceso no autorizado mediante consola o parecidos puede llevar a la extracción de cantidades de información ingentes por parte de un actor malicioso que, posteriormente, puede analizar y utilizar para realizar explotaciones frente a las vulnerabilidades que presenten los distintos sistemas del dominio.
- El tratamiento de la información que se aloje en los equipos de usuario final debe regirse a las políticas, guías, procedimientos y normativa establecidos por la organización. Esta aplicación no es solo exclusiva de dichos equipos, sino también de los sistemas que almacenan la información de manera intermedia para hacérsela llevar al grupo de SOC. El propio sistema empleado por el grupo de SOC debe seguir obligatoriamente dicho requerimiento.
- La implantación de nuevas dependencias en la arquitectura ofrecida debe realizarse primeramente desde un entorno de preproducción; y seguir firmemente esta misma guía. Adicionalmente, una vez incorporada exitosamente al grueso de la arquitectura, debe ser analizada detalladamente desde un punto de vista funcional y de ciberseguridad, antes de establecerla en un entorno productivo. Ignorar este requerimiento puede llevar a errores graves de ejecución, incompatibilidades, fallos de seguridad y del propio sistema.
- Se recomienda establecer cursos de aprendizaje en torno al funcionamiento de la herramienta, así como de la estructura que maneja, información tratada y utilización de datos de autenticación en torno a ésta; a los equipos especializados en ciberseguridad, particularmente al equipo forense y de SOC.
- Es obligatorio establecer un procedimiento de respuesta a amenazas que debe ser leído y consentido por parte de los empleados donde se establezca la solución EDR en sus equipos *endpoint*. De esta manera, tendrán pautas

de comportamiento y códigos de conducta a seguir en caso de reporte de amenaza por parte de la arquitectura.

Tras todo este proceso, restaría realizar una comparativa de los requerimientos implementados en base a los riesgos producidos, y la biblioteca de controles de seguridad establecida por la propia organización; con el objetivo de precisar un nivel de confianza objetivo y actual por parte de la aplicación específica. Este paso no puede detallarse en la propia guía, ya que dicha biblioteca de controles organizacionales es subjetiva al ámbito de la empresa, y no puede ser generalizada.

Aun así, el procedimiento consistiría en realizar una traza de aquellos requerimientos que se correlacionen con alguno de los controles de seguridad de la biblioteca de la corporación. Cuanto más controles se satisfagan, mayor es el nivel de confianza actual de la aplicación, y por lo tanto, más se acerca al nivel de confianza objetivo esperado por ésta. Hay que tener en cuenta que, este proceso no debería hacerse siguiente un modelo 1-1, si no que, en función de la importancia dada para ese control de seguridad por parte de la organización, el nivel de confianza aportado será mayor o menor si se cumple el requisito aplicativo que lo satisface.

Por último, una vez establecido este nivel de confianza actual; es labor del responsable asignado por el comité ONF de aceptar el nivel de riesgo residual que resta de no cumplir la totalidad de los controles de seguridad de la organización (si fuera el caso), y decidir en ese caso, implementar la arquitectura EDR propuesta en el documento.

En la tercera fase del proceso de gestión de seguridad de la aplicación, se deben definir medidas para crear y mantener el marco normativo de la aplicación. Este paso es crucial, ya que dicho marco normativo será incluido a la totalidad del ONF de la empresa en cuestión. De esta fase deben obtenerse:

- Un ANF actualizado y disponible para su uso, el cual contiene todos los elementos necesarios para la seguridad de la aplicación.
- Un ciclo de vida de la aplicación para el proyecto en el que es empleada.
- Controles de seguridad de la organización aplicables a este ANF.

Primeramente, se asignan los roles y responsabilidades en función de la métrica RACI:

Realización de actividad	Comité ONF	Gerente de proyectos	Equipo de SOC	Equipo de forense	Responsable principal SOC	Auditores	Usuario final
1) Identificar y seleccionar procesos y actividades clave del ONF para establecer en el ANF	C	A	R	I	R		
2) Comprobar el alineamiento interno del modelo de ciclo de vida de la seguridad de las aplicaciones, correlacionando con las fases y actividades de este modelo	C	A	I	I	C		
3) Importar en el ANF los procesos requeridos y los controles de seguridad de las aplicaciones identificados para el nivel de confianza actual asignado a la aplicación	I	C	R	C	A		
4) Mantener y comunicar el contenido del ANF a las personas involucradas	I	A	I	I	A		I

Tabla 9. Realización de actividades para el ANF de la aplicación

Verificación de actividad	Comité ONF	Gerente de proyectos	Equipo de SOC	Equipo de forense	Responsable principal SOC	Auditores	Usuario final
1) Asegurar que el marco normativo de la aplicación fue definido	I	R	C	C	R	A/R	
2) Asegurar que el ciclo de vida de la aplicación para el proyecto en el que se aplica fue derivado	I	R	I	I	C	A/R	
3) Asegurar que los controles de seguridad	I	R	C	C	R	A/R	

para la aplicación fueron seleccionados							
4) Asegurar que el contenido del ANF fue validado y firmado por el responsable elegido	R	C	I	I	C	A/R	

Tabla 10. Verificación de actividades para el ANF de la aplicación

Una vez definidos estos roles y responsabilidades, es necesario comprobar que los requisitos y ámbito de la aplicación fueron correctamente definidos. Para ello y en base a todo lo escrito, se debe:

- Desarrollar el ANF, de manera que cualquier información que aparezca en el ONF y le afecte, debe considerarse. En este ANF debe incluirse: nivel de confianza objetivo para la aplicación, contextos (tecnológico, de negocio y regulatorio) para la aplicación, responsabilidades de los actores definidos y cualificación profesional de cada uno, especificaciones de la aplicación, diseño de requerimientos, y procesos o elementos que intervienen en la definición, gestión y verificación de la seguridad de la aplicación. Toda esta información, como ya se ha citado, se ha ido recogiendo a lo largo del escrito. Además, en el caso de que el proyecto donde participa la aplicación fuera actualizado, el ANF debe ser renovado del mismo modo, adaptándose y recogiendo la nueva información referente a éste.
- Derivar el ciclo de vida de la aplicación. Para ello, se comprueba el modelo de ciclo de vida de la seguridad de las aplicaciones establecido por la organización, y en base a éste, determinar el propio ciclo de vida de la aplicación. De cualquier manera, y como apunte genérico, debido a las dependencias y tecnologías que se han utilizado en el desarrollo de la arquitectura, se ha estimado un ciclo de vida extremadamente largo para la aplicación.
- Seleccionar controles de seguridad para el proyecto donde participa la aplicación. Estos controles de seguridad son propios de cada organización, están recogidos en un documento, y deben ser trasladados del ONF al ANF.
- Debe cerciorarse que la seguridad de la aplicación fue especificada en función de los documentos que lo apoyan, como aquellos regulatorios, de cumplimiento y las especificaciones de los requerimientos de los *software* utilizados.
- Indicar el flujo de información de la aplicación detalladamente.
- Hay que asegurar que los contextos tecnológicos, de negocio y regulatorio de la aplicación están identificados.

- Identificar que actores fueron involucrados en los procesos de realización.
- Verificar que toda la información relevante producida ha sido recopilada en la creación del ANF.
- Preservar los resultados del proceso de verificación en el propio ANF.

Del mismo modo el ANF debería ser actualizado:

- Antes del establecimiento de objetivos considerables en el proyecto donde participa la aplicación.
- Cuando el contexto tecnológico, de negocio o regulatorio cambia.
- En procesos de auditoría periódicos.

Por último, las organizaciones deberían validar el ANF, con firma del responsable elegido por el ONF para constatar que la tarea de validación ha sido completada.

En la cuarta etapa, se especifican los detalles de aprovisionamiento y manejo de la aplicación. El objetivo principal es la implementación de los controles de seguridad especificados en el paso anterior (ANF) de manera teórica y la inclusión de éstos en el ciclo de vida de la aplicación. Se recuerda que, dichos controles de seguridad fueron identificados durante la determinación del nivel de confianza objetivo de la propia aplicación.

Los roles y responsabilidades definidas para cada uno se hacen siguiendo la métrica RACI:

Realización de actividad	Comité ONF	Gerente de proyectos	Equipo de SOC	Equipo de forense	Responsable principal SOC	Auditores	Usuario final
1) Dirección detallada de los análisis de los riesgos de seguridad de la aplicación		A/R	R	C	R	I	
2) Implementación de la actividad de seguridad correspondiente a cada control de seguridad		A	R	I	R	I	
3) Implementación de las medidas de verificación de cada control de seguridad		C	C	I	C	A/R	
4) Dar retroalimentación al		A/R	I	I	I	I	

proceso de gestión del ONF sobre lo obtenido							

Tabla 11. Realización de actividad para aplicación de controles de seguridad de la aplicación

Verificación de actividad	Comité ONF	Gerente de proyectos	Equipo de SOC	Equipo de forense	Responsable principal SOC	Auditores	Usuario final
1) Asegurar que la organización cuenta con una lista de actividades de seguridad asociada con los controles de seguridad en función del nivel de confianza objetivo del proyecto donde participa la aplicación		I			C	A/R	
2) Cerciorarse que las actividades de seguridad de dicha lista han sido ejecutadas		I			C	A/R	
3) Comprobar que la organización tiene una lista de actividades de medición de verificación que forma parte de los controles de seguridad		I			C	A/R	
4) Asegurar que las actividades de medición de la lista fueron ejecutadas en el curso de un proyecto de aplicación		I			C	A/R	
5) Verificar que un prototipo de la aplicación con los correspondientes controles de seguridad de la organización ha sido implementado y constatado		I			C	A/R	

Tabla 12. Verificación de actividad para aplicación de controles de seguridad de la aplicación

La decisión de adaptar los controles de seguridad indicados viene dada por el equipo del proyecto y el responsable (en este caso, el equipo de SOC) durante el proceso de análisis de riesgo detallado, de manera que estos controles deben ser adaptados, corregidos o modificados, o que nuevos controles de seguridad deben ser realizados para cumplir correctamente con los requisitos de seguridad de la aplicación. Si sucediese este último caso, debe ser reportado directamente al comité ONF para que ponga la puesta en marcha del proceso de gestión del documento ONF con la nueva información sobre este control de seguridad incluido en el ANF referente a la aplicación.

En la última etapa, se tiene en cuenta la manera en la que dicha aplicación debe auditarse; es decir, la verificación de la seguridad de la aplicación. Esta paso del proceso de gestión de la aplicación puede ser realizado en cualquier momento del ciclo de vida de ésta. Dependiendo de nivel de confianza objetivo de la aplicación, el proceso de auditoría se realizará una vez, periódicamente o con un seguimiento continuado.

El resultado de esta fase es el nivel de confianza actual para la aplicación en un momento determinado. Este nivel de confianza actual se considera suficiente cuando equivale al nivel de confianza objetivo o lo sobrepasa. Esto es de vital importancia ya que se verifica y se mantiene una prueba que da validación de que la aplicación es segura y se ha mantenido el nivel de confianza objetivo con el tiempo.

Se incluyen las tablas genéricas de roles y responsabilidades asociados siguiendo la métrica RACI:

Realización de actividad	Comité ONF	Gerente de proyectos	Equipo de SOC	Equipo de forense	Responsable principal SOC	Auditores	Usuario final
1) Verificar que todos los controles de seguridad de la aplicación asociados con el nivel de confianza objetivo fueron importados en el ANF e implementados en la aplicación		I			C	A/R	
2) Verificar que el nivel de confianza actual de la aplicación corresponde al menos, con el nivel de seguridad objetivo		I			C	A/R	

3) Probar si con las evidencias provistas se comprueba que se asegura y se mantiene el nivel de confianza objetivo de la aplicación con el tiempo		I			C	A/R	
4) Comprobar que las actividades de verificación para los controles de seguridad de la aplicación presentes en el ANF fueron implementadas y se dieron los resultados esperados		I			C	A/R	
5) Medir el nivel de confianza actual de la aplicación		I			C	A/R	

Tabla 13. Realización de actividad para auditoría de la aplicación

Verificación de actividades	Comité ONF	Gerente de proyectos	Equipo de SOC	Equipo de forense	Responsable principal SOC	Auditores	Usuario final
1) Comprobar que los resultados del proceso de revisión de seguridad de la aplicación demuestran que todas las medidas de verificación provistas por los controles de seguridad de la aplicación en el ANF para la aplicación en específico han sido probadas y que los resultados fueron verificados		I			C	A/R	
2) Verificar que el nivel actual de confianza de la aplicación ha sido medido		I			C	A/R	

3) Comprobar que se informe, independientemente de que se haya obtenido el nivel de confianza actual, sobre la seguridad y mantenimiento de la aplicación en específico		I			C	A/R	
4) Constatar que los resultados de la verificación, y la evidencia sobre la seguridad y mantenimiento del nivel de confianza objetivo en un momento específico, fueron archivados		I			C	A/R	

Tabla 14. Verificación de actividad para auditoría de la aplicación

Resumidamente, para llevar a cabo todo el proceso de auditoría en relación con la aplicación, se debe:

- Identificar y validar elementos del ANF en el ONF.
- Identificar y evaluar los riesgos de seguridad presentes en la aplicación.
- Identificar y validar los requerimientos de seguridad de la aplicación, incluyendo los objetivos mínimos de seguridad requeridos por la aplicación.
- Identificar y validar el nivel de confianza objetivo para la aplicación, es decir, que cumple todos los requerimientos de seguridad detectados.
- Validar que el nivel de confianza objetivo fue aprobado por el responsable elegido.
- Realizar actividades de medidas de verificación de los controles de seguridad de la aplicación.
- Actualizar el ANF.

7) Marco normativo de la aplicación. –

Application Normative Framework (ANF) o marco normativo de la aplicación es la fuente autoritaria que proporciona la información detallada de los requerimientos que una aplicación necesita para alcanzar el nivel de confianza objetivo.

Para que se dé por correcto este marco normativo, deben definirse los siguientes conceptos de manera adecuada:

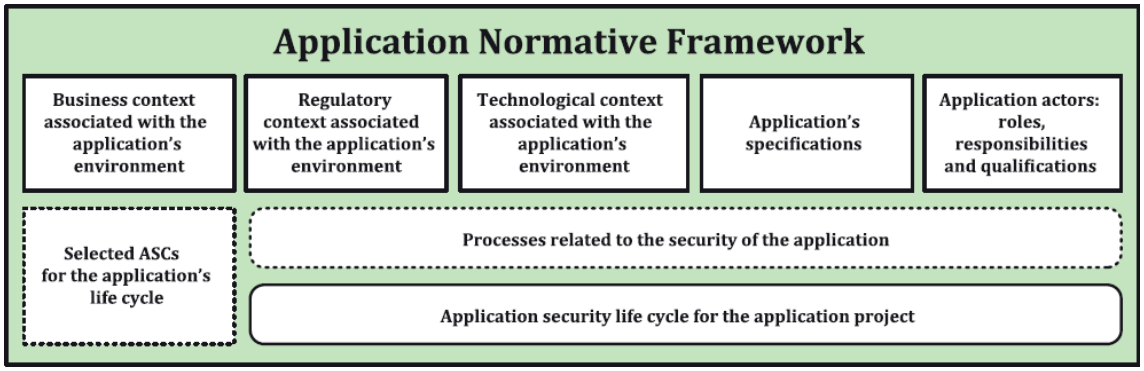

Ilustración 8. Representación de los conceptos que debe englobar el Application Normative Network (ANF)

Este ANF, una vez definido, estará sujeto a cambios, ya que si, cualquiera de los componentes que lo constituyen cambia, se produce un efecto en escalera que también afecta al ANF. Del mismo modo, cambios en este marco normativo afectarán a la seguridad de la propia aplicación. Estos cambios deberán ser aprobados por el responsable asignado, en este caso, un responsable del equipo de SOC.

De manera breve, se procede a indicar los detalles de los componentes que constituyen el marco normativo de la aplicación tratada.

- **Contexto de negocio para la aplicación:** La aplicación será utilizada únicamente en las operaciones tratadas dentro de los límites de la propia organización. Dicho de otra manera, el uso aplicativo de ésta se reduce a la protección y monitorización de los datos, sistemas y equipos que se encuentran dentro de la propia empresa; con el objetivo de proporcionar una capa de defensa adicional ante vulnerabilidades no deseadas, reduciendo el riesgo producido por éstas, y en consecuencia, el impacto reputacional y económico que pueda derivarse si se excluyese el uso de esta aplicación. En lo referente a documentos donde se involucre el funcionamiento o directivas de esta aplicación, se recogen un cumplimiento de tratamiento y uso de los datos y la información, y del mismo modo, un cumplimiento de ciberseguridad en lo referente a la protección de éstos. Del mismo modo, para la línea en la que actúa la aplicación se tienen en cuenta los siguientes documentos internos: política de manejo e implantación de contraseñas seguras actualizada, procedimiento de lista blanca para conexiones remotas con los *endpoints*, política de roles y responsabilidades,

política de auditorías internas, política de clasificación y manejo de la información y política de gestión de *logs* y registros. Entre los posibles activos de información relevantes y asociados a la organización que maneja la aplicación se pueden encontrar: información referente al dominio de la organización, datos personales de los empleados, usuario y contraseña de acceso al sistema, ficheros sensibles para la organización (como nóminas o datos de preproducción) y ficheros contractuales con terceros. El desarrollo metodológico usado en el proyecto, así como recomendaciones de buenas prácticas, lenguajes de programación, entre otros; donde participa la aplicación, es subjetivo a la propia organización. En cuanto a los marcos normativos utilizados por la propia aplicación se recogen: el estándar ISO-27034-3:2018, la guía de seguridad CCN-STIC 817, la guía de seguridad CCN-STIC 470F1 y el marco de referencia de MITRE ATT&CK. Además, deben considerarse aquellos marcos normativos del proyecto donde participa la aplicación, los cuales son inherentes a la propia definición del proyecto marcado por la organización en concreto. En cuanto a los riesgos para tener en cuenta se dividen en dos: riesgos generales y específicos, recogidos en esta misma guía, al igual que una lista de requerimientos para reducir la probabilidad de que se den dichos riesgos. La lista de controles de seguridad para la aplicación así como los niveles de confianza objetivo y actual de ésta; deben ser definidos por la propia compañía en función de la correlación de los requerimientos dados con los mismos controles que la empresa haya establecido. A pesar de esto, se señalan dos fuentes que recogen miles de ASC de uso general en muchas organizaciones y que pueden ser utilizadas para implementar los ASC en la empresa que decida usar esta guía. Dichas fuentes son la ISO 15408-3 y una publicación especial del NIST, concretamente NIST 800-53 (UNE, 2020) (NIST, 2020).

- **Contexto regulatorio para la aplicación:** La aplicación está destinada principalmente al uso de especialistas en ciberseguridad, concretamente, del campo un *blue-team* o ciberdefensa. No obstante, parcialmente los usuarios finales recibirán una notificación en el caso de que se genere una amenaza con información que debe ser apuntada para correlacionar con los resultados obtenidos por el equipo de SOC. La información gestionada por la aplicación abarca todos aquellos datos que se encuentren en el equipo del usuario final, lo que significa que pueden manejarse datos personales, financieros, de terceros

o de la propia empresa; las posibilidades son ilimitadas. Debido a que la aplicación ha sido desarrollada en territorio español, algunas de las referencias normativas empleadas deben trasladarse a aquellas correspondientes al territorio donde se utilice la aplicación, o bien, adaptarse a un contexto internacional.

- **Contexto tecnológico para la aplicación:** La aplicación debe tener plena disponibilidad de todos los eventos que ocurren en el sistema, o al menos de aquellos que aparecen en las reglas del fichero de configuración de *Sysmon*. Del mismo modo, se asegura que la integridad y confidencialidad de los archivos e información tratada se verá inalterada en todo momento; a no ser que se determine como potencialmente dañino para el sistema, y en consecuencia para la organización, en cuyo caso, serán abordados según las funcionalidades establecidas por la propia solución y reportadas al equipo de SOC correspondiente. Los detalles correspondientes a arquitectura, protocolos y lenguajes de programación utilizados pueden ser revisados en las secciones anteriores y que acompañan a esta guía. En lo determinante a la infraestructura se recomienda establecer la solución EDR *endpoint* en tantos equipos de usuario final como existan en los límites de la empresa, o al menos, en aquellos que supongan un impacto importante para la organización en caso de que se vean vulnerados. Además, se recomienda el establecimiento de dos servidores: un servidor vigilado en todo momento destinado a la instalación del ELK reducido, y otro servidor cuyo propósito será albergar las utilidades de *TheHive Project*, y que será de uso exclusivo por el equipo de SOC.
- **Roles, responsabilidades y cualificaciones para con la aplicación:** A continuación se nombran aquellos roles que se ven involucrados en el transcurso de la aplicación:
    - **Arquitecto de aplicaciones:** Su responsabilidad principal es el diseño, mantenimiento e implementación de la arquitectura. Debe tener experiencia en gestión de aplicaciones, y un conocimiento minucioso del funcionamiento de la arquitectura.
    - **Administrador de la aplicación:** Perteneciente a un integrante del equipo de SOC elegido por el comité ONF. Su labor principal recae en todas las funcionalidades de la arquitectura, así como en dictaminar tareas de gestión, modificación y decisiones finales.

- **Operador de la aplicación:** Pertenecientes al equipo de SOC, sus labores son las de monitorización constante de las amenazas que lleguen al *Security Incident Response Platform* (SIRP). Adicionalmente, deben aceptar las responsabilidades asignadas por parte del administrador de la aplicación.
- **Auditor:** Encargado de verificar que la seguridad para y con la aplicación se efectúa de manera correcta. Puede formar parte de la propia organización, o de terceros.
- ***Chief Information Security Officer* (CISO):** Responsable de definir y mantener los controles de seguridad en la organización.
- **Arquitecto de infraestructuras IT:** Encargado de seleccionar la segregación de los distintos elementos de la arquitectura, su adaptación y, en el caso de que sea necesario, la habilitación de nuevas infraestructuras para elementos recién incorporados.
- **Project Manager:** Encargado de la dirección de los proyectos estipulados en la organización. Concretamente, tendrá responsabilidades directas con el proyectos de aplicación donde la solución se encuentra incorporada; asegurándose que se alcanzan los objetivos marcados; dentro del coste, tiempo y calidad exigidos.
- **Arquitecto de seguridad:** Responsable de diseñar los controles de seguridad para mitigar los riesgos bajo unos niveles aceptables.
- **Tester:** Su función principal será la de dar testimonio de que la arquitectura funciona correctamente junto a elementos incorporados, en el entorno de preproducción.
- **User:** Empleados que se encuentran en equipos de usuario final. Su tarea principal es la de reportar al equipo de SOC cualquier incidente de seguridad que se produzca en su sistema.

- **Selección de controles de seguridad de la aplicación en las etapas de ciclo de vida de la aplicación:** Esta sección queda reservada a la documentación de controles de seguridad de las aplicaciones realizada por parte de la organización que decida utilizar esta aplicación. Sino existiese, se ofrecen como referencia la ISO 15408-3:2020 y el NIST 800-53.

- **Procesos relacionados con la seguridad de la aplicación:** Definidos anteriormente en esta misma guía. Se trata de aquellos procesos relevantes en el ONF que son importados en el ANF.
- **Ciclo de vida de la aplicación:** Esta sección queda reservada a la documentación de controles de seguridad de las aplicaciones realizada por parte de la organización que decida utilizar esta aplicación. Sino existiese, se ofrecen como referencia la ISO 15408-3:2020 y el NIST 800-53. Una vez elegidos, deben comprobarse que los controles definidos y sus medidas se aplican de manera adecuada en las distintas etapas del ciclo de vida de la aplicación.
- **Información utilizada por la aplicación:** La información manejada por la aplicación varia ampliamente, dado que es dependiente del contexto del propio equipo de usuario final donde se encuentre instalada la solución. Por ejemplo, si la solución EDR se aloja en el equipo del CISO de la organización; es más probable que la información tratada sea más sensible que aquella incluida en los sistemas de un desarrollador. No obstante, puede ser que en el equipo del desarrollador se encuentre información de carácter personal, lo que la convierte también en un ámbito sensible a ser tratado. Como resumen genérico, podría decirse que la aplicación trata con información sin impacto aparente, hasta información privilegiada.

4.3. PLANIFICACIÓN

4.3.1. PLANIFICACIÓN DEL TRABAJO REALIZADO

A continuación, se presenta una serie de puntos que, de forma muy resumida, indican las distintas etapas por las que ha ido pasando el proyecto para el desarrollo en su completitud:

- **Fase inicial:**
  - **Elección de temática y primera idea del proyecto:** En una primera instancia, el proyecto ha sido elegido entre el tutor y el autor del trabajo, y planteado más adelante por el propio director del TFM. Debido al interés personal de conocer más sobre las actividades que realizan *red team* y *blue team*, y la necesidad de aportar una solución al problema; la elección del tema fue un acierto.
  - **Investigación y viabilidad del proyecto en función de la idea inicial:** Previo a la comunicación del TFM al director del máster, se comprobó si era posible realizar

un trabajo de dichas capacidades con el tiempo disponible y las herramientas deseadas.

- **Comunicación del TFM al organismo pertinente:** Se comunica al director del máster la decisión por parte de ponente y tutor del TFM, de realizar un proyecto bajo esta temática. Esta decisión viene tomada debido a que se realizó un cambio en el primer tema que se había decidido realizar.
- **Contacto con el tutor y realización de la propuesta del TFM:** Tras la aprobación por parte del director del máster, se comunica al tutor y se comienza con la realización de la propuesta del TFM, la cual recoge diversos aspectos iniciales relacionados con éste.
- **Entrega del documento de la propuesta final:** El documento se deposita en la sección correspondiente del campus virtual del máster.

- **Fase de desarrollo:**
  - **Investigación de las tecnologías a emplear:** Para plantear una definición de la arquitectura completa, se realiza una investigación exhaustiva de las tecnologías que van a ser utilizadas en el conjunto del proyecto, y que tiene como objetivo la conformación de una arquitectura EDR compuesta por todos los elementos que definen a una solución de este tipo.
  - **Realización de la introducción, estado de la cuestión y descripción del problema:** Tal y como se indica, y en función de la información recopilada en las distintas referencias bibliográficas, se procede a la realización de la introducción, estado de la cuestión y descripción del problema. En estas tres secciones se recoge un resumen general de todo lo que se va a documentar en el escrito con el objetivo de que un lector pueda tener una primera percepción sobre el proyecto. Al mismo tiempo, se recogen antecedentes en el ámbito, motivación para la realización del proyecto y principal necesidad detectada en base a los antecedentes y el estado actual del tema.
  - **Implementación de las MV *Windows 10* y *Ubuntu* e instalación y cohesión de las dependencias en el equipo de *endpoint.* Búsqueda de nuevos artículos relacionados con el ámbito:** Antes de la integración de todos los elementos de la arquitectura, se toma la decisión de diseño de cohesionar y comprobar el correcto funcionamiento de todas las dependencias que se encuentran en el equipo del usuario final, ya que son verificables sin necesidad de recorrer la ruta completa hasta el sistema alojado con las utilidades para el equipo de SOC.

Entre estos elementos se recogen *Sysmon* y *script* en *Powershell* que proporciona funcionamiento, agente de *Elasticsearch* para la transmisión de información, una recopilación de las primeras reglas *YARA*, creación de la carpeta de cuarentena y de los ficheros de configuración genéricos para *Sysmon* y *winlogbeat*. Además de conectar todos los *software* entre sí, también se realiza una investigación sobre la siguiente tecnología que debe ser implementada en los equipos aparte, además de futuras herramientas que son necesarias para completar la arquitectura.

- **Adaptación de ficheros de configuración. Recogida de reglas *YARA*. Instalación de *ELK* reducido (sin *Elastalert)*:** Se realiza una adaptación de los ficheros de configuración del equipo *endpoint*. Esta configuración está enfocada para la demostración de que el funcionamiento del EDR es correcto, que permite flexibilidad y que es completamente de código abierto. Además, se recoge una gran cantidad de reglas *YARA* de un repositorio público y gratuito para ser usado por individuos y organizaciones, las cuales darán mayor seguridad a la hora de analizar ficheros que entran en contacto con el disco del sistema operativo. Adicionalmente, se realiza la instalación y configuración entre *Elasticsearch* y *Kibana*.
- **Configuración entre sistema *endpoint* con *winlogbeat* y equipo intermedio con *Elasticsearch*. Instalación de dependencias para SOC:** Una vez finalizada la configuración de la parte de *endpoint* se realiza la conexión entre el agente y *Elasticsearch* de manera que la información proporcionada por los eventos del sistema relacionados con *Sysmon* llegan a dicho equipo. Se realizan un par de pruebas con las *Ingest Pipeline* de *Elasticsearch* y además, se instalan en el sistema del equipo de SOC los elementos que forma el proyecto de *TheHive*.
- **Instalación y configuración de *Elastalert*. Cohesión de los componentes de *TheHive Project*:** En esta etapa, se realiza la conexión entre las dependencias del SIRP: *TheHive, Cortex* y *MISP*. Una vez instaladas todas las herramientas que se consideraron desde las primeras fases del TFM, se plantea como se pueden generar alertas sin la utilización de utilidades complicadas como *TheHive4py* o de pago, como *Splunk*. Tras la lectura sobre diversas tecnologías, se llega a *Elastalert*, una herramienta completamente gratuita y fácil de utilizar que se integra con el ELK y genera alertas a diversas plataformas, entre ellas, *TheHive.*

- **Recogida de batería de pruebas para testeo de la solución EDR:** Una vez instalada y configurada la arquitectura EDR en su completitud, se pretende la generación de una batería de pruebas que pongan en evaluación su correcto funcionamiento. Aunque como primera idea se pretende la realización o recopilación de dichas herramientas como ficheros individuales, la opción es descartada cuando se encuentran proyectos como *SysmonSimulator* o *Atomic Red Team*, las cuales están especializadas precisamente, en la prueba de reglas de *Sysmon* y de soluciones EDR, respectivamente (RootDSE, 2022) (Red Canary, 2022).
- **Desarrollo de pruebas y evidencias, resultados obtenidos, conclusiones y trabajo futuro:** Una vez realizadas las pruebas, se toman las capturas referentes a cada una de ellas y son redactadas paso a paso junto a los resultados finales recopilados tras la realización de todo el TFM. Así mismo, se incluyen las conclusiones obtenidas post concreción de éste y las líneas de trabajo futuro que podrían ser aplicadas tomando este documento como base teórico-práctica.
- **Inclusión de bibliografía y anexos:** Posteriormente, se incluyen las referencias bibliográficas empleadas como ayuda para la producción de este proyecto, así como los anexos correspondientes, entre los que se incluye un diario de investigación.

- **Fase de entrega de la memoria y defensa:**
  - **Revisión por parte del tutor del trabajo realizado por el ponente:** En esta etapa, se realiza una revisión de la memoria que resume la totalidad del TFM, y en el caso de que sea necesario, se comentan aquellas correcciones que se consideren y que debe realizar el ponente para darse por cerrado el proyecto.
  - **Entrega de la memoria del TFM:** Se deposita la memoria en el apartado correspondiente del campus virtual.
  - **Desarrollo de la presentación a utilizar en la defensa:** Se procede a la realización de una presentación en *PowerPoint* donde de una manera resumida, debe explicarse el contenido del TFM.
  - **Preparación de la defensa por parte del ponente y exposición de ésta:** Finalmente, se presenta el fruto de todo el trabajo realizado ante el tribunal mediante el *PowerPoint* anteriormente mencionado, junto a una demo técnica y las preguntas correspondientes.

Se incluye una tabla que indica el desarrollo de estas fases, junto a los días necesitados para su implementación; y el total del proyecto:

Nombre	Duración (días)
Fase inicial	**28 días**
Elección de temática y primera idea del proyecto	4 días
Investigación y viabilidad del proyecto en función de la idea inicial	4 días
Comunicación del TFM al organismo pertinente	7 días
Contacto con el autor y realización de la propuesta del TFM	12 días
Entrega del documento de la propuesta final	1 día
Fase de desarrollo	**56 días**
Investigación de las tecnologías a emplear	9 días
Realización de la introducción, estado de la cuestión y descripción del problema	7 días
Implementación de las MV *Windows 10* y *Ubuntu* e instalación y cohesión de las dependencias en el equipo de *endpoint.* Búsqueda de nuevos artículos relacionados con el ámbito. Realización de la metodología	17 días
Adaptación de ficheros de configuración. Recogida de reglas *YARA.* Instalación de *ELK* reducido (sin *Elastalert*)	5 días
Configuración entre sistema *endpoint* con *winlogbeat* y equipo intermedio con *Elasticsearch*. Instalación de dependencias para SOC	4 días
Instalación y configuración de *Elastalert*. Cohesión de los componentes de *TheHive Project*	3 días
Recogida de batería de pruebas para testeo de la solución EDR	2 días
Desarrollo de pruebas y evidencias, resultados obtenidos, conclusiones y trabajo futuro	10 días
Inclusión de bibliografía y anexos	1 día
Fase de entrega de la memoria y defensa (estimado)	**15 días (máx)**
Revisión por parte del tutor del trabajo realizado por el ponente	5-6 días
Entrega de la memoria del TFM	1 día
Desarrollo de la presentación a utilizar en la defensa	1 día
Preparación de la defensa por parte del ponente y exposición de ésta	6-7 días
Duración total	**99 días**

Tabla 15. Duración de las fases del proyecto

4.3.2. PLANIFICACIÓN DE COSTES DEL PROYECTO

En esta sección se detalla en formato tabular los costes estimados para la implementación del proyecto. Debido a que todas y cada una de las tecnologías empleadas son gratuitas, el impacto económico que suscita el proyecto que describe este TFM recae particularmente en el *hardware* usado. Este *hardware* ha sido elegido cómo recomendación para una infraestructura posible:

Tecnología	Coste	Absoluto acumulado
PC/s locales	900€ x PC	900€ (se considera 1)
Servidor *Dell PowerEdge T150 Intel Xeon E-2314*/16GB/2TB para ELK reducido	1259,01€	2159,01€
Servidor *Dell PowerEdge T40 Intel Xeon E-2224G*/8GB/1TB para *TheHive Project*	483,99€	2642,99€

Tabla 16. Planificación de costes del proyecto – hardware

Tecnología	Coste	Absoluto acumulado
Sysmon	0€	0€
Reglas YARA	0€	0€
Script en Powershell	0€	0€
Winlogbeat	0€	0€
Elasticsearch	0€	0€
Kibana	0€	0€
Elastalert	0€	0€
TheHive	0€	0€
Cortex	0€	0€
MISP	0€	0€

Tabla 17. Planificación de costes del proyecto – software

Nuevamente, todo esto a costes de implementación de la arquitectura. Los costes propios de la organización; como costes humanos, de nuevas tecnologías, entre otros; no son considerados en este cálculo.

4.4. SOLUCIÓN

Se expone la solución planteada para la resolución de los objetivos y la necesidad marcados, estableciendo la definición de la arquitectura y la implementación y configuración de sus

componentes. Para ello, la solución se ha dividido en dos fases; una primera fase de *Diseño* donde se muestra la arquitectura formalizada con diagramas incluidos y que explica de una manera resumida cómo funciona el flujo de datos desde un punto a otro de la infraestructura, y una segunda fase *Implementación y configuración de componentes* la cual explica qué aportan cada uno de los componentes y como deben configurarse para llegar a la solución.

4.4.1. DISEÑO

Para la arquitectura que representa la solución, se han definido dos diagramas. Un primer diagrama que muestra el funcionamiento únicamente desde el punto de vista del *endpoint*, y un segundo diagrama que muestra todo el conjunto de manera general.

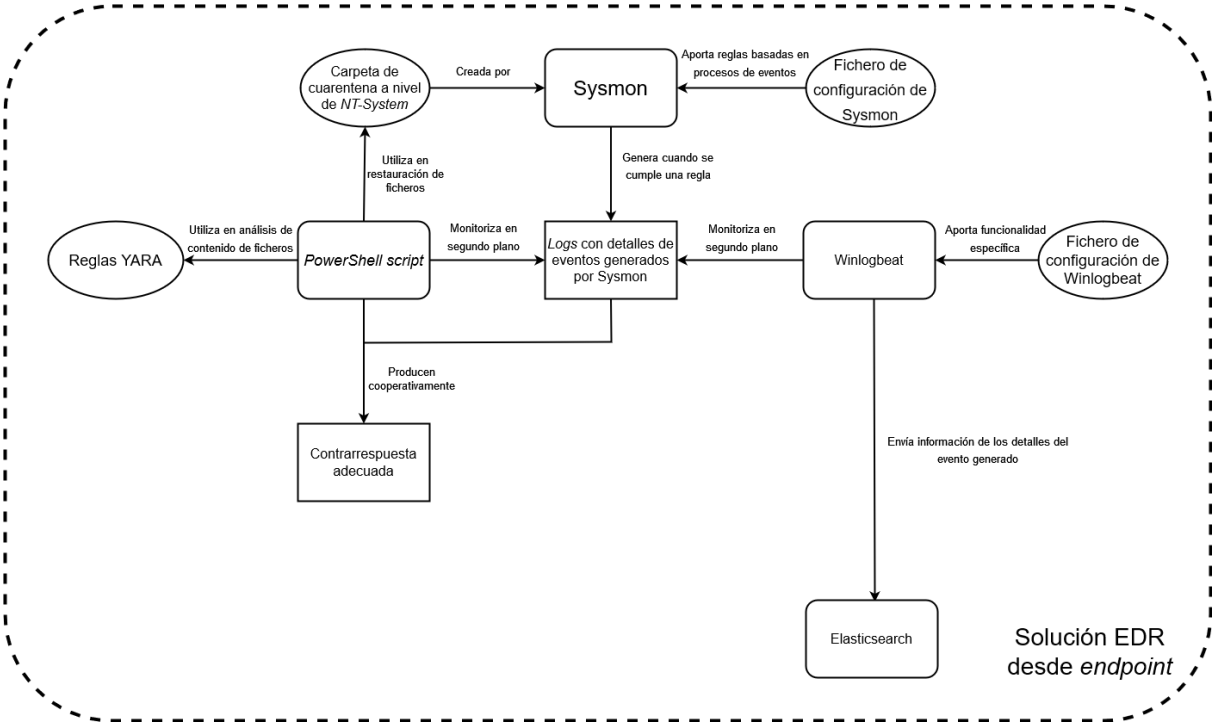

Ilustración 9. Diagrama de arquitectura - punto de vista endpoint

En este diagrama se muestra la interconexión entre los componentes de la arquitectura. Siguiendo el flujo de datos, se empieza con *Sysmon*, el cual se encarga de monitorizar lo que ocurre en el sistema en base a unas reglas definidas por el propio usuario. Si una de estas reglas se cumple, *Sysmon* generará un *log* con una serie de metadatos que varían en función del tipo de evento que será almacenado en el gestor de eventos del sistema. Cuando *Sysmon* realiza esta acción, y se almacena un nuevo *log* en el gestor de eventos se producen dos acciones principales, ambas al mismo tiempo.

- **Realización de actividades de respuesta:** Por un lado, hay un *script* desarrollado capaz de monitorizar en todo momento el gestor de eventos de la parte de *Sysmon*, función que hace gracias al uso de un registro creado previamente en el sistema y que permite establecer el control mediante WMI. Cada vez que un nuevo *log* entra, una *query* surge desde WMI y será pedida por el *script* en concreto. Con los datos recabados, se obtiene toda la información recopilada en el *log* concreto de *Sysmon*, y que será tratada en base al tipo de identificador de evento y unos parámetros adicionales que llegan a modo de *flags*, las cuales señalan que clase de respuesta debe aplicarse al evento que ha llegado. Entre estas posibles respuestas, existen el análisis de contenido de fichero y ejecutables mediante reglas *YARA*, previamente configuradas, y la posibilidad de restaurar contenido de una carpeta de cuarentena.
- **Envío de información al equipo intermedio:** En su contraparte, en el equipo se aloja una instancia de *winlogbeats*, la cual está monitorizando constantemente los *logs* que llegan al gestor de eventos del sistema por parte de *Sysmon*. Si se encuentra un nuevo registro en los eventos, *winlogbeat* envía esta información con los metadatos pertinentes al ELK reducido del equipo intermedio, más concretamente a una instancia de *Elasticsearch*.

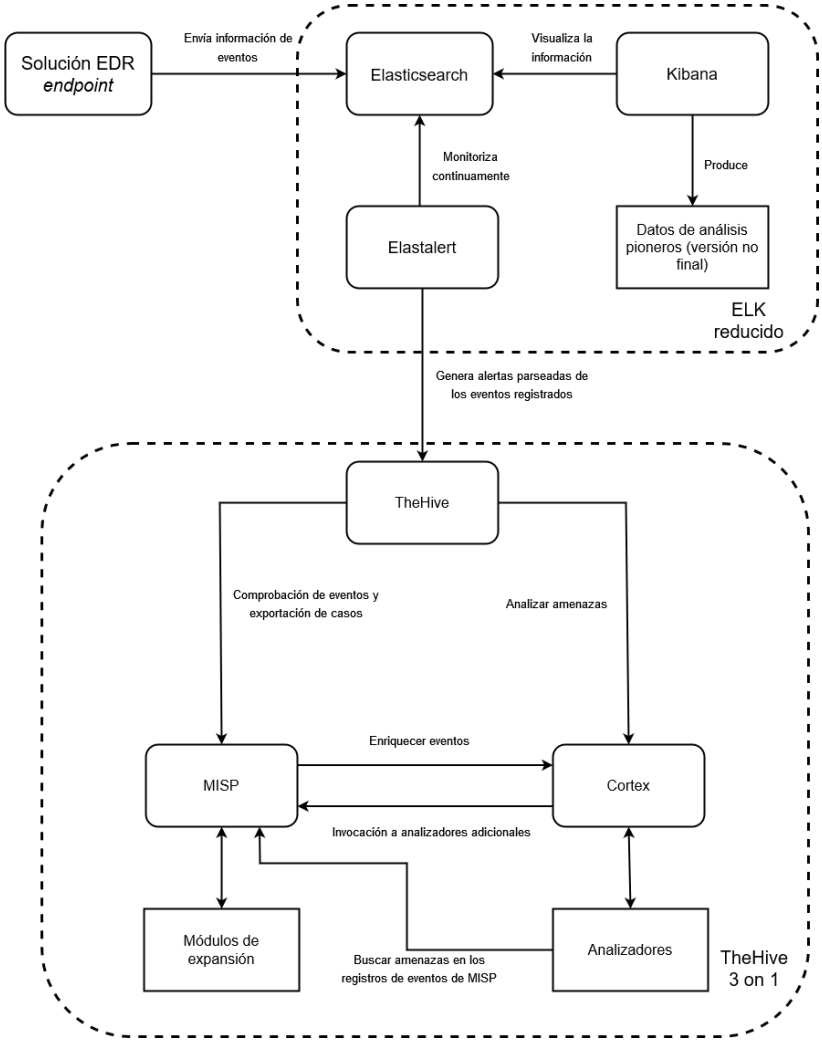

Ilustración 10. Diagrama de arquitectura - punto de vista completo

Se sigue con el segundo esquema, el cual muestra la interconexión entre los distintos equipos que componen la infraestructura. Continuando desde donde se dejó en el párrafo anterior, hay que situarse en la *Solución EDR endpoint*, la cual le envía la información a *Elasticsearch*. Debido a como se envía esta información por parte de *winlogbeat* y a la implementación básica de *Elasticsearch* mediante ficheros *JSON*, los datos pueden ser almacenados del mismo modo que salieron del equipo *endpoint*. Aun así, en este caso en concreto, una vez llegan a *Elasticsearch* un metadato bajo el nombre de *Message* es *parseado* por las *Ingest Pipelines* con el objetivo de desfragmentar los distintos campos dentro del metadato, y que se recojan de una manera óptima.

Para poder visualizar la información que se registra en *Elasticsearch* y poder tener una mejor monitorización, se hace uso de una instancia de *Kibana*, la cual permite visualizar datos de varias instancias de *Elasticsearch* y recogerlas en una sola interfaz si fuese necesario. Con el objetivo de poder enviar los datos preparados en formato *.JSON* a la plataforma de respuesta a incidentes

de seguridad; se hace uso de una herramienta conocida como *Elastalert*, la cual permite generar alertas a distintas fuentes de datos que puedan llegar a *Elasticsearch*.

Con todo esto en mente, se llega al punto en el que se genera una alerta por parte de *Elastalert* y llega a *TheHive*; y en consecuencia, al resto de módulos que acompañan a la plataforma de respuesta a incidentes de seguridad que será gestionada por el SOC.

4.4.2. IMPLEMENTACIÓN Y CONFIGURACIÓN DE COMPONENTES

El orden de desarrollo de los bloques funcionales, disponible en el apartado anterior, se ha definido así ya que unos son dependientes de otros. Por ejemplo, para poder visualizar información en *Kibana,* es necesario que *Elasticsearch* tenga datos disponibles y ordenados, y para ello es necesario que *winlogbeat* haya captado un nuevo *log* que haya sido generado por parte de *Sysmon* y registrado en el gestor de eventos del sistema.

4.4.2.1. SYSMON

Sysmon es la herramienta pilar de todo el proyecto, ya que es la base sobre la que se asientan el resto de los componentes, y es el programa utilizado para la generación de ficheros *log* con la información de los metadatos interesantes en el gestor de eventos de *Windows*.

Al ser una herramienta oficial del paquete de *Sysinternals* de *Microsoft*, únicamente es necesario descargarla e instalarla en el sistema. Además de la instalación, automáticamente elevará una tarea y un servicio que harán que *Sysmon* se mantenga funcional aunque se reinicie el sistema.

Inicialmente, *Sysmon* viene con un fichero de configuración básico, que aunque registra la información en el gestor de eventos, carece del formato necesario para que el *script* que provoca respuestas en base a los eventos pueda funcionar. Concretamente, para que una regla funcione en el fichero de configuración de *Sysmon*, debe estar constituida de la siguiente manera:

```
<Rule name="MitreRef=T1203,Technique=Exploitation for Client Execution,Tactic=Execution,Alert=Office Hacking Detected,kpp=y,kp=y" groupRelation="and">
    <ParentImage condition="contains any">winword.exe;excel.exe;powerpnt.exe;outlook.exe;msaccess.exe;mspub.exe;visio.exe;notepad.exe;wordpad.exe;eqned
    <Image condition="contains any">cscript.exe;wscript.exe;cmd.exe;powershell.exe;bash.exe;scrcons.exe;schtasks.exe;hh.exe;regsvr32.exe;regsvcs.exe;sh
    <CommandLine condition="excludes">C:\Windows\system32\spool\DRIVERS\</CommandLine>
    <CommandLine condition="excludes">\AppData\Roaming\com.ringcentral.rcoutlook</CommandLine>
    <CommandLine condition="excludes">Wget\MSRS.bat</CommandLine>
    <CommandLine condition="excludes">PhotoViewer.dll</CommandLine>
</Rule>
```

Ilustración 11. Ejemplo regla en fichero Sysmon

Es decir, cada regla que encapsule a unas condiciones en concreto debe incluir un campo *name* que funcione como estructura de una serie de campos obligatorios, entre los que se incluye:

- **TX-X-X:** ID de referencia a MITRE ATT&CK.
- **Technique:** Nombre de la técnica que recibe este identificador
- **Tactic:** Táctica asociada a la técnica
- **Alert:** Alerta elevada al usuario *endpoint* y al SOC
- **Banderas**: Indican el comportamiento que debe seguir el *script*

Esto es debido a que la manera en la que el *script* actúa respecto a la información obliga a la utilización de dichos campos. Es posible añadir campos adicionales, aunque requeriría la adaptación de las variables pertinentes en el *script*.

Una vez se haya finalizado con el fichero de configuración, es necesario guardar dicha configuración mediante el comando *'.\Sysmon64.exe -c .<path-absoluto-fichero-de-configuración>,* desde la carpeta en la que se encuentra la instancia de *Sysmon*. No existe un fichero de configuración "correcto" como tal, sino que debe ser transcrito y generado en base a las necesidades de la organización o el usuario.

Para facilidad de la generación de reglas de *Sysmon*, y consecuentemente del fichero de configuración el proyecto incluye un fichero de configuración predeterminado que aporta reglas para las pruebas aportadas por *SysmonSimulator*. Además, se incluye a continuación todos los posibles eventos generados por *Sysmon* junto a los campos asociados a cada evento, los cuales se usan en la generación de susodichas reglas:

ID	Nombre del evento	Información recogida
1	*Process Creation*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • ProcessGuid • ProcessId • Image • FileVersion • Description • Product • Company • CommandLine

		• CurrentDirectory • User • LogonGuid • LogonId • TerminalSessionId • IntegrityLevel • Hashes • ParentProcessGuid • ParentProcessId • ParentImage • ParentCommandLine
2	*A process changed a file creation time*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • ProcessGuid • ProcessId • Image • TargetFilename • CreationUtcTime • PreviousCreationUtcTime
3	*Network connection*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • ProcessGuid • ProcessId • Image • User • Protocol • Initiated • SourceIsIpv6 • SourceIp • SourceHostname

		• SourcePort • SourcePortName • DestinationIsIpv6 • DestinationIp • DestinationHostname • DestinationPort • DestinationPortName
4	*Sysmon service state changed*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • State • Version • SchemaVersion
5	*Process terminated*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • ProcessGuid • ProcessId • Image
6	*Driver loaded*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • ImageLoaded

		• Hashes • Signed • Signature • SignatureStatus
7	*Image loaded*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • ProcessGuid • ProcessId • Image • ImageLoaded • FileVersion • Description • Product • Company • Hashes • Signed • Signature • SignatureStatus
8	*CreateRemoteThread*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • SourceProcessGuid • SourceProcessId • SourceImage • TargetProcessGuid • TargetProcessId • TargetImage • NewThreadId • StartAddress • StartModule

		• StartFunction
9	*RawAccessThread*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • RawAccessRead • UtcTime • ProcessGuid • ProcessID • Image • Device
10	*ProcessAccess*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • SourceProcessGUID • SourceProcessId • SourceThreadId • SourceImage • TargetProcessGUID • TargerProcessId • TargetImage • GrantedAccess • CallTrace
11	*FileCreate*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description

		• UtcTime • ProcessGuid • ProcessId • Image • TargetFilename • CreationUtcTime
12	*RegistryEvent (Object create and delete)*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • EventType • UtcTime • ProcessGuid • ProcessId • Image • TargetObject
13	*RegistryEvent (Value Set)*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • EventType • UtcTime • ProcessGuid • ProcessId • Image • TargetObject • Details
14	*RegistryEvent (Key and Value Rename)*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User

		• Computer • Description • EventType • UtcTime • ProcessGuid • ProcessId • Image • TargetObject • NewName
15	*FileCreateStreamHash*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • ProcessGuid • ProcessId • Image • TargetFileName • CreationUtcTime • Hash
16	*ServiceConfigurationChange*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • UtcTime • Configuration • ConfigurationFileHash
17	*PipeEvent (Pipe Created)*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User

		• Computer • Description • Pipe Created • UtcTime • ProcessGuid • ProcessId • PipeName • Image
18	*PipeEvent (Pipe Connected)*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • Pipe Connected • UtcTime • ProcessGuid • ProcessId • PipeName • Image
19	*WmiEvent (WmiEventFilter activity detected)*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • EventType • UtcTime • Operation • User • EventNamespace • Name • Query
20	*WmiEvent (WmiEventConsumer activity detected)*	• Log Name • Source • Date • Event ID • Task Category • Level

		• Keywords • User • Computer • Description • EventType • UtcTime • Operation • User • Name • Type • Destination
21	*WmiEvent (WmiEventConsumerToFilter activity detected)*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • EventType • UtcTime • Operation • User • Consumer • Filter
22	*DNSEvent (DNSQuery)*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • RuleName • UtcTime • ProcessGuid • ProcessId • QueryName • QueryStatus • QueryResults • Image
23	*FileDelete (File Delete archived)*	• Log Name • Source

		• Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • RuleName • UtcTime • ProcessGuid • ProcessId • User • Image • TargetFilename • Hashes • IsExecutable • Archived
24	*ClipboardChange (New content in the clipboard)*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • RuleName • UtcTime • ProcessGuid • ProcessId • Image • Session • ClientInfo • Hashes • Archived
25	*ProcessTampering (Process image change)*	• Log Name • Source • Date • Event ID • Task Category • Level • Keywords • User • Computer • Description • RuleName

		• UtcTime • ProcessGuid • ProcessId • Image • Type
26	*FileDeleteDetected (File Delete logged)*	• Log Name • RuleName • UtcTime • ProcessGuid • ProcessId • User • Image • TargetFileName • Hashes • IsExecutable
255	*Error*	• Log Name • UtcTime • Description

Tabla 18. Generación de reglas Sysmon - Correlación ID-campos

```
<!--                                                              By: Fco Javier Perez        -->
<!--                                                                                          -->
<!--   NOTICE: This is a sysmon configuration file manually created to cover a big load of existing malware behaviour   -->
<!--              generated since begginings to 08/24/2022 (the last day this file was modified).                    -->
<!--          This doesn't mean that P-EDR is strictly thought to be used only with this sysmon config file.          -->
<!--                                                                                          -->
<!--   Github:                                                                                -->
<!--                                                                                          -->
<!--  In order to add new Rules which the EDR will act to,                                    -->
<!--  a series of tags must be added inside the 'name' parameter, all separated by comas:     -->
<!--       -MitreRef:  Reference to the Mitre ATT&CK tecnique (ex: T1127)                     -->
<!--       -Technique: Name of the said technique referenced by 'MitreRef' (ex: Trusted Developer Utilies Proxy Execution) -->
<!--       -Tactic:    Type of tacting referenced to this technique in Mitre ATT&CK (ex: Defense Evasion)  -->
<!--       -Alert:     Alert generated which will be visible for the user and the SOC team (ex: Office Hacking Detected) -->
<!--       -Flags:     Flags that will tell the EDR what actions to take. Existing flags are:  -->
<!--                   -kp=y   Kill process with child processes                               -->
<!--                   -kpp=y  Kill Parent Processes & all Child Processes                     -->
<!--                   -kc=y   Kill network connections                                        -->
<!--                   -ki=y   Kill Injected Thread                                            -->
<!--                   -sd=y   Shutdown System                                                 -->
<!--                   -fw=y   Add Windows Firewall Rule to block inbound/outbound network connectivity from process -->
<!--                   -yara=y Yara Scan file                                                  -->
<!--                   -ydel=y Delete on Yara Detection                                        -->
<!--                   -rf=y   Restore Deleted File                                            -->
<!--                   -dp=y   RAM Dumping from a Process                                      -->
<!--                   -si=y   System isolation                                                -->
<!--                                                                                          -->
```

Tabla 19. Fichero de configuración proporcionado para Sysmon por el proyecto

Finalmente, es importante señalar que en el fichero de configuración de *Sysmon*, es importante establecer una carpeta de cuarentena y recuperación de ficheros que sirva para, o bien, recuperar información eliminada por un actor malicioso, o en su contraparte, extraer ficheros borrados por el propio sistema por ser maliciosos para su análisis.

Sysmon, permite realizar esta capacidad, creando una carpeta oculta y protegida por *System ACL*, y solamente accesible mediante el usuario *NT-System*, lo que hace que sea muy complicada de localizar para un usuario que tenga actitudes dañinas para el sistema.

4.4.2.2. SCRIPT DE RESPUESTA A EVENTOS

Para la respuesta a los eventos del sistema generados por parte de las reglas implementadas en *Sysmon*, se ha diseñado un *script* en *Powershell* que hace uso de los propios comandos del sistema junto a otras herramientas adicionales para tratar de cubrir un gran abanico de funcionalidades importantes para el tratamiento de procesos, ficheros y registros del sistema. Dicho *script* es descargable desde el enlace a *Github* proporcionado y presenta las siguientes funcionalidades:

- ***Kill process*:** Termina un proceso según un PID proporcionado por el propio *Sysmon* y tratado adecuadamente por el *script*. Su bandera es 'kp=y'.
- ***Kill parent process*:** Termina un proceso padre y sus procesos hijos en función de un PID proporcionado por el propio *Sysmon* y tratado adecuadamente por el *script*. Su bandera es 'kpp=y'.
- ***YARA rule analysis for file content:*** Realiza un escaneo de un fichero o ejecutable concreto mediante la utilización de los patrones presentes en las reglas *YARA*. Su bandera es 'yara=y'.
- ***Delete on YARA detection*:** Realiza una eliminación de un fichero o ejecutable si algún contenido de este coincide con los patrones de una de las reglas *YARA* proporcionadas. Su bandera es 'ydel=y'.
- ***Restore deleted file:*** Recupera un fichero eliminado por el sistema, bien por un actor malicioso, o en su defecto, por el propio EDR para su posterior análisis. Su bandera es 'rf=y'.
- ***Shutdown system:*** Realiza un apagado del sistema si se encuentra alguna actividad inusual que debe detenerse. Su bandera es 'sd=y'.
- ***Establish firewall rule:*** Establece una regla de *firewall* especifica contra el fichero, IP o dato posible, para evitar su empleo o conexión con el equipo. Su bandera es 'fw=y'.

- ***Kill injected Thread:*** Termina un hilo inyectado en un proceso para evitar que continue una ejecución no autorizada en el sistema. Su bandera es 'ki=y'.
- ***Stop network connection:*** Detiene una conexión hacia o contra el equipo local. Es muy utilizado para sistemas que usan protección por lista blanca. Su bandera es 'kc=y'.
- ***System isolation***: Aisla un equipo completamente de la red y de Internet, con el objetivo de evitar que una amenaza continúe o se propague. Su bandera es 'si=y'.
- ***RAM file dumping:*** Realiza un volcado de memoria completo de un proceso a través de su PID. Este volcado servirá a equipos forenses para comprobar actividades inusuales por parte del proceso en base a lo registrado en memoria. Su bandera es 'fd=y'.

Para su instalación y configuración iniciales, se incluye un *script* bajo el nombre de *P-EDR-Arch_install.ps1*, que solo debe ser lanzado por consola de administrador.

Los ficheros referentes al funcionamiento del *script* serán almacenados en la ruta *'C:\ProgramData\edr'* una ruta referente a una carpeta oculta. Si la instalación ha sido correcta, debería presentarse la siguiente carpeta en la ruta:

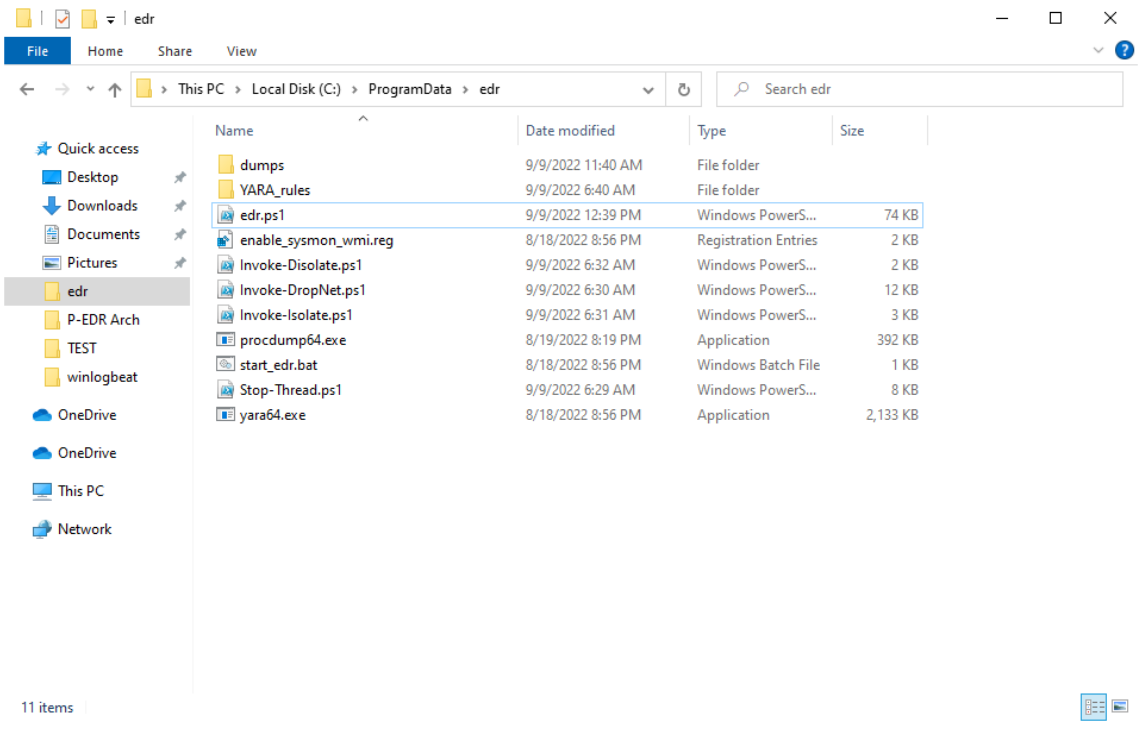

Ilustración 12. Contenido de la carpeta del script de respuesta EDR

En cuanto a las reglas YARA, la solución cuenta con un total de 592 ficheros de reglas; algunos de los cuales contienen múltiples patrones en su interior. La capacidad de análisis de contenido de ficheros y ejecutables es extraordinariamente grande. Por defecto, se ha elegido la regla *'index.yar'*, la cual utiliza todo el catálogo de reglas contra un fichero una a una. Aunque es la

opción que consume más tiempo, también es la más segura para el sistema. En cualquier caso, si quiere cambiarse, debe realizarse una modificación del *script* en sí.

```
<#
   ___        __  ___  __     _           _
  / _ \      /__\/   \/__\   /_\  _ __ ___| |__
 / /_)/____ /_\ / /\ / \//  //_\\| '__/ __| '_ \
/ ___/_____//__/ /_// _  \ /  _  \ | | (__| | | |
\/         \__/___,'\/ \_/ \_/ \_/_|  \___|_| |_|
                                    ooo,    .____.
                                    o`  o   /    |\_________________
                                   o`    'oooo()  | ________   _   _)
                                    `oo   o` \    |/         | | | |
                                      ooo'   `____'          "-"|_|

                                          By: Fco Javier Pérez

    Available features:

    -Kill process
    -Kill parent process and processes depending of it
    -YARA rules analysis for file content
    -Stop network connections
    -Add firewall rules
    -Shutdown system
    -Desktop notification for endpoint user
    -Terminate injected threads
    -Remove file after YARA detection
    -System dis/isolation
    -RAM dumping from process
    -Based on MITRE ATT&CK for alerting

#>
<# Import tools and modules #>

$yararules = "C:\ProgramData\edr\YARA_RULES\index.yar"
$procdump = "C:\ProgramData\edr\procdump64.exe"
$yaraexe = "C:\ProgramData\edr\yara64.exe"
$dumpsfolder = "C:\ProgramData\edr\dumps\"
$movedumps = "C:\ProgramData\edr\*.dmp"
Import-Module C:\ProgramData\edr\Invoke-DropNet.psl
Import-Module C:\ProgramData\edr\Stop-Thread.psl
Import-Module C:\ProgramData\edr\Invoke-Isolate.psl
Import-Module C:\ProgramData\edr\Invoke-Disolate.psl

<# Active response feature starting #>

$notification = "An anomaly was detected in the events of your system. A report was
Register-WmiEvent -Query "Select * From __InstanceCreationEvent Where TargetInstanc
```

Ilustración 13. Fichero de configuración genérico del script de respuesta

Respecto a este *script* es posible añadir funcionalidades extra mediante la modificación de éste. Esto incluye desde añadir nuevos *scripts* o ejecutables, hasta nuevas banderas que provoquen comportamientos nuevos. Para este último caso, debe seguirse la estructura definida como condicional de tipo *if*, y además otorgar el *parseo* adecuado al campo *name* que estructura técnica, táctica, alertas; entre otros. Se presenta un ejemplo de esto último:

```
#Process Terminated
if($EventID -eq 5)
{
    $msg = $Message -split "`r`n"
    $msg2 = $msg.replace(': ','=').replace('\','\\')
    $msg3 = $msg2 | Select-Object -Skip 2
    $sysmon = $msg3 | ConvertFrom-StringData
    $data = $msg[1] -split ','
    $data2 = $data |ConvertFrom-StringData
    #
    # Key/Value Tags from Sysmon RuleName
    $MitreRef = $data2."MitreRef"
    $Technique = $data2."Technique"
    $Tactic = $data2."Tactic"
    $Alert = $data2."Alert"
    #
    # Sysmon Event ID 5
    $UtcTime = ([DateTime]$sysmon."UtcTime").toLocalTime()
    $ProcessGuid = $sysmon."ProcessGuid"
    $ProcessId = $sysmon."ProcessId"
    $Image = $sysmon."Image"
    $User = $sysmon."User"
    #
    if($msg[1].ToLower().Contains("alert=")) #Mitre Attack Desktop Alerts
    {
        Write-Host "[+] Alert: $Alert $User Terminated Critical Process $Image with PID $ProcessId at $Ut
        $message = "Alert: " + $Alert + "`n" + "Technique: " + $Technique + "`n" + "Tactic: " + $Tactic +
        $message | msg *

        # Shutdown System
        if($msg[1].ToLower().Contains("sd=y")){
            Write-Host "[+] Shutting down system..."
            shutdown.exe -s -t 30 -c "This system is shutting down in 30 seconds, save your work immediat
        }
        #RAM dump process
        if($msg[1].toLower().Contains("dp=yes")){
            Write-Host "[+] Dumping full memory process to EDR folder..."
            Start-Process -FilePath $procdump -ArgumentList "-ma $ProcessId"
            Move-Item -Path $movedumps -Destination $dumpsfolder
        }
    }
    else {
    }
```

Ilustración 14. Ejemplo de ajuste de flag en script de respuesta

Por último, se indica que para que la solución funcione, el *script* debe ser lanzado en todo momento con privilegios de *NT-System*. Por defecto, se crea una tarea y servicios asociados al *script* que se inicializan junto al sistema en segundo plano, pero en el caso de que se quiera ver la información presentada por el EDR en la consola de comandos, se ha habilitado un fichero '*start_edr.bat*' que facilita lo indicado.

4.4.2.3. WINLOGBEAT

Para la recopilación de la información del gestor de eventos del sistema por parte de los *logs* generados por *Sysmon* y paso al ELK reducido del equipo intermedio, es necesaria la instalación de un agente que sea capaz de realizar esta funcionalidad. Dicho de otra manera, es el enlace principal para realizar la compartición de información de la arquitectura *EDR* con la base de datos donde se almacena toda la información de los *endpoints* de la organización.

Con objetivo de obtener dicha función, hay que descargar el agente *winlogbeat* de la página oficial, y descomprimirlo en la ruta *'C:\Program Files'* con la finalidad de evitar problemas con el *firewall* del sistema. Para su instalación es tan simple como lanzar los siguientes comandos en orden:

```
Set-ExecutionPolicy Unrestricted

.\install-service-winlogbeat.ps1
```

Seguidamente, es necesario levantar el servicio creado desde *'services.msc'* ya que por defecto no lo está:

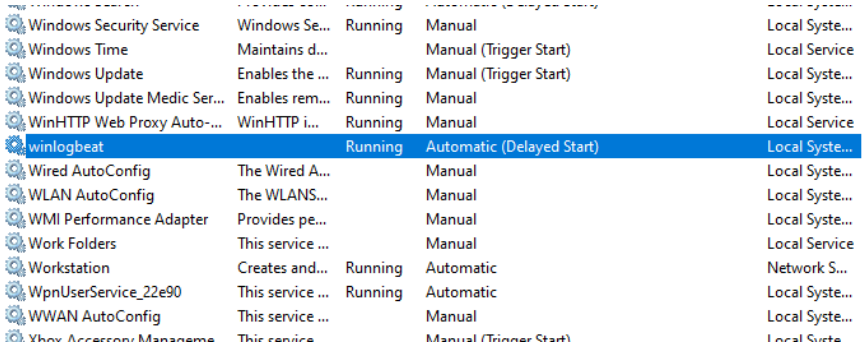

Ilustración 15. Inicialización del servicio winlogbeat

Accediendo a la ruta donde se ha instalado *winlogbeat*, se abre su fichero de configuración, y se deja escrito de la siguiente manera:

```
###################### Winlogbeat Configuration Example #########################

# This file is an example configuration file highlighting only the most common
# options. The winlogbeat.reference.yml file from the same directory contains
# all the supported options with more comments. You can use it as a reference.
#
# You can find the full configuration reference here:
# https://www.elastic.co/guide/en/beats/winlogbeat/index.html

# ======================== Winlogbeat specific options ========================

# event_logs specifies a list of event logs to monitor as well as any
# accompanying options. The YAML data type of event_logs is a list of
# dictionaries.
#
# The supported keys are name, id, xml_query, tags, fields, fields_under_root,
# forwarded, ignore_older, level, event_id, provider, and include_xml.
# The xml_query key requires an id and must not be used with the name,
# ignore_older, level, event_id, or provider keys. Please visit the
# documentation for the complete details of each option.
# https://go.es.io/WinlogbeatConfig

winlogbeat.event_logs:
  - name: Microsoft-Windows-Sysmon/Operational

# ====================== Elasticsearch template settings ======================

setup.template.settings:
  index.number_of_shards: 1
  #index.codec: best_compression
  #_source.enabled: false


# ================================== General ===================================

# The name of the shipper that publishes the network data. It can be used to group
# all the transactions sent by a single shipper in the web interface.
```

Ilustración 16. Configuración de fichero winlogbeats

De manera que solo tenga en cuenta los eventos de *Sysmon*. Adicionalmente, siguiendo la guía de instalación de *winlogbeat*, hay que añadir la dirección IP perteneciente al equipo donde se encuentra *Elasticsearch* y *Kibana.*

Se lanzan los siguientes comandos:

```
.\winlogbeat.exe test config -c .\winlogbeat.yml -e

.\winlogbeat.exe –setup dashboards
```

Dejando por finalizada la compartición de información entre el *endpoint* y el *ELK* reducido.

4.4.2.4. ELK reducido

Tanto para la instalación de *Elasticsearch* como la de *Kibana*, se siguen las guías oficiales, siguiendo los ficheros de configuración genéricos para proporcionar la conexión entre *Elasticsearch* y *Kibana* y hacerlas visibles a la red de la corporación.

La utilización de *Elasticsearch* es fundamental para la solución, ya que proporciona una base de datos *No-SQL*, la cual almacena la información que llega desde *Sysmon* como ficheros *JSON*, fáciles de *parsear*, ampliamente usados y además, permiten la introducción de nuevos campos que sean implementados desde el *endpoint*. Su funcionalidad principal por lo tanto, es el almacenamiento de los eventos del sistema referentes a los *logs* generados por *Sysmon*, con una estructura adecuada para incluir nuevos campos y un fácil tratamiento.


```
{
  "_index": ".ds-winlogbeat-8.4.1-2022.09.06-000001",
  "_id": "qowtGoMBgWuFV65u0D25",
  "_version": 1,
  "_score": null,
  "fields": {
    "event.category": [
      "process"
    ],
    "process.name.text": [
      "cmd.exe"
    ],
    "host.os.name.text": [
      "Windows 10 Enterprise"
    ],
    "winlog.provider_guid": [
      "{5770385f-c22a-43e0-bf4c-06f5698ffbd9}"
    ],
    "winlog.provider_name": [
```


Ilustración 17. Representación de datos de Elasticsearch

Kibana, por otra parte, permite la gestión de la base de datos de *Elasticsearch* para poder observar en primera instancia los eventos captados desde el *endpoint*, y además gestionar *plugins* que se integran con la herramienta para poder realizar distintas funcionalidades.

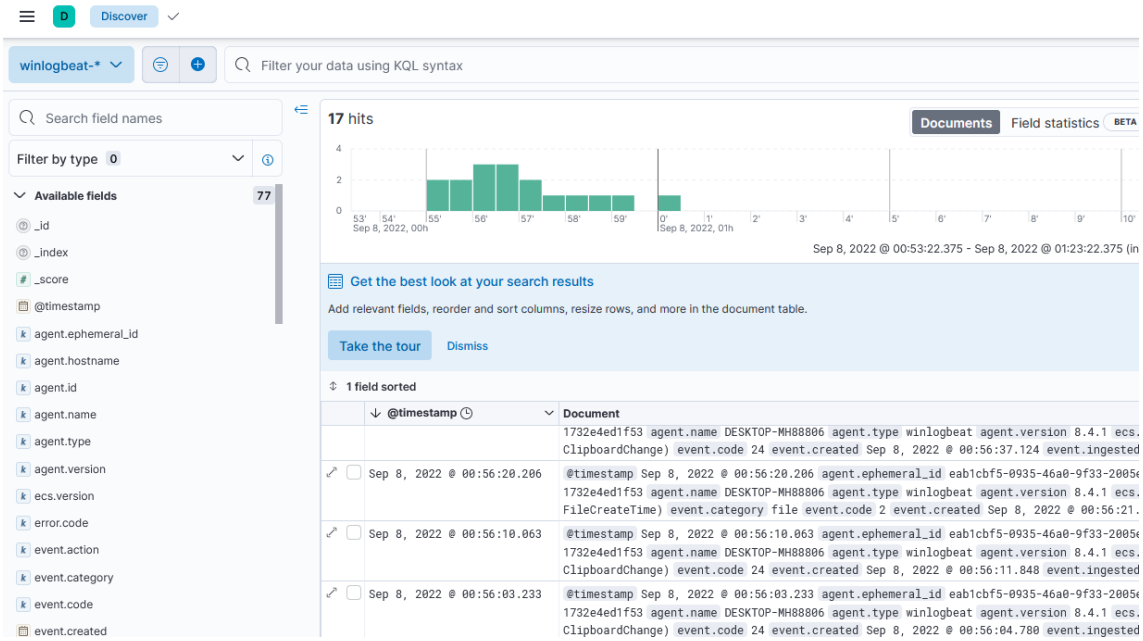

Ilustración 18. Visualización de datos de los eventos del sistema de Sysmon en Kibana

En este caso, es necesario instalar una herramienta que se integra con este ELK reducido, conocida como *Elastalert*, una pieza fundamental de toda la arquitectura ya que permite la generación de alertas al SIRP en el momento que se detectan nuevos *logs* llegando a la base de datos. Estas alertas pueden ser *parseadas* como observables, una variable especial de *TheHive* que permite realizar el tratamiento e investigación de la información de una manera más optimizada. Para proporcionar este funcionamiento, es necesario escribir y aplicar reglas especificadas en formato *.yaml*. Concretamente, para la generación de alertas a la plataforma, es necesario establecer tantas reglas como eventos de *Sysmon* se quieran pasar desde *Elasticsearch* a *TheHive*. Se presenta un ejemplo de configuración para una regla ID-1:

```
es_host: 192.168.63.3
es_port: 9200
name: TheHiveAlerts-SysmonID1
type: frequency
index: winlogbeat.-*
num_events: 1
timeframe:
    minutes:15
filter:
- term:
    winlog.event_id: "1"
realert:
    minutes: 0
alert: hivealerter
hive_connecton:
  hive_host: http://192.168.63.4
  hive_port: 9000
  hive_apikey: 5uo7NyMtalN+t/z2KyGW7kDJDz6NV42G

hive_alert_config:
  type: 'external'
  source: 'elastalert'
  description: '{rule[name]}'
  severity: 6
  tags: '{rule[name]}', '{match[host][ip]}','{match[process][hash][md5]}'
  tlp: 3
  status: new
  follow: True

hive_observable_data_mapping:
    - ip: "{match[host][ip]}"
    - hash: "{match[process][hash][md5]}"
    - file: "{match[process][name]}"
```

Ilustración 19. Ejemplo de regla para evento de Sysmon ID-1 en Elastalert

Puede concluirse por lo general que, la utilización del ELK reducido en conjunto con *Elastalert* proporcionan a la solución las medidas necesarias para gestionar, almacenar y *parsear* grandes cantidades de información antes de que sean recibidas por el equipo de respuesta a incidentes de seguridad; lo que supone un paso crucial para que la información pueda ser visualizada de manera directa y clara por los especialistas que deben establecer una contrarrespuesta de manera rápida. Además, este equipo intermedio, también sirve para generar las alertas a la plataforma adecuada, y en el caso de que se requiera en un futuro, a varias plataformas al mismo tiempo.

4.4.2.5. TheHive Project

Nuevamente, el proceso de instalación de *TheHive Project* es sencillo siguiendo la guía de instalación del propio dominio web. Ofrece dos opciones, una opción manual y una opción *dockerizada.* Ambas ofrecen la misma funcionalidad, así que la elección de cçomo instalar estas tecnologías es irrelevante.

Una vez en *TheHive,* las alertas comenzarán a llegar de manos desde *Elastalert*, con toda la información que viene recopilándose de los pasos anteriores. *TheHive Project* es la segunda

herramienta más importante integrada dentro de esta arquitectura, ya que supone la plataforma para que el equipo de SOC pueda monitorizar todas las amenazas que surgen en los equipos *endpoint*. Esto engloba amenazas ya conocidas y tratadas por el *script* de respuesta, como amenazas conocidas que necesitan estudiarse para ser incorporadas como regla al *Sysmon* bajo una funcionalidad en específico del *script* de respuesta, como amenazas nuevas que deben ser estudiadas minuciosamente y a ser posible, compartir la información obtenida.

Adicionalmente, la información que venga tipificada a modo de observables reconocidos por *TheHive* puede ser analizada más a fondo para sacar información adicional. Por ejemplo, si existe un campo *hash* de tipo *MD5*, es posible solicitarle a *Cortex* un análisis de dicho *hash* junto a un módulo que incorpore hacia *Virustotal* y del que pueda extraer antivirus que vulneran o detectan dicha amenaza, nombre y tamaño del fichero asociado al *hash*, comportamiento, criticidad; entre otros. Esta capacidad de análisis ofrecida por la solución hace que se ofrezca un motor muy potente que otras soluciones de este tipo no incorporaban en su arquitectura EDR. Concretamente una vez la información llega a *TheHive*, el conjunto de alertas u observables, entrarán en *query* para su análisis mediante *Cortex* y elevación a *MISP* en el caso de que sea necesario. Este proceso es imperceptible y queda a modo teórico en cuanto a funcionamiento; ya que es prácticamente instantáneo, siempre por supuesto, dependiendo de la cantidad de analizadores configurados a los que deban someterse los observables. La ilustración que se muestra a continuación indica el proceso seguido:

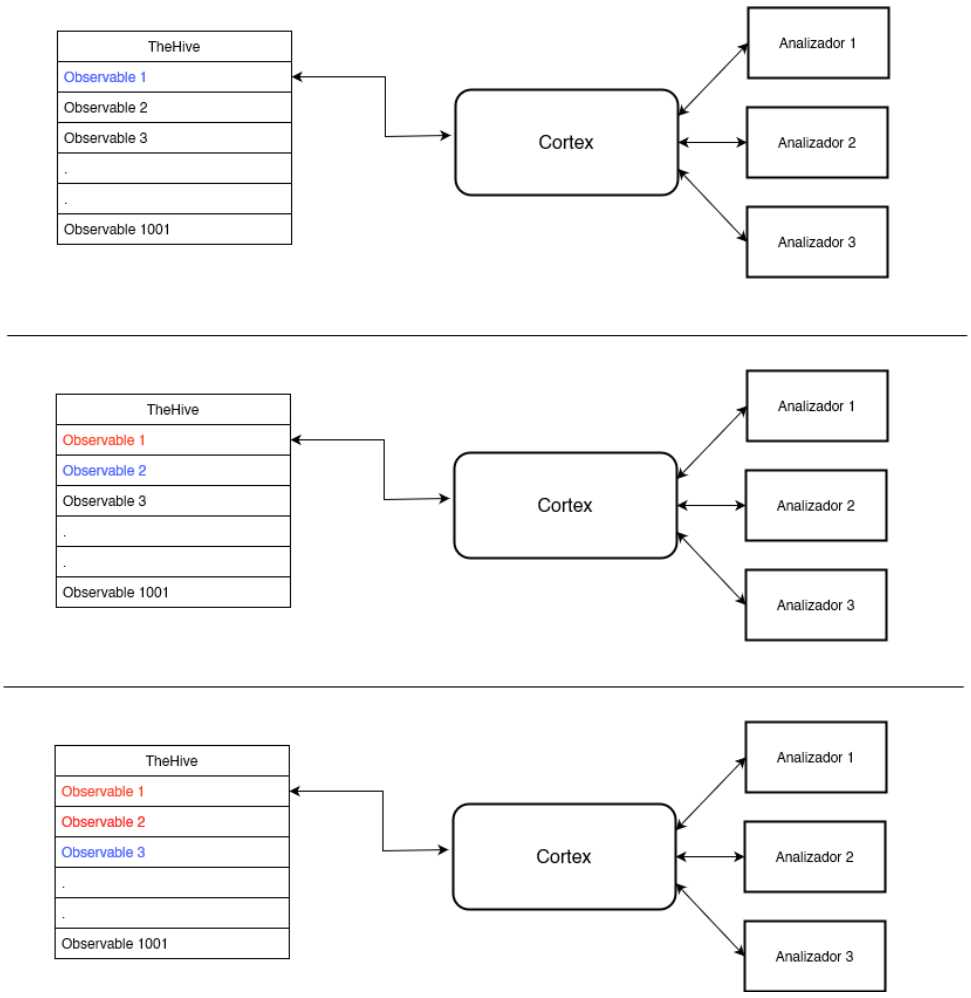

Ilustración 20. Funcionamiento de Cortex en la arquitectura

Al mismo tiempo las alertas necesarias serán elevadas a MISP; donde mediante correlación del registro de MISP se intentará proceder a la detección y puesta en contramedidas frente a la amenaza de la forma más eficaz posible. Este proceso está doblemente apoyado por Cortex; que se retroalimentará de MISP y al mismo tiempo MISP hará lo mismo mediante el uso de los analizadores de Cortex. Todo esto con el objetivo de acelerar el proceso de análisis de la manera más rápida posible.

Finalmente, las alertas en TheHive serán renovadas con todas las actualizaciones o refuerzos que los módulos incorporados hayan proporcionado frente a un observable específico.

Como puede verse, toda la solución proporcionada promete todo lo indicado en el objetivo global y específicos; ya que se ofrece una arquitectura altamente configurable en todos sus puntos, gratuita, *open-source*, flexible, con un ciclo de vida extremadamente prolongado y que destaca frente al resto de proyectos similares a éste en que ofrece todas las capacidades de un

EDR, sin falta de un solo comportamiento como ocurre con el resto de las soluciones investigadas.

5. ANÁLISIS

En esta sección, se comentará de forma breve y concisa cuales son los requisitos que la arquitectura debe cumplir para ser considerada estrictamente funcional.

5.1. REQUISITOS

- **Requisitos funcionales**
  - Capacidad de generar nuevas reglas de inclusión o exclusión para *Sysmon*.
  - Capacidad de generar nuevas funcionalidades de respuesta a eventos del sistema mediante el *script* en *Powershell* proporcionado.
  - Capacidad de *parsear* la información proveniente de *winlogbeat* como se desee.
  - Capacidad de incluir información adicional a los eventos del sistema, además de la proporcionada por el propio *Sysmon.*
  - Capacidad de generar alertas a múltiples plataformas.
  - Posibilidad de modificar los campos de los observables de la manera que el usuario desee para tener alertas completamente personalizadas con los parámetros que se interesen investigar.
  - Integración correcta y obtención de los datos originales en todas y cada una de las instancias de la arquitectura.
  - Obtención de la mayor cantidad de información posible de todas las amenazas de manera individual; para una mejor investigación.
  - Capacidad de añadir tantos analizadores como se desee.
  - Posibilidad de compartir la información de un caso en concreto en una plataforma de compartición de información de *malware*.
  - Correcto funcionamiento e implementación de una plataforma de respuesta a incidentes de ciberseguridad, junto a todos los módulos que lo rodean.
  - Monitorización, prevención, detección, respuesta, análisis y contrarrespuesta en un sistema donde la arquitectura se encuentre en funcionamiento.
- **Requisitos no funcionales:**
  - La arquitectura debe tener un **rendimiento promedio y un tiempo de respuesta aceptable**, es decir, debe ser capaz de realizar respuestas a las amenazas encontradas en tiempo real; así como elevar la alerta pertinente al equipo de SOC.

- La arquitectura debe ser **escalable**, es decir, debe soportar la posibilidad de que reciba ampliaciones o refuerzos en el futuro.
- La arquitectura debe ser **fiable**, si no lo fuese, supondría un riesgo crítico para los sistemas de la información que la empresa mantiene, y en consecuencia, a la reputación, economía y vigencia de la propia.
- La arquitectura debe ser **gratuita**, es decir, debe contar con unas dependencias que hagan que sea asequible para cualquier organización o individuo que lo desee.
- La arquitectura debe ser **fácilmente mantenible**, es decir, si en algún momento debe realizarse alguna tarea de mantenimiento frente a la arquitectura; debe realizarse de la manera más sencilla y directa posible.
- La arquitectura debe ser capaz de adecuar los datos a la base de datos correspondiente.
- Los datos almacenados deben guardarse sin perder ningún tipo de información o corromperse.

- **Requisitos de seguridad:**
  - Los ficheros de configuración en los equipos de usuario final solo deben ser modificados por la autoridad pertinente, elegida por la propia organización.
  - Deben reportarse las amenazas encontradas para una correlación efectiva entre SOC y datos originales.
  - Las cuentas de acceso a los elementos de la arquitectura deben estar almacenadas de manera segura.
  - Las credenciales para acceder a los elementos de la arquitectura deben cumplir unos requisitos mínimos: entre 8 y 24 caracteres, minúsculas, mayúsculas, números, caracteres especiales y, cambio de contraseña al menos una vez al mes.
  - El servidor que almacena la ELK reducida debe ser solo accesible por consola físicamente. Dicho servidor debe estar vigilado en todo momento.
  - El servidor que almacena *TheHive Project*, solo debe ser accesible por los integrantes del equipo de SOC.

5.2. CASOS DE USO

CASO DE USO 1.-	Configuración de *Sysmon*
Objetivos	El usuario tiene la capacidad de configurar el fichero de *Sysmon,* añadiendo las reglas que considere necesarias para aplicar funcionalidad de inclusión o exclusión a *Sysmon.*
Descripción	Se configura el fichero de *Sysmon* mediante la modificación del fichero de configuración y .xml.
Precondición	Existe una instancia de *Sysmon* correctamente instalada en el sistema.
Pasos por seguir	1.- El usuario accede a la carpeta donde se encuentra el fichero de configuración de *Sysmon.* 2.- Abre el fichero de configuración y añade las reglas que considere necesarias en la sección correspondiente, usando el lenguaje XML. 3.- Abre una terminal de *Powershell* en la ruta donde está el fichero de configuración y recarga las reglas de *Sysmon* mediante el comando: “sysmon -c <ruta-fichero-config>”.
Postcondición	Se realiza la configuración deseada en el sistema y la funcionalidad deseada se ve reflejada en la recogida de eventos del sistema.

Tabla 20. Caso de uso - Configuración de Sysmon

CASO DE USO 2.-	Añadir funcionalidad al EDR
Objetivos	El usuario tiene la capacidad de añadir nuevo comportamiento de respuesta ante las actividades de los eventos del sistema registradas por *Sysmon.*
Descripción	Se modifica el *script* en *Powershell* que proporciona funcionalidad de respuesta.
Precondición	La solución EDR propuesta está instalada correctamente en el sistema.
Pasos por seguir	1.- El usuario accede a la carpeta donde se encuentra el *script* .ps1 que referencia a la respuesta por parte de la solución EDR. 2.- Abre el *script* y añade una nueva condición referenciada por una bandera que desee. 3.- Añade la funcionalidad que desee que realice el sistema al detectarse dicha bandera en el evento generado por *Sysmon*. Para ello, utiliza *Powershell.*
Postcondición	La nueva funcionalidad se añade de manera efectiva al sistema cuando se detecta la bandera asignada.

Tabla 21. Caso de uso - Añadir funcionalidad al EDR

CASO DE USO 3.-	Login
Objetivos	El usuario tiene la capacidad de ingresar con un usuario y contraseña a cualquiera de las aplicaciones de la arquitectura (que tenga dicha función).
Descripción	Se inicia sesión con la aplicación concreta, accediendo a la interfaz de usuario.

Precondición	La tecnología se encuentra funcional y los datos introducidos son correctos.
Pasos por seguir	1.- El usuario introduce su usuario y contraseña en cualquiera de las aplicaciones. 2.- El sistema reconoce las credenciales como correctas y permite el acceso al usuario.
Postcondición	El usuario accede sin problemas.

Tabla 22. Caso de uso - Login

CASO DE USO 4.-	**Monitorización de información básica relevante desde Kibana**
Objetivos	El usuario tiene la capacidad de ver cierta información anteriormente *parseada* por las '*Ingest Pipelines*' de *Elasticsearch* en *Kibana.*
Descripción	Se pueden observar ciertas variables relevantes de las amenazas en *Kibana.*
Precondición	Hay datos disponibles en *Elasticsearch* proporcionados por *winlogbeat.*
Pasos por seguir	1.- El usuario accede al servidor donde se encuentra instalado el ELK reducido. Accede a *Kibana* mediante sus credenciales, el sistema las reconoce como correctas. 2.- Accede a la sección de '*Discover*' de *Kibana* y selecciona el *indexador* de *winlogbeat.*-.* 3.- El usuario puede realizar la comprobación de información relevante como *hash*, hora del ataque; entre otros.

Postcondición	Toda la información traída por *winlogbeat* y almacenada en *Elasticsearch* se muestra de manera correcta en *Kibana*.

Tabla 23. Caso de uso - Monitorización de información básica relevante desde Kibana

CASO DE USO 5.-	**Elevar alertas al SIRP (TheHive, Cortex y MISP)**
Objetivos	El usuario eleva alertas desde *Elasticsearch* a *TheHive* y éstas son recogidas de manera correcta.
Descripción	Cada vez que se almacena nueva información en la base de datos de *Elasticsearch*, se genera una alerta en *TheHive* con los nuevos datos, gracias al uso de *Elastalert*.
Precondición	Hay datos disponibles en *Elasticsearch* proporcionados por *winlogbeat.*
Pasos por seguir	1.- El usuario accede a la carpeta de reglas de *Elastalert*, la cual se encuentra en el servidor donde se localiza el ELK reducido. 2.- El usuario añade un nuevo fichero de regla a la carpeta de reglas, indicando la información exacta que quiere pasarse, las etiquetas a crear y los datos a *parsear* como observables. 3.- Se recarga el proceso de *Elastalert* 4.- El usuario accede a *TheHive* mediante sus credenciales, el sistema las verifica de manera correcta. Al acceder a *TheHive,* pueden verse nuevos casos en función de las alertas generadas.

Postcondición	Se muestran los casos referentes a las alertas generados en *TheHive* con los nuevos datos.

Tabla 24. Caso de uso - Elevar alertas al SIRP

CASO DE USO 6.-	Selección de analizadores en *Cortex*
Objetivos	El usuario debe tener la capacidad de elegir aquellos analizadores en *Cortex* que considere convenientes para el análisis de amenazas.
Descripción	Se seleccionan aquellos analizadores en *Cortex* que se consideren relevantes y se añaden de manera directa mediante la interfaz de *Cortex*.
Precondición	El *SIRP 3-on-1* está instalado, cohesionado y funcional.
Pasos por seguir	1.- El usuario accede a *Cortex* mediante sus credenciales. El sistema las verifica como correctas y da acceso. 2.- Utilizando la interfaz de *Cortex*, el usuario se desplaza a la lista de analizadores posibles y elige aquellos que considere. 3.- Los analizadores se añaden sin problema de manera directa.
Postcondición	Los analizadores indicados son instalados en *Cortex* de manera directa y eficaz, manteniéndose a la espera de recibir datos para su análisis.

Tabla 25. Caso de uso - Selección de analizadores en Cortex

6. PRUEBAS Y VALIDACIÓN

En esta sección se recopila de manera textual toda la información constituida por las pruebas realizadas, junto a las imágenes que validan que dichas pruebas son verídicas. Estas pruebas se resumirán de manera breve.

En líneas generales, se han llevado a cabo un total de 18 pruebas diferentes, una por cada ID posible dentro de los eventos que *Sysmon* puede originar, quitando aquellos relacionados con *WMI* y carga de *Drivers*, no porque el funcionamiento no exista; sino porque no se han encontrado buenos candidatos para someter a las pruebas. Estas pruebas se han sometido desde el punto de vista del *endpoint*, ya que, la ruta que siguen los eventos generados es siempre la misma para todos los eventos hasta llegar al SOC. A continuación, se expone un ejemplo de un evento visible desde el ELK reducido:

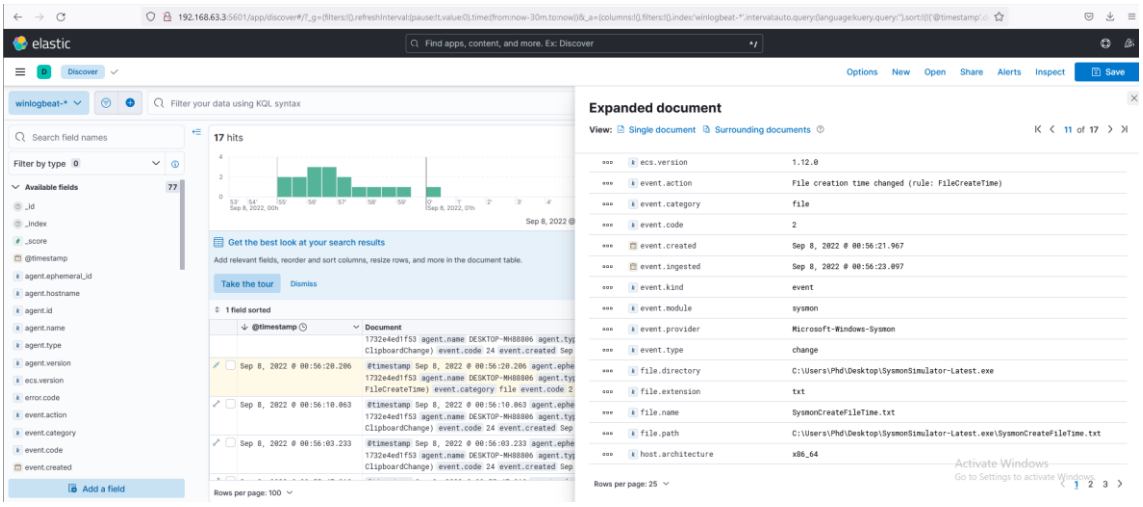

Ilustración 21. Pruebas y evidencias: Ejemplo de evento llegando al ELK reducido

Concretamente, las pruebas realizadas vienen de mano de *SysmonSimulator*, pruebas que ya vienen integradas en la propia herramienta; junto a algunas propias. Las pruebas efectuadas por *Red-Atomic-Team*, serán utilizadas y demostradas en la demo técnica pertinente.

En lo referente a reglas y posibles contrarrespuestas, todas vienen indicadas en el fichero de configuración de *Sysmon*, además de en este documento:

Ilustración 22. Fichero de configuración de Sysmon para P-EDR Arch

6.1. RESUMEN DE PRUEBAS

Como visión general de todas las pruebas realizadas, se incluye una tabla junto a un conjunto de gráficos que servirán para tener una vista rápida de la eficacia presentada por *P-EDR Arch.*

Nº prueba	ID *Sysmon*	Prueba	Resultado	Evaluación
1	1	Ejecución de comando por consola a través de macro VBA mediante *Microsoft Excel*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Se para proceso padre, hijo y análisis	✓
2	1	Ejecución de prueba *SysmonSimulator (1)* – Generación de proceso hijo WMIC	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Se para proceso padre, hijo y análisis	✓
3	2	Ejecución de prueba *SysmonSimulator (2)* – Proceso cambia fecha creación fichero	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Se para el proceso y se analiza	✓
4	3	Ejecución de prueba *SysmonSimulator (3)* – Simulación de *nmap*	Se ha detectado el proceso y generado una alerta en *host*. El mensaje no llega al equipo SOC. Se	○

			corta la conexión y se añade regla de FW	
5	3	Ejecución de *reverse-shell* a máquina Kali	Se ha detectado el proceso y generado una alerta en *host*. El mensaje no llega al equipo SOC. Se corta la conexión y se añade regla de FW	○
6	4	Detención del servicio *Sysmon*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Aislamiento del sistema e inicialización de *Sysmon*	✓
7	5	Terminación de proceso PING.EXE	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Reinicio del sistema y análisis de fichero	✓
8	5	Ejecución de prueba *SysmonSimulator (5)* – Terminación de proceso	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Reinicio del sistema y análisis de fichero	✓
9	6	Carga de driver: *natfilter.sys*	No controlado	✗
10	6	Ejecución de prueba *SysmonSimulator (6)* – Carga de *WdNisDrv.sys*	No controlado	✗
11	7	Carga de *payload* almacenada en fichero .GIF	No detectado	✗
12	7	Ejecución de prueba *SysmonSimulator (7)* – Carga de imagen *crypt32.dll*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso padre, hijo y análisis de fichero	✓
13	8	Ejecución de prueba *SysmonSimulator (8)* – Creación de hilo en memoria dinámica	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso, hilo y análisis de fichero	✓
14	9	Ejecución de prueba *SysmonSimulator (9)* – Movimiento lateral mediante '\.\'	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso y análisis de fichero	✓
15	10	Ejecución de *mimikatz* > *lsadump::sam*	Se ha detectado el proceso y generado una alerta tanto en el	✓

			equipo como en equipo SOC. Terminación proceso padre e hijo	
16	10	Ejecución de prueba *SysmonSimulator (10)* – Acceso a proceso no autorizado	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso padre e hijo	✓
17	11	Ejecución de *mimikatz* > *lsadump::sam*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso padre e hijo	✓
18	11	Ejecución de prueba *SysmonSimulator (11)* – Creación de fichero	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso y análisis de fichero creado. Aislamiento del equipo	✓
19	12	Ejecución de prueba *SysmonSimulator (12)* – Creación de clave de registro	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso y análisis de fichero creado. Aislamiento del equipo y volcado de memoria	✓
20	13	Ejecución de prueba *SysmonSimulator (13)* – Establecer valor registro	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso y análisis de fichero creado. Aislamiento del equipo y volcado de memoria	✓
21	14	Ejecución de prueba *SysmonSimulator (14)* – Renombrar clave y valor	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso y análisis de fichero creado. Aislamiento del equipo y volcado de memoria	✓
22	15	Ejecución de prueba *SysmonSimulator (15)* – Detección de fichero malicioso a través de flujo de *hash*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso y análisis de fichero creado. Terminación de proceso, análisis de fichero generado y reinicio del sistema	✓
23	16	Cambio de fichero de configuración *Sysmon*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso y análisis de	✓

			fichero creado. Aislamiento del equipo y volcado de fichero	
24	17	Ejecución de prueba *SysmonSimulator (17)* – Creación de *pipe*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso y reinicio del sistema	✓
25	18	Ejecución de prueba *SysmonSimulator (18)* – Generación de *pipe*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso, regla de FW y reinicio del sistema	✓
26	19	Ejecución de prueba *SysmonSimulator (19)* – *WmiEventFilter* actividad detectada	No controlado	✗
27	20	Ejecución de prueba *SysmonSimulator (20)* – *WmiEventConsumer* actividad detectada	No controlado	✗
28	21	Ejecución de prueba *SysmonSimulator (21)* – *WmiEventConsumerToFilter* actividad detectada	No controlado	✗
29	22	Ejecución de prueba *SysmonSimulator (22)* – DNS malicioso detectado	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso y aislamiento del sistema	✓
30	23	Creación de fichero *test* y eliminación mediante *del*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso y recuperación de fichero eliminado	✓
31	23	Ejecución de prueba *SysmonSimulator (23)* – Eliminación de fichero no autorizado (almacenado)	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. CONFUSIÓN CON ID 26 (eliminación de fichero no almacenado)	○
32	24	Copiado de fichero en *notepad++*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Aislamiento/reinicio del sistema	✓

33	24	Ejecución de prueba *SysmonSimulator (24) – Copia de portapapeles*	No detectado	✗
34	25	Ejecución de prueba *SysmonSimulator (25) - Hollowing*	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Aislamiento/reinicio del sistema	✓
35	26	Ejecución de prueba *SysmonSimulator (26) –* Eliminación de fichero no autorizado (no almacenado)	Se ha detectado el proceso y generado una alerta tanto en el equipo como en equipo SOC. Terminación proceso, análisis fichero y creación de regla FW	✓

Ilustración 23. Tabla resumen pruebas

Por lo tanto las pruebas positivas (✓), mejorables (○) y negativas (✗), pueden ser resumidas en el siguiente gráfico circular, que transcribe la efectividad de la arquitectura implementada:

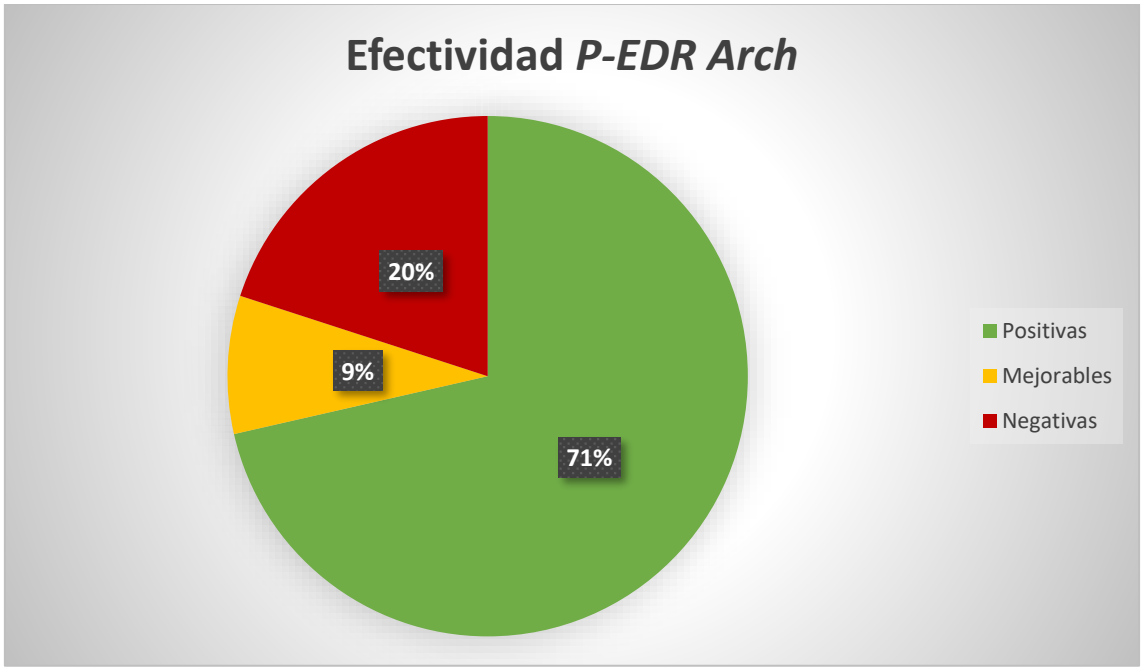

Ilustración 24. Porcentajes efectividad P-EDR Arch

Siendo la efectividad concretamente para cada prueba, según el ID de *Sysmon* bajo el que se rigen, son:

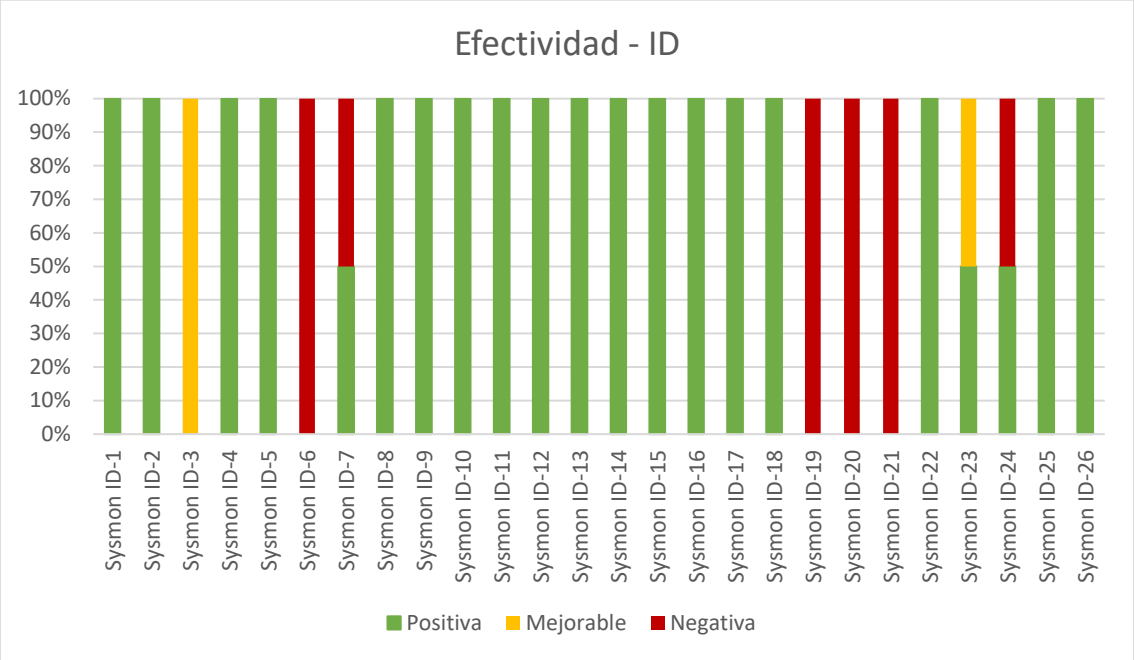

Ilustración 25. Porcentajes Efectividad - Sysmon ID

6.2. PRUEBAS ESPECÍFICAS

En este apartado se incluyen algunas de las pruebas realizadas que han tenido un funcionamiento positivo respecto a la arquitectura implementada. Dichas pruebas son presentadas a modo de demostración y funcionamiento de la solución dada para todos los ID de eventos presentes en *Sysmon*.

6.2.1. CREACIÓN DE PROCESOS NO AUTORIZADOS

Para esta prueba, se ha generado un fichero *Excel* con un macro *Visual Basic for Applications* (VBA) almacenada; la cual abre una consola de comandos con privilegios que lanza el parámetro *'/whoami'*. Cuando se abre dicho fichero, se genera un evento en el gestor de eventos del *Sysmon*, además de elevar una alerta en el *endpoint* comunicándole al usuario en específico la clase de amenaza que se está produciendo, y su obligación de informar al equipo de SOC (aunque, ya recibirán la alerta de manera automatizada). La consola de comandos no llega a ejecutarse, y el fichero *Excel* se cierra de manera inmediata.

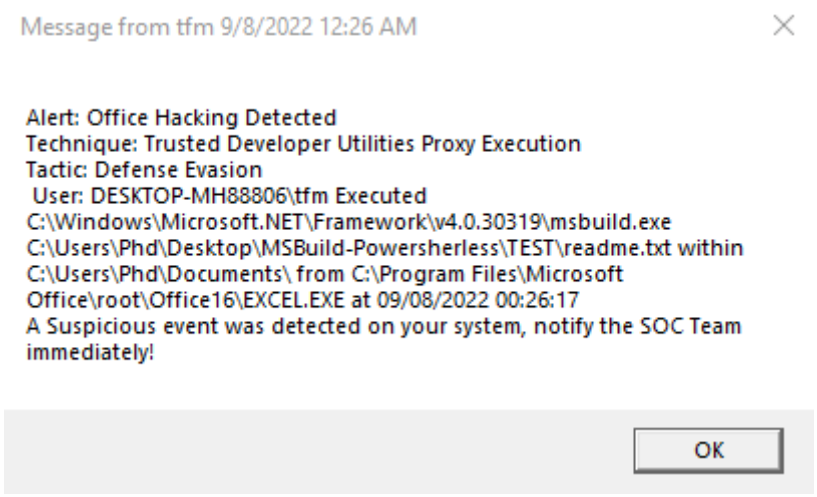

Ilustración 26. File Creation (parte 1)

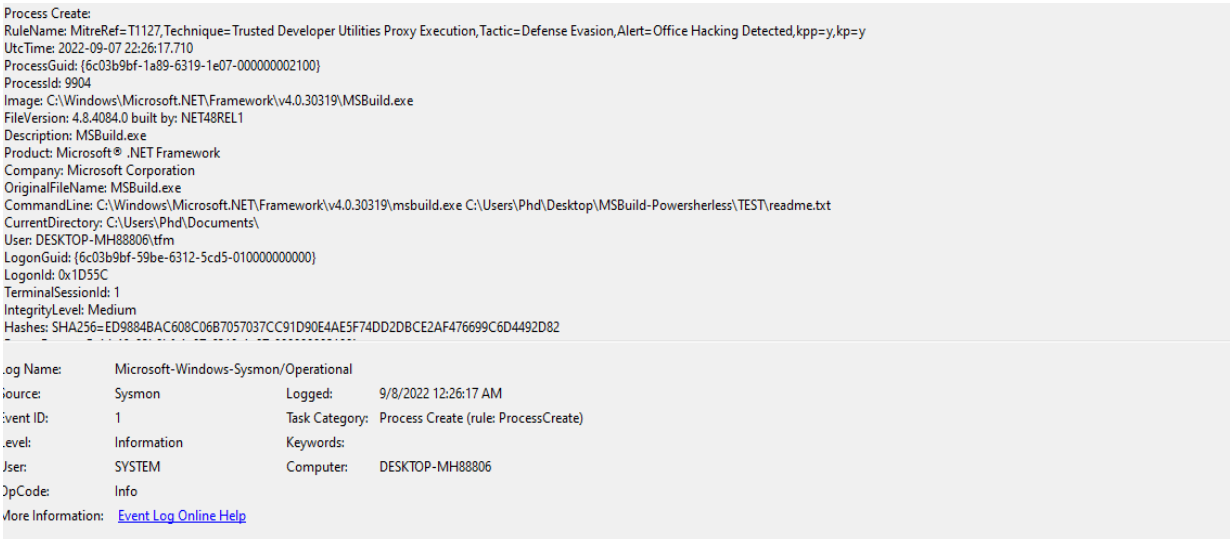

Ilustración 27. File Creation (parte 2)

6.2.2. PROCESO CAMBIANDO FECHA DE CREACIÓN DE FICHERO

Para esta prueba, se utiliza la ID 2 de *SysmonSimulator*. Se presenta el código que será ejecutado:

```
void FileCreateTime2() {

    FILETIME ft1 = { 0 };
    LPCWSTR fileName = { L"SysmonCreateFileTime.txt" };

    HANDLE hFile = CreateFileW(
        fileName,
        GENERIC_WRITE,
        FILE_SHARE_READ,
        NULL,
        CREATE_ALWAYS,
        FILE_ATTRIBUTE_NORMAL,
        NULL);

    if (hFile == INVALID_HANDLE_VALUE)
    {
        printf("[-] Could not create file SysmonCreateFileTime.txt. Error code is: %lu\n", GetLastError());
    }
    else {
        printf("[+] File Creation: %S is created in the same directory\n", fileName);
    }
    ft1.dwLowDateTime = 2421641397;
    ft1.dwHighDateTime = 30933186;

    if (!SetFileTime(hFile, &ft1, NULL, NULL)) {
        printf("[-] Error changing file creation time : %lu\n", GetLastError());
    }
    else {
        printf("[+] Time changed : Creation time of file SysmonCreateFileTime.txt is changed\n");
    }
    CloseHandle(hFile);
}
```

Ilustración 28. Cambio de fecha de creación de fichero (parte 1)

De manera resumida, se crea un fichero de nombre *SysmonCreateFileTime.txt*, del cual se modificarán dos de sus parámetros; ambos referentes a la fecha de creación del fichero. En los compiladores antiguos, no había soporte para tipos de 64 bits. La solución era dividir dicho tipo en dos valores de 32 bit, y trabajar en torno a ellos. El resultado es el cambio de la fecha de creación de un fichero; que elevará una alerta para el SOC además de una ventana en el *frontend* explicando al usuario la clase de amenaza a la que se está enfrentando. El equipo de SOC recibirá la fecha original y la modificada; lo que les permitirá aplicar una medida correctiva inmediata.

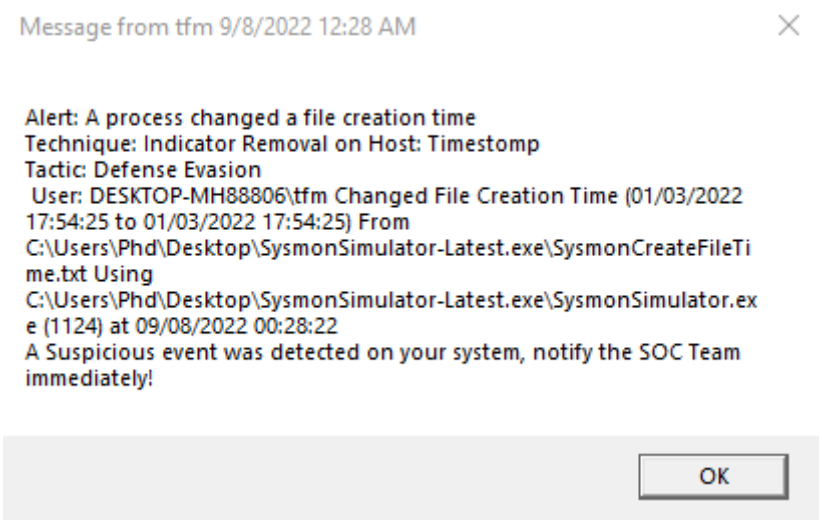

Ilustración 29. Cambio de fecha de creación de fichero (parte 2)

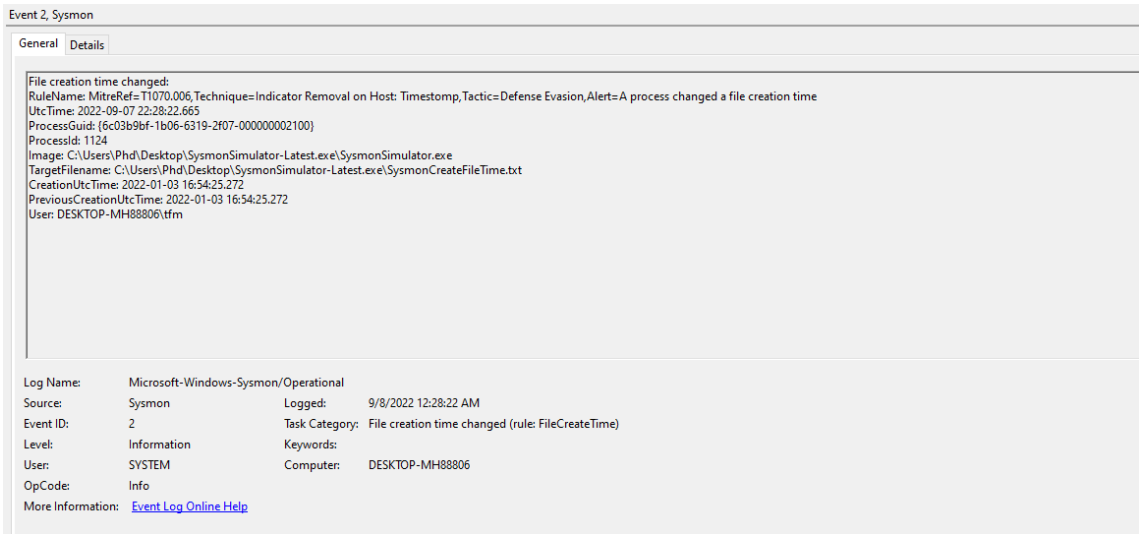

Ilustración 30. Cambio de fecha de creación de fichero (parte 3)

6.2.3. CONEXIÓN AL EQUIPO NO AUTORIZADA

Para esta prueba, se utiliza la ID 3 de *SysmonSimulator*. Se presenta el código que será ejecutado:

```c
void NetworkConnect3() {
    WSADATA version = {0};
    WSAStartup(MAKEWORD(2, 2), &version);
    u_short port = 31337;

    SOCKET newSocket = {0};
    struct sockaddr_in addr = { 0 };
    newSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    addr.sin_family = AF_INET;
    addr.sin_addr.s_addr = inet_addr("45.33.32.156");
    addr.sin_port = htons(port);

    if (connect(newSocket, (SOCKADDR*)&addr, sizeof(addr)) == SOCKET_ERROR) {
        printText("[+] Description  : Tried to initiate a network connection to port 31337 on NMAP which is closed\n", FOREGROUND_RED);
    }
    else {
        printf("[+] Description  : Tried to initiate a network connection to port 31337 on NMAP which is opened\n");
        printf("[+] Successful   : Created Network connection Event successfully\n");
    }

    closesocket(newSocket);
    WSACleanup();
}
```

Ilustración 31. Conexión al equipo no autorizada (parte 1)

De manera resumida, se generan todos los componentes necesarios (*socket, IP, puerto)* para realizar una conexión *TCP* al puerto 31337 con dirección IP 45.33.32.156; una dirección IP asociada a un dominio de *nmap*. Una vez realizada la conexión, se detiene de manera inmediata al ser una conexión no autorizada según las reglas de *Sysmon* y se crea una regla de *firewall*. Adicionalmente, se generan las alertas pertinentes:

Technical details

here you see the reverse hostname and if the given IP Address is a public or private IP Address.

IP address	**45.33.32.156**
Hostname	scanme.nmap.org
Type	Public
CIDR	45.33.32.156/24

Ilustración 32. Conexión al equipo no autorizada (parte 2)

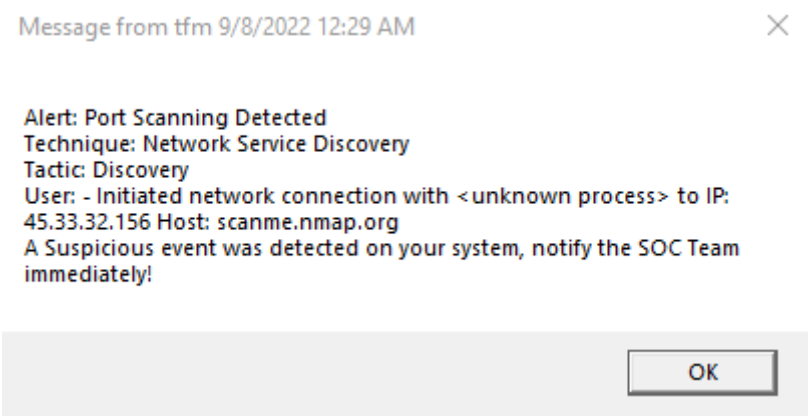

Ilustración 33. Conexión al equipo no autorizada (parte 3)

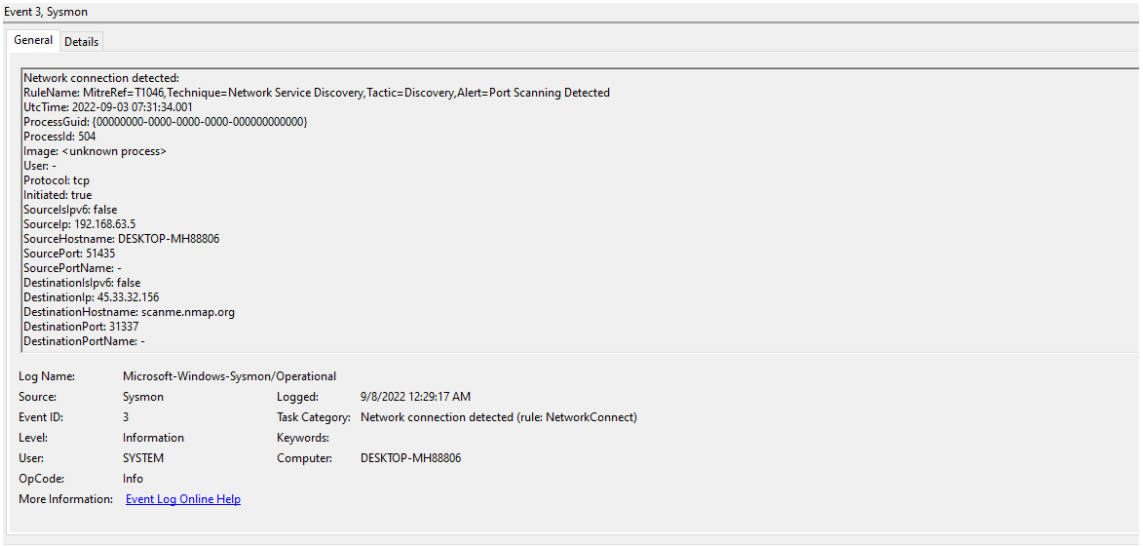

Ilustración 34. Conexión al equipo no autorizada (parte 4)

6.2.4. TERMINACIÓN NO AUTORIZADA DE PROCESO

Para esta prueba, se utiliza una regla de *Sysmon* que controla cuando se detiene el proceso PING.EXE, asociado a la comunicación entre dos equipos. Se ejecuta el siguiente comando en la *Powershell*:

```
PING.EXE -T 192.168.63.3
```

La cual crea una comunicación continua entre el equipo *endpoint* y el intermedio, referente al ELK reducido. Mediante una instancia de *Process Explorer*, del paquete de *Sysinternals*, se localiza el proceso referente a PING.EXE, y se termina. Como resultado; se generan la serie de alertas referentes al evento necesario, tanto a nivel de *endpoint* como para el equipo de SOC. Además, se avisa de que el equipo se reiniciará en breve. El objetivo principal es aplicar esta regla al propio *Sysmon* o al proceso de respuesta del EDR, ya que se inician con el sistema y en

el caso de que un atacante quisiera terminarlos; habría una respuesta para expulsar al actor malicioso del sistema y volver a levantar los procesos de protección de éste.

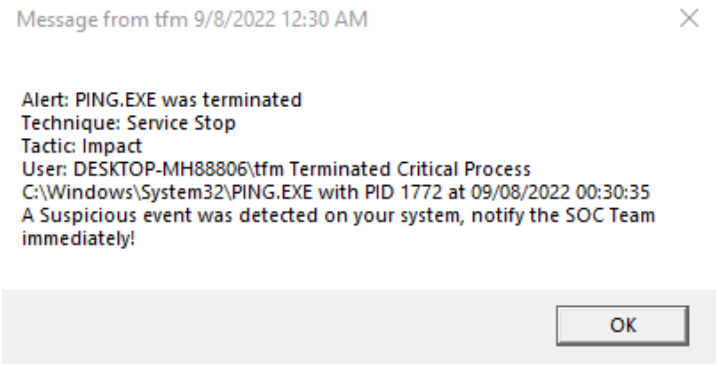

Ilustración 35. Terminación no autorizada de proceso (parte 1)

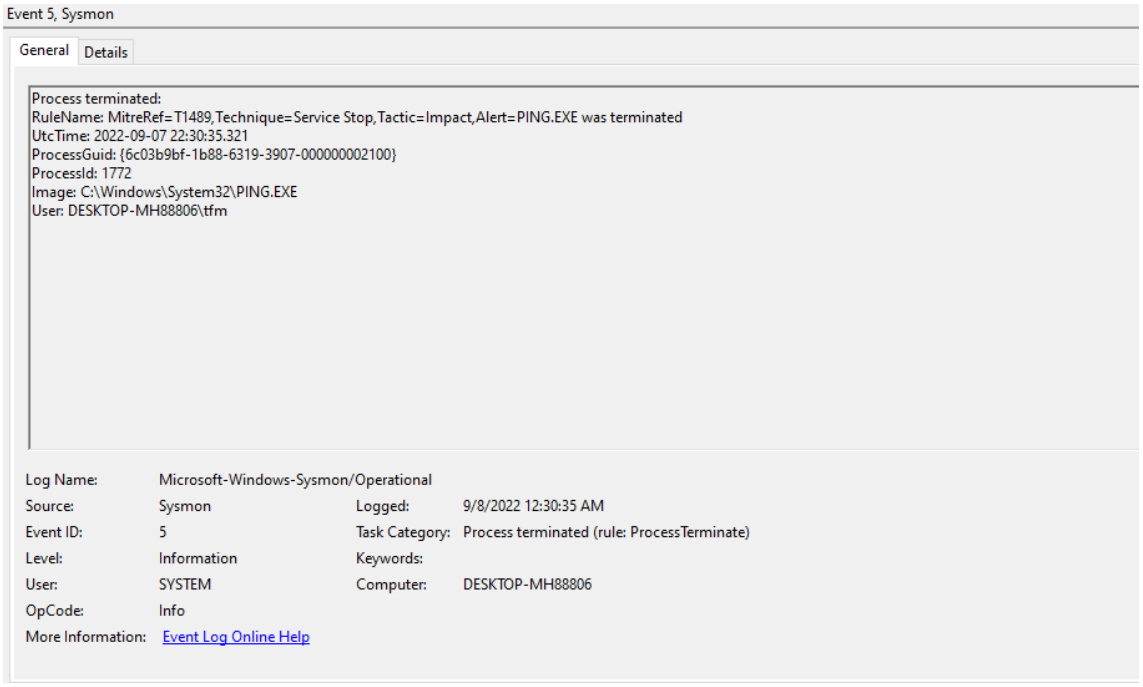

Ilustración 36. Terminación no autorizada de proceso (parte 2)

6.2.5. CARGA DE IMAGEN NO AUTORIZADA A TRAVÉS DE PROCESO

Para la realización de la prueba, se utiliza el ID 7 de *SysmonSimulator*. Se adjunta el código pertinente:

```
void ImageLoaded7() {
    HMODULE hntdll = LoadLibraryA("crypt32.dll");
    if (hntdll) {
        printf("[+] Image Loaded : Loaded crypt32.dll\n");
        FreeLibrary(hntdll);
        CloseHandle(hntdll);
    }
    else
    {
        printf("Error -: %lu\n", GetLastError());
    }
}
```

Ilustración 37. Carga de imagen no autorizada a través de proceso (parte 1)

La explicación es simple; a través del proceso de *SysmonSimulator*; se carga la librería *crypt32.dll* en el sistema (la cual, podría estar modificada con un *payload*), y posteriormente la libera y cierra el objeto para que no haya efectos indeseados en la prueba. Aun así, sin esta liberación sería posible realizarla matando el proceso padre y sus hijos.

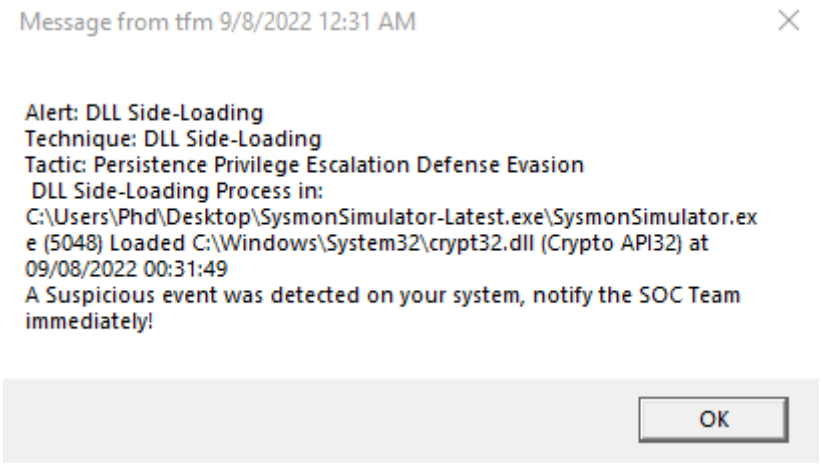

Ilustración 38. Carga de imagen no autorizada a través de proceso (parte 2)

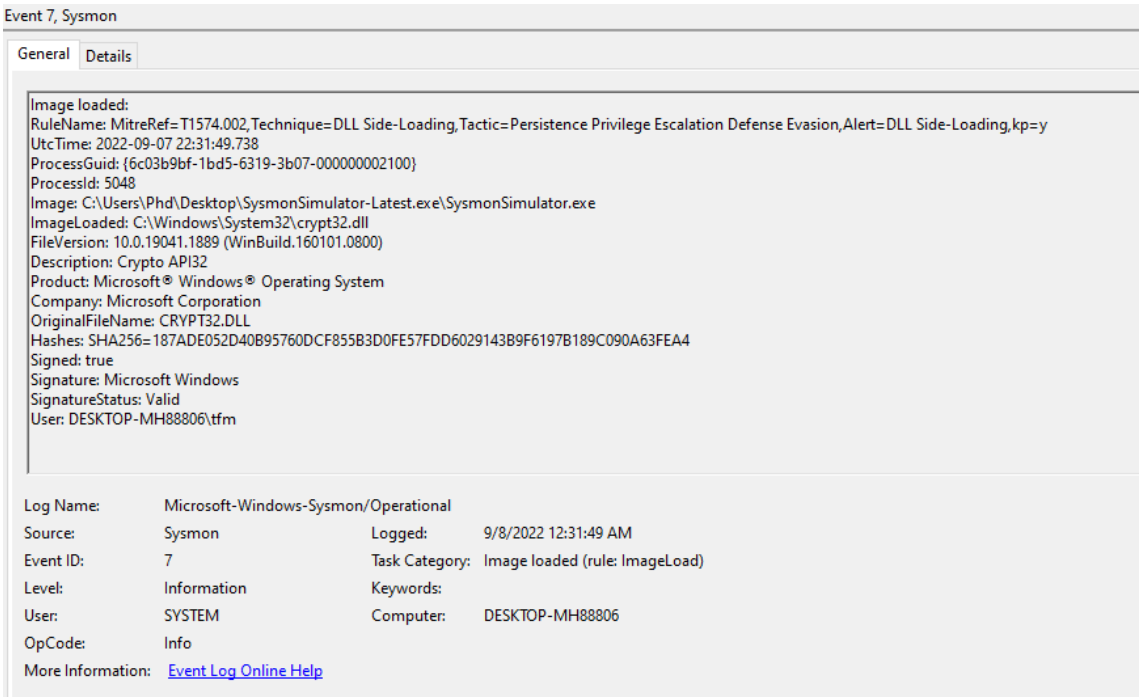

Ilustración 39. Carga de imagen no autorizada a través de proceso (parte 3)

6.2.6. CREACIÓN DE HILO REMOTO A TRAVÉS DE PROCESO

Para la realización de este proceso, se utiliza la ID 8 de *SysmonSimulator*. Para ello, se carga un hilo remoto a través del propio proceso; que ejecutará una ventana con un mensaje de saludo. Se muestra el código pertinente:

```
void createRemoteThread8() {
    HANDLE processHandle;
    PVOID buff;
    LPSTR cmdline = "C:\\Windows\\System32\\PING.exe";
    HANDLE hProcess = INVALID_HANDLE_VALUE;
    STARTUPINFOA sinfo = { 0 };
    sinfo.cb = sizeof(STARTUPINFOA);
    PROCESS_INFORMATION pinfo = { 0 };


    //Messagebox Shellcode
    unsigned char shellcode[] =
        "\x48\x83\xEC\x28\x48\x83\xE4\xF0\x48\x8D\x15\x66\x00\x00\x00"
        "\x48\x8D\x0D\x52\x00\x00\x00\xE8\x9E\x00\x00\x00\x4C\x8B\xF8"
        "\x48\x8D\x0D\x5D\x00\x00\x00\xFF\xD0\x48\x8D\x15\x5F\x00\x00"
        "\x00\x48\x8D\x0D\x4D\x00\x00\x00\xE8\x7F\x00\x00\x00\x4D\x33"
        "\xC9\x4C\x8D\x05\x61\x00\x00\x00\x48\x8D\x15\x4E\x00\x00\x00"
        "\x48\x33\xC9\xFF\xD0\x48\x8D\x15\x56\x00\x00\x00\x48\x8D\x0D"
        "\x0A\x00\x00\x00\xE8\x56\x00\x00\x00\x48\x33\xC9\xFF\xD0\x4B"
        "\x45\x52\x4E\x45\x4C\x33\x32\x2E\x44\x4C\x4C\x00\x4C\x6F\x61"
        "\x64\x4C\x69\x62\x72\x61\x72\x79\x41\x00\x55\x53\x45\x52\x33"
        "\x32\x2E\x44\x4C\x4C\x00\x4D\x65\x73\x73\x61\x67\x65\x42\x6F"
        "\x78\x41\x00\x48\x65\x6C\x6C\x6F\x20\x77\x6F\x72\x6C\x64\x00"
        "\x4D\x65\x73\x73\x61\x67\x65\x00\x45\x78\x69\x74\x50\x72\x6F"
        "\x63\x65\x73\x73\x00\x48\x83\xEC\x28\x65\x4C\x8B\x04\x25\x60"
        "\x00\x00\x00\x4D\x8B\x40\x18\x4D\x8D\x60\x10\x4D\x8B\x04\x24"
        "\xFC\x49\x8B\x78\x60\x48\x8B\xF1\xAC\x84\xC0\x74\x26\x8A\x27"
        "\x80\xFC\x61\x7C\x03\x80\xEC\x20\x3A\xE0\x75\x08\x48\xFF\xC7"
        "\x48\xFF\xC7\xEB\xE5\x4D\x8B\x00\x4D\x3B\xC4\x75\xD6\x48\x33"
        "\xC0\xE9\xA7\x00\x00\x00\x49\x8B\x58\x30\x44\x8B\x4B\x3C\x4C"
        "\x03\xCB\x49\x81\xC1\x88\x00\x00\x00\x45\x8B\x29\x4D\x85\xED"
        "\x75\x08\x48\x33\xC0\xE9\x85\x00\x00\x00\x4E\x8D\x04\x2B\x45"
        "\x8B\x71\x04\x4D\x03\xF5\x41\x8B\x48\x18\x45\x8B\x50\x20\x4C"
        "\x03\xD3\xFF\xC9\x4D\x8D\x0C\x8A\x41\x8B\x39\x48\x03\xFB\x48"
        "\x8B\xF2\xA6\x75\x08\x8A\x06\x84\xC0\x74\x09\xEB\xF5\xE2\xE6"
        "\x48\x33\xC0\xEB\x4E\x45\x8B\x48\x24\x4C\x03\xCB\x66\x41\x8B"
        "\x0C\x49\x45\x8B\x48\x1C\x4C\x03\xCB\x41\x8B\x04\x89\x49\x3B"
        "\xC5\x7C\x2F\x49\x3B\xC6\x73\x2A\x48\x8D\x34\x18\x48\x8D\x7C"
        "\x24\x30\x4C\x8B\xE7\xA4\x80\x3E\x2E\x75\xFA\xA4\xC7\x07\x44"
        "\x4C\x4C\x00\x49\x8B\xCC\x41\xFF\xD7\x49\x8B\xCC\x48\x8B\xD6"
        "\xE9\x14\xFF\xFF\xFF\x48\x03\xC3\x48\x83\xC4\x28\xC3";


        if (CreateProcessA(NULL, cmdline, NULL, NULL, FALSE, CREATE_SUSPENDED, NULL, NULL, &sinfo, &pinfo)) {
            int process_id = pinfo.dwProcessId;
            printf("[+] Inject into  : PID %lu\n", process_id);

            processHandle = OpenProcess(PROCESS_ALL_ACCESS, FALSE, process_id);
            printf("[+] Opened process's handle\n");
```

```
            buff = VirtualAllocEx(processHandle, NULL, sizeof(shellcode), (MEM_RESERVE | MEM_COMMIT), PAGE_EXECUTE_READWRITE);

            if (buff) {
                WriteProcessMemory(processHandle, buff, shellcode, sizeof(shellcode), NULL);
                CreateRemoteThread(processHandle, NULL, 0, (LPTHREAD_START_ROUTINE)buff, NULL, 0, NULL);
                printf("[+] Created Remote Thread\n");
                printf("[+] Closed Handle to the process\n");
            }
            else {
                printf("[-] Error code is : %lu\n", GetLastError());
            }

            CloseHandle(pinfo.hProcess);
            CloseHandle(pinfo.hThread);
        }
}
```

Ilustración 40. Creación de hilo remoto a través de proceso (parte 1)

De manera resumida, esta prueba crea un proceso PING.EXE que inyectará en memoria dinámica una *shellcode* que está en formato de *buffer* y que abrirá una pestaña con un mensaje de saludo al usuario. En este caso, tanto el hilo inyectado como el propio proceso PING.EXE son parados antes de que puedan ejecutare, y además, pasan por reglas *YARA* para comprobar si son objetos maliciosos. Por último, se elevan las alertas concernientes:

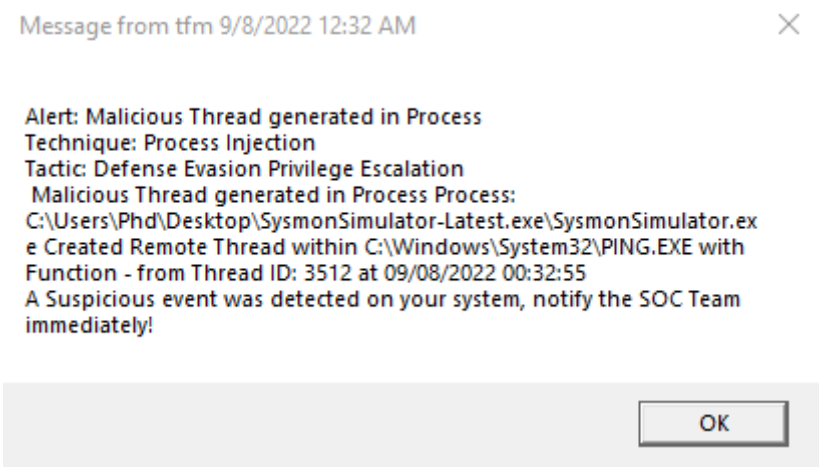

Ilustración 41. Creación de hilo remoto a través de proceso (parte 2)

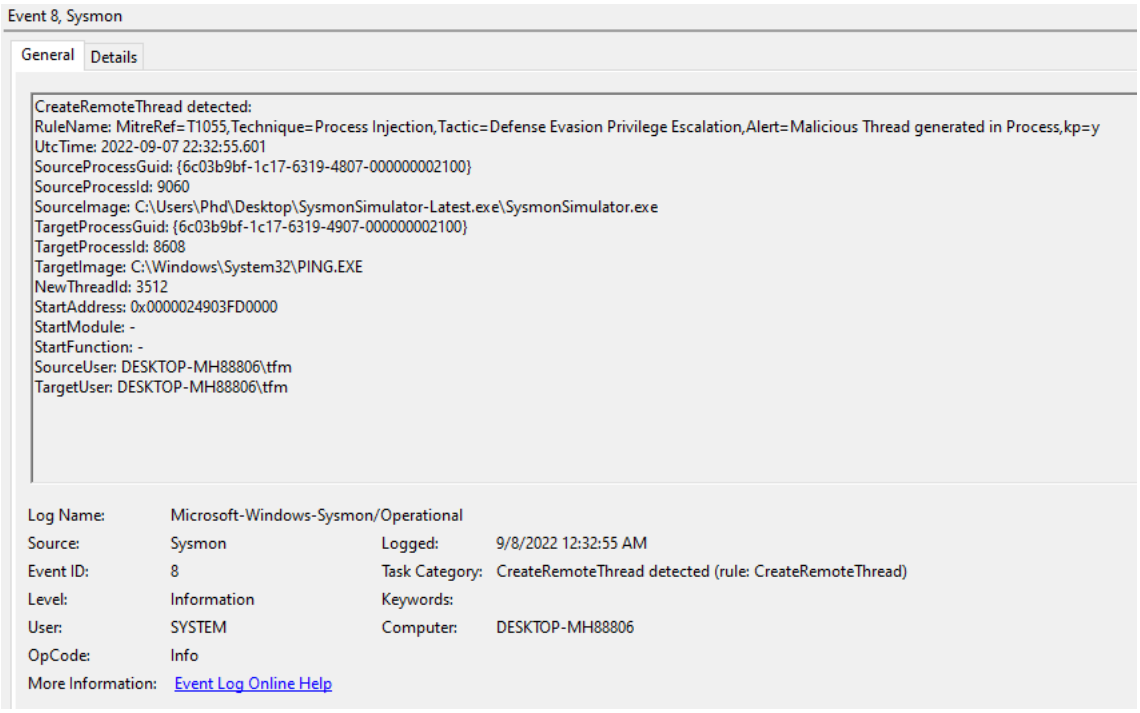

Ilustración 42. Creación de hilo remoto a través de proceso (parte 3)

6.2.7. MOVIMIENTO LATERAL MEDIANTE ACCESO DIRECTO (RAW)

Para esta prueba, se utiliza la ID 9 de *SysmonSimulator*. Se presenta el código pertinente:

```
DWORD rawaccessread9() {

    PTCHAR deviceName = _T("\\\\.\\C:");
    PWCHAR search = _T("*lsass*.dmp");
    HANDLE hfileHandle = INVALID_HANDLE_VALUE;

    hfileHandle = CreateFile(
        deviceName,
        FILE_WRITE_ATTRIBUTES,
        FILE_SHARE_READ | FILE_SHARE_WRITE,
        NULL,
        OPEN_EXISTING,
        FILE_ATTRIBUTE_NORMAL,
        NULL
    );

    if (hfileHandle == INVALID_HANDLE_VALUE)
    {
        printf("\r\nERROR code is : %lu\r\n", GetLastError());
        if (GetLastError() == 5) {
            printText("\r\n[!] This command requires administrator privileges\r\n\n", FOREGROUND_RED);
            return GetLastError();
        }
    }
    else
    {
        printf("[+] Successful   : Successfully created RawAccessRead Event\r\n");
        CloseHandle(hfileHandle);
    }
    return GetLastError();
}
```

Ilustración 43. Movimiento lateral mediante acceso directo (parte 1)

Resumidamente, se realiza acceso *raw* mediante '\\.\'. Este tipo de comando permite desplazarse por el sistema, incluso por ciertas carpetas privilegiadas, saltándose las medidas de seguridad genéricas. En cuanto se observa este comportamiento, se cierra el proceso y se analiza el fichero mediante reglas *YARA*. Además, se generan las alertas correspondientes.

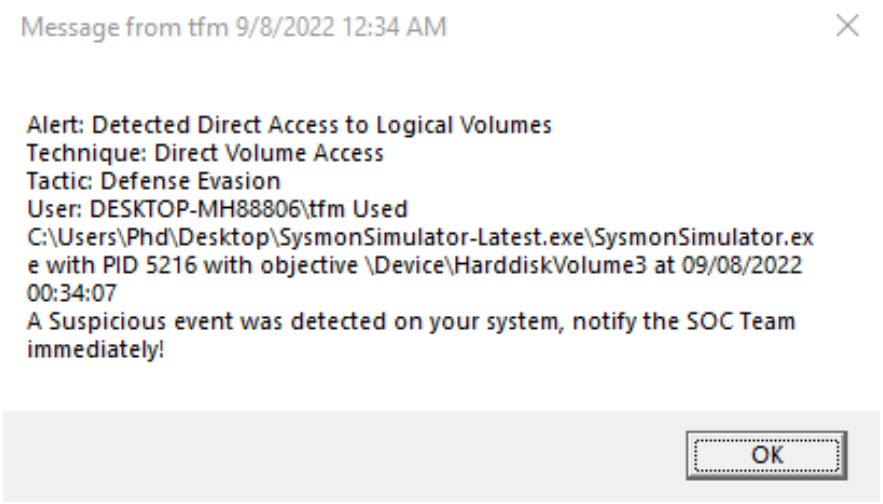

Ilustración 44. Movimiento lateral mediante acceso directo (parte 2)

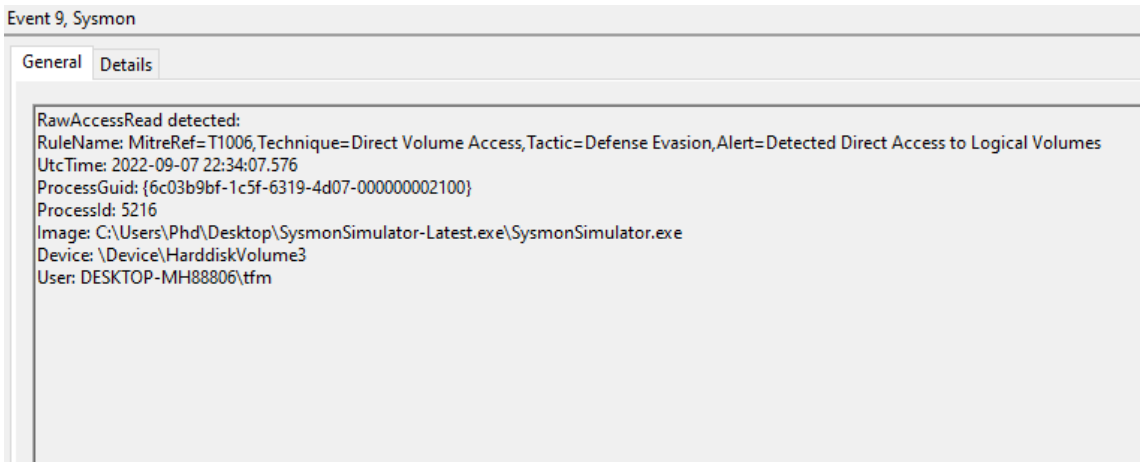

Ilustración 45. Movimiento lateral mediante acceso directo (parte 3)

6.2.8. ACCESO NO AUTORIZADO A PROCESO

Para esta prueba se utiliza la ID 10 de *SysmonSimulator*. Se incluye el código pertinente:

```
void processaccess10(int process_id) {

    printf("[+] ProcessAccess: Process ID %lu\n", process_id);
    HANDLE hProcessToAccess = OpenProcess(PROCESS_QUERY_INFORMATION | PROCESS_VM_READ, 0, process_id);
    if (hProcessToAccess) {
        printf("[+] Successful   : Process handle was opened\n");
        CloseHandle(hProcessToAccess);
    }
    else
    {
        printf("Error code is : %lu\n", GetLastError());
    }
}
```

Ilustración 46. Acceso no autorizado a proceso (parte 1)

La explicación es simple: se abre un proceso a partir del mismo proceso que está lanzando. Se eleva una alerta para todas las partes correspondientes, y además, se mata al proceso padre e hijos:

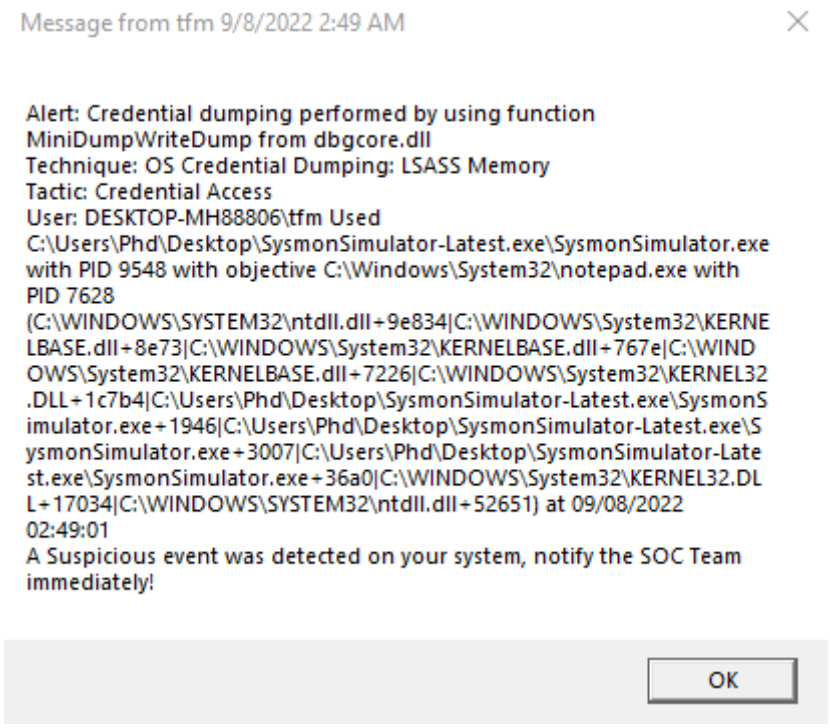

Ilustración 47. Acceso no autorizado a proceso (parte 2)

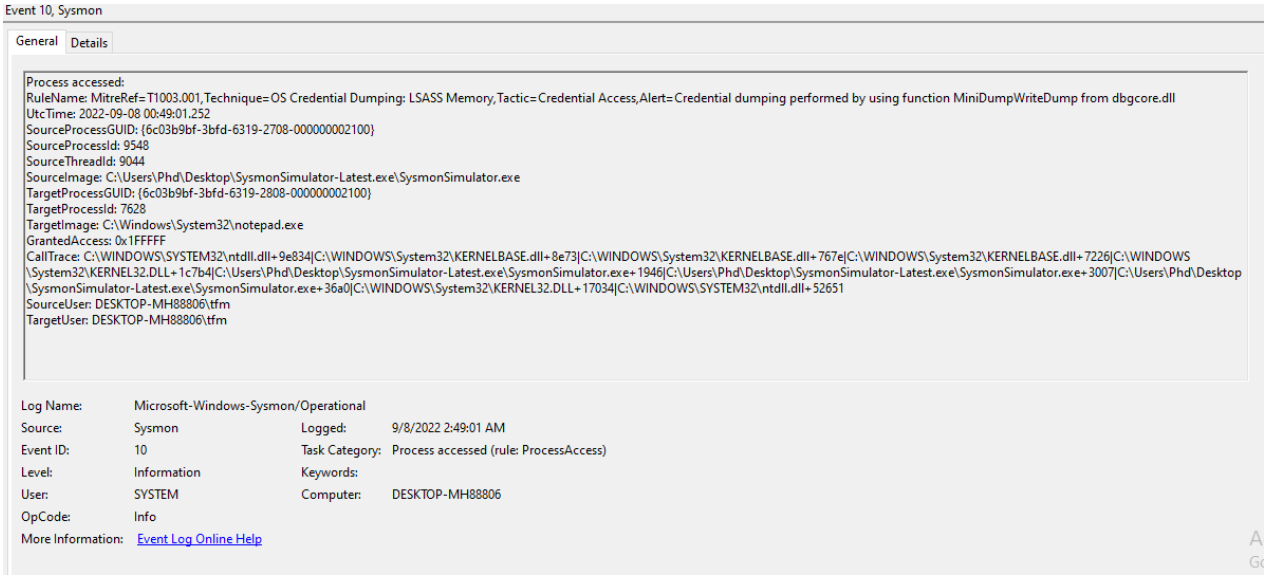

Ilustración 48. Acceso no autorizado a proceso (parte 3)

6.2.9. CREACIÓN DE FICHEROS NO AUTORIZADOS

Para esta prueba, se utiliza la ID 11 de *SysmonSimulator*. El código pertinente es el siguiente:

```
void fileCreate11() {
    HANDLE hFile = CreateFileW(
        L"NewFile.bat",
        GENERIC_WRITE,
        FILE_SHARE_READ,
        NULL,
        CREATE_ALWAYS,
        FILE_ATTRIBUTE_NORMAL,
        NULL);

    if (hFile == INVALID_HANDLE_VALUE)
    {
        printf("[!] Error creating file. Error code is : %lu\n", GetLastError());
    }
    else {
        printf("[+] Created File : NewFile.bat\n");
        CloseHandle(hFile);
    }
}
```

Ilustración 49. Creación de ficheros no autorizados (parte 1)

A partir del proceso empleado, se crea un fichero de tipo *bat* bajo el nombre *NewFile*. El EDR generará una alerta y eliminará dicho fichero hasta que sea revisado. Además, será analizado por las reglas *YARA* del sistema.

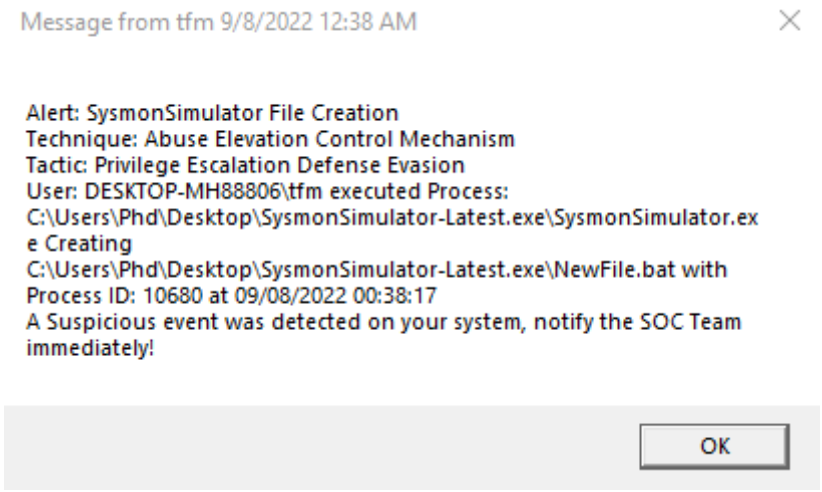

Ilustración 50. Creación de ficheros no autorizados (parte 2)

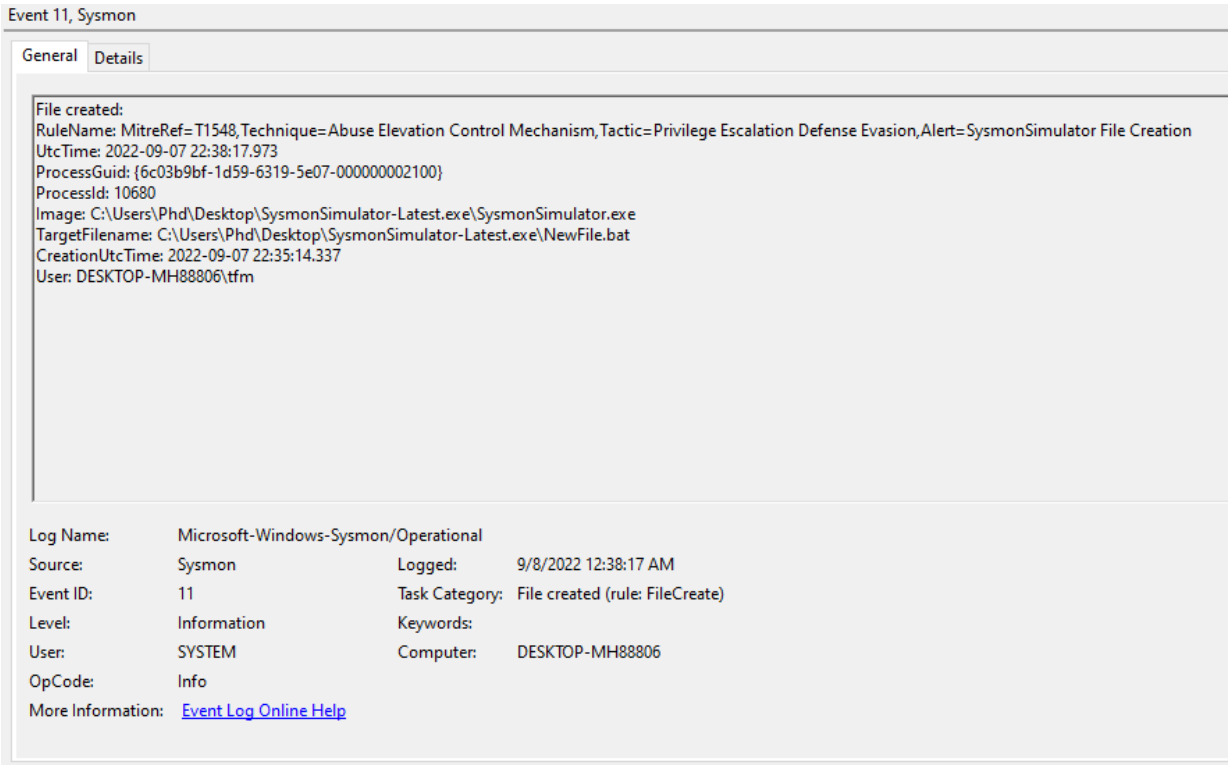

Ilustración 51. Creación de ficheros no autorizados (parte 3)

6.2.10. REGISTROS NO AUTORIZADOS

En lo referente a los registros, como ya se ha dicho anterioridad, se dividen en tres tipos: creación y eliminación no autorizadas, establecimiento de valor no autorizado o modificación

de clave y valor no autorizado. Las pruebas utilizadas se irán incluyen en cada subapartado correspondiente:

6.2.10.1. CREACIÓN Y ELIMINACIÓN

Para la prueba, se ha utilizado la ID 12 de *SysmonSimulator*. El código pertinente se incluye a continuación:

```
BOOL CreateRegistryKey()
{
    HKEY  hKey = NULL;
    if (RegCreateKeyA(HKEY_CURRENT_USER, "TestSysmon", &hKey) != ERROR_SUCCESS) {
        printf("[!] Error opening or creating key. Error code is : %lu\n", GetLastError());
        return FALSE;
    }
    else {
        RegCloseKey(hKey);
        return TRUE;
    }
}
```

Ilustración 52. Registros no autorizados - creación y eliminación (parte 1)

```
void registryEvent12()
{
    if(CreateRegistryKey()) {
        printf("[+] Successful   : Registry object created\n");
    }
    else {
        printf("Error code is : %lu\n", GetLastError());
    }
}
```

Ilustración 53. Registros no autorizados - creación y eliminación (parte 2)

Se resume la explicación del código. En la primera imagen se realiza un método estático el cual crea una nueva clave de registro bajo el usuario actual, clave con nombre *TestSysmon* y valor nulo. Este método es llamado en la imagen número dos. El resultado es la creación de una clave de registro no autorizada a través de un proceso. Se matará el proceso pertinente, y se generará una alerta al usuario *endpoint* y al SOC, el cual recibirá el nombre de la nueva clave creada para erradicarla de manera inmediata.

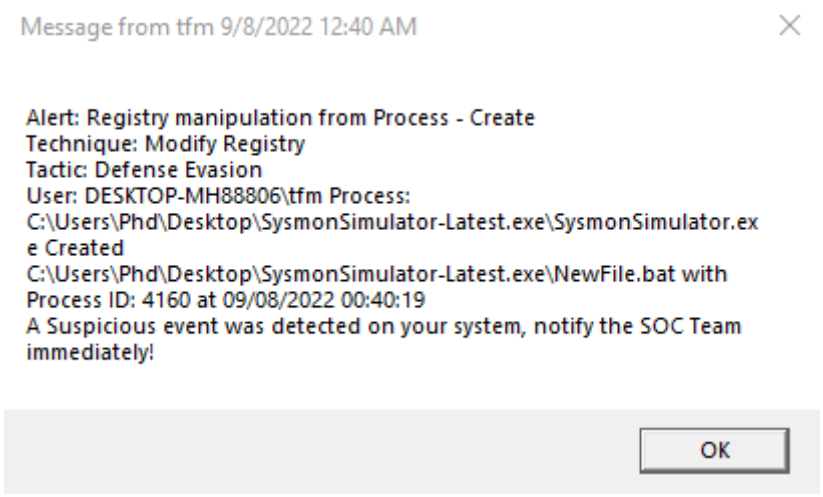

Ilustración 54. Registros no autorizados - creación y eliminación (parte 3)

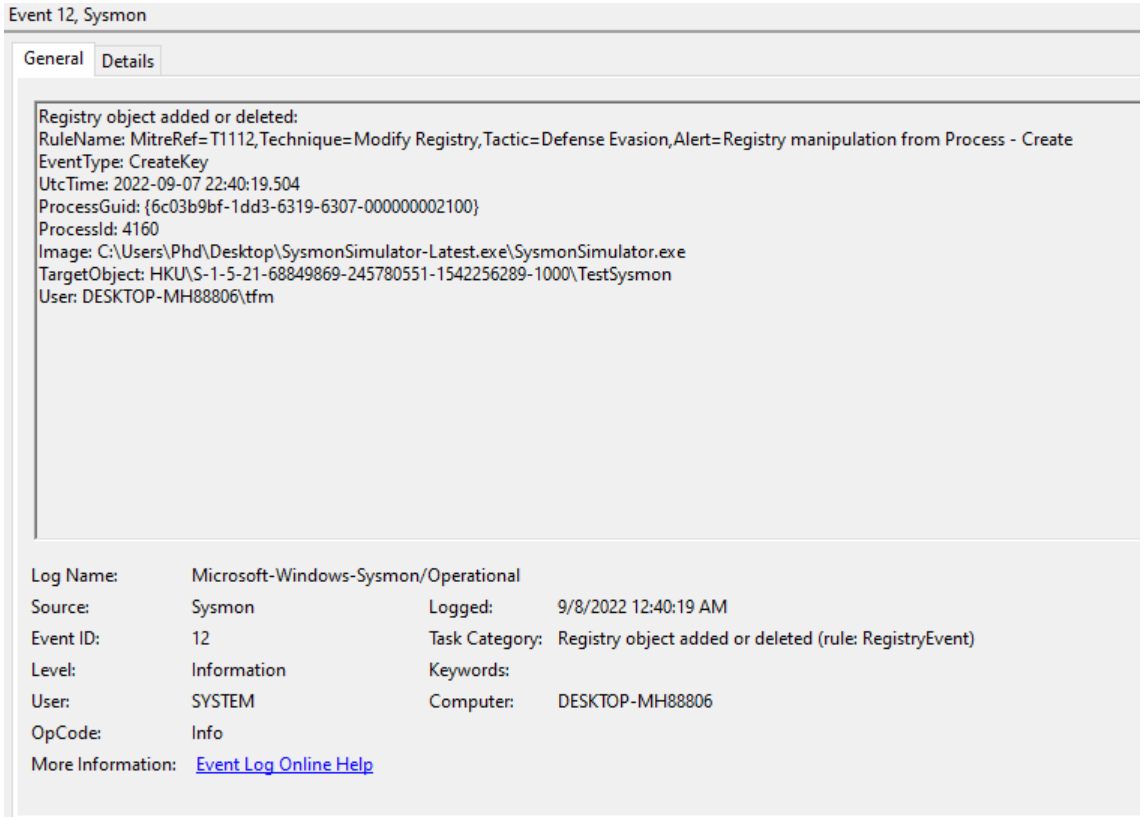

Ilustración 55. Registros no autorizados - creación y eliminación (parte 4)

6.2.10.2. ESTABLECIMIENTO DE VALOR

Para la prueba, se ha utilizado la ID 11 de *SysmonSimulator*. En cuanto al código, se presenta en la siguiente imagen:

```
BOOL CreateRegistryKey()
{
    HKEY  hKey = NULL;
    if (RegCreateKeyA(HKEY_CURRENT_USER, "TestSysmon", &hKey) != ERROR_SUCCESS) {
        printf("[!] Error opening or creating key. Error code is : %lu\n", GetLastError());
        return FALSE;
    }
    else {
        RegCloseKey(hKey);
        return TRUE;
    }
}

BOOL writeStringInRegistry()
{
    HKEY hKey = NULL;
    if (RegOpenKeyEx(HKEY_CURRENT_USER, L"TestSysmon", 0, KEY_WRITE, &hKey) == ERROR_SUCCESS)
    {
        if (ERROR_SUCCESS != RegSetValueEx(hKey, L"Message", 0, REG_SZ, (LPBYTE)(L"Testing"), ((((DWORD)strlen("Tested") + 1)) * 2)))
        {
            RegCloseKey(hKey);
            printf("FALSE.\n");
            return FALSE;
        }
        RegCloseKey(hKey);
        return TRUE;
    }
    return FALSE;
}
```

Ilustración 56. Registros no autorizados - establecimiento de valor (parte 1)

```
void registryEvent13()
{
    HKEY subKey = NULL;
    LONG result = RegOpenKeyEx(HKEY_CURRENT_USER, L"TestSysmon", 0, KEY_READ, &subKey);
    if (result != ERROR_SUCCESS) {
        CreateRegistryKey();
    }
    if (writeStringInRegistry()) {
        printf("[+] Successful   : Registry value modified to 'Tested'\n");
    }
    else {
        printf("Error code is : %lu\n", GetLastError());
    }
}
```

Ilustración 57. Registros no autorizados - establecimiento de valor (parte 2)

En la primera imagen, aparte del método de la ID anterior, se recoge un nuevo método que registra un valor '0' a la clave anteriormente creada *TestSysmon*. Dicho de otra manera, se le otorga un valor a una clave de los registros del sistema. En la segunda imagen, se almacena en un valor de tipo *long*, el resultado de leer la clave y comprobar que tiene el valor '0'. Si no es así, se crea el registro (sin valor). Al igual que en el caso anterior, se mata al proceso pertinente, se

eleva una alerta al SOC con los datos que se han modificado en registro, y por último, se aisla el equipo afectado.

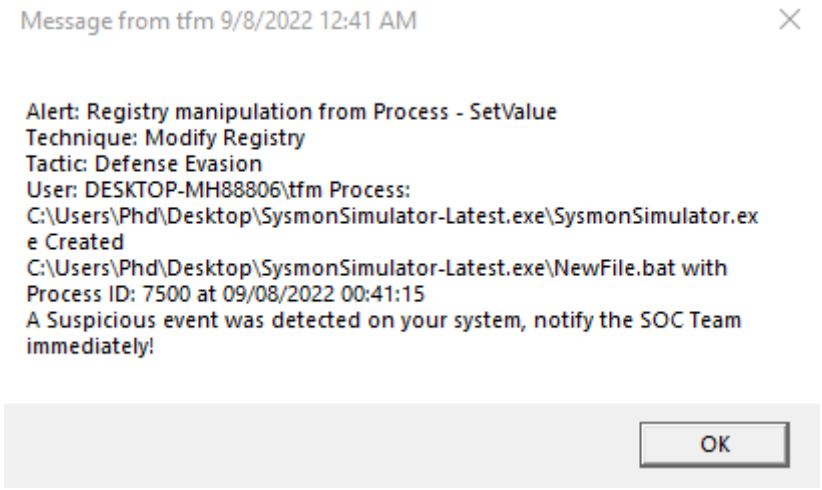

Ilustración 58. Registros no autorizados - establecimiento de valor (parte 3)

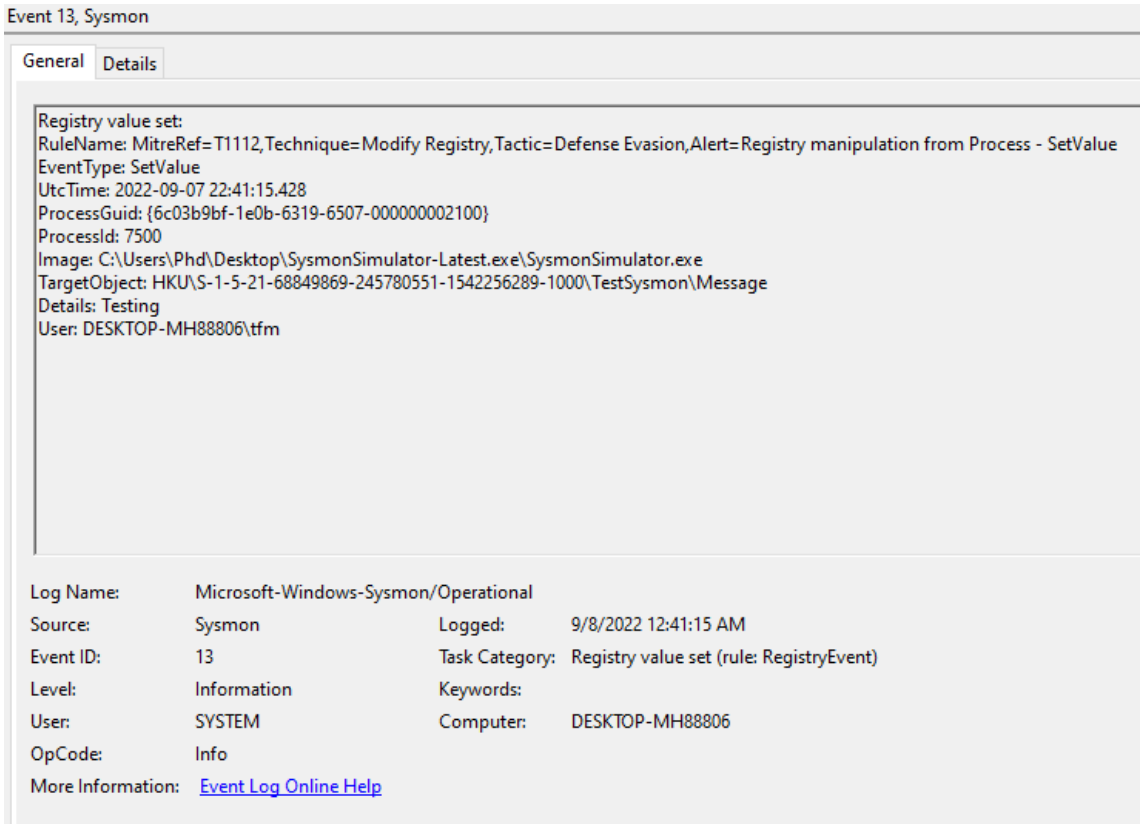

Ilustración 59. Registros no autorizados - establecimiento de valor (parte 4)

6.2.10.3. MODIFICACIÓN DE CLAVE Y VALOR

Para esta prueba, se ha utilizado el ID 14 de *SysmonSimulator*. Se incluye el código pertinente a dicha prueba:

```
void registryEvent14()
{
    HKEY  hKey = NULL;
    if (RegCreateKeyA(HKEY_CURRENT_USER, "NewRegistrySysmonTesting", &hKey) != ERROR_SUCCESS) {
        printf("[!] Error opening or creating key. Error code is : %lu\n", GetLastError());
    }
    else {
        RegCloseKey(hKey);
    }

    RegRenameKey(
        HKEY_CURRENT_USER,
        L"NewRegistrySysmonTesting",
        L"RegistrySysmonTestingRenamed"
    );
}
```

Ilustración 60. Registros no autorizados - modificación de clave y valor (parte 1)

Se crea o se abre el registro con nombre de clave *NewRegistrySysmonTesting*. Una vez se realiza dicha acción; se modifica el nombre de *NewRegistrySysmonTesting* a *RegistrySysmonTestingRenamed*. Al igual que en el caso anterior, se mata al proceso pertinente, se eleva una alerta al SOC con los datos que se han modificado en registro, y por último, se aisla el equipo afectado.

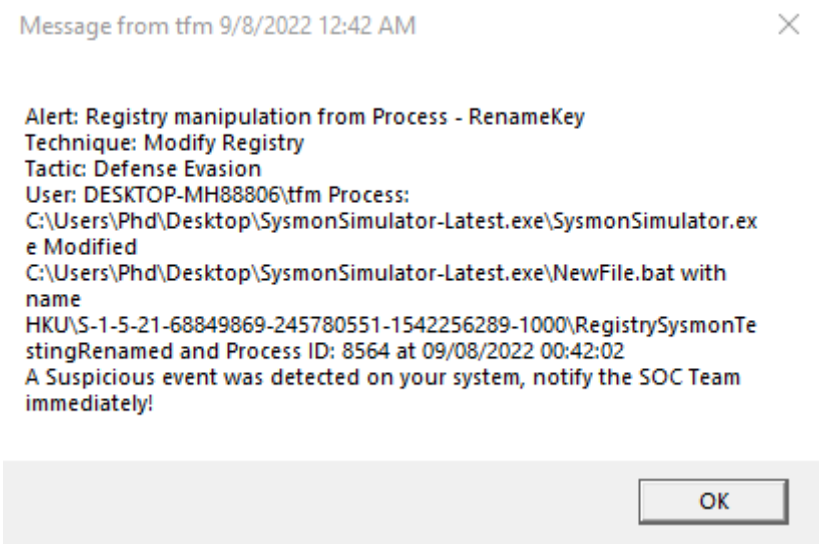

Ilustración 61. Registros no autorizados - modificación de clave y valor (parte 2)

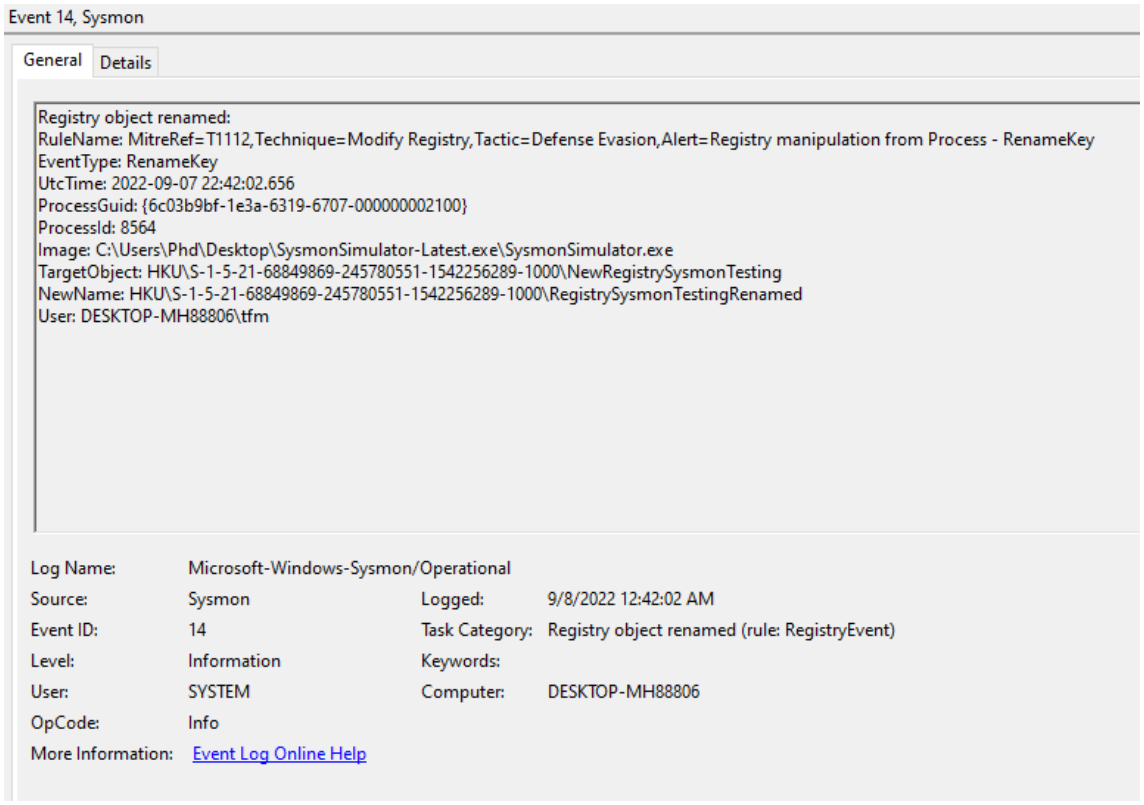

Ilustración 62. Registros no autorizados - modificación de clave y valor (parte 3)

6.2.11. DETECCIÓN DE FICHERO MALICIOSO A TRAVÉS DE FLUJO DE HASH

Para esta prueba, se ha utilizado la ID 15 de *SysmonSimulator*. Se presenta a continuación, una imagen donde se observa el código correspondiente:

```
void fileCreateStreamHash15() {
    DWORD dwRet = 0;
    char testdata[] = "Hello World";
    WIN32_FIND_STREAM_DATA streaminfo = { 0 };
    HANDLE hFile = CreateFileA("Streamfile.txt:SysmonStream", GENERIC_WRITE, FILE_SHARE_WRITE, NULL, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
    if (hFile == INVALID_HANDLE_VALUE) {
        printf("[!] Could not create stream for file Streamfile.txt\n");
        printf("Error code is : %lu\n", GetLastError());
    }
    else {
        printf("[+] Successful   : Created stream for file Streamfile.txt\n");
        if (!WriteFile(hFile, "Sysmon simulator has written in ADS SysmonStream of Streamfile.txt", 67, &dwRet, NULL)) {
            printf("Error code is : %lu\n", GetLastError());
            CloseHandle(hFile);
        }
    }
}
```

Ilustración 63. Detección de fichero malicioso a través de flujo de hash (parte 1)

Eleva un buscador de flujo para sistemas *Windows*, y al mismo tiempo crea un fichero *Streamfile.txt* con flujos alternativos de datos asociados. La solución *EDR* detectará el comportamiento de este fichero previo a que llegue a tocar disco. Esto es debido a que analiza el *hash* que va acompañado a la completitud del fichero, y, cuando se reconoce algo que encaje

en las reglas de *Sysmon* respecto a la ID, lo elimina. Utilizar los flujos de datos de esta manera es algo común utilizado por variantes de *malware* donde el flujo alternativo de información esconde ejecutables maliciosos para ocultar información. En este caso, se detiene el proceso y se hace un volcado de memoria de éste; además de elevar las alertas adecuadas a usuario *endpoint* y equipo de SOC.

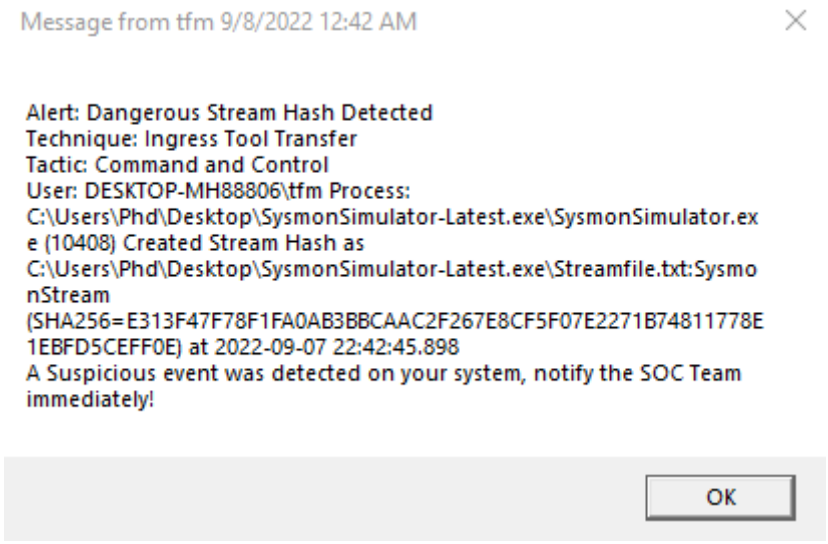

Ilustración 64. Detección de fichero malicioso a través de flujo de hash (parte 2)

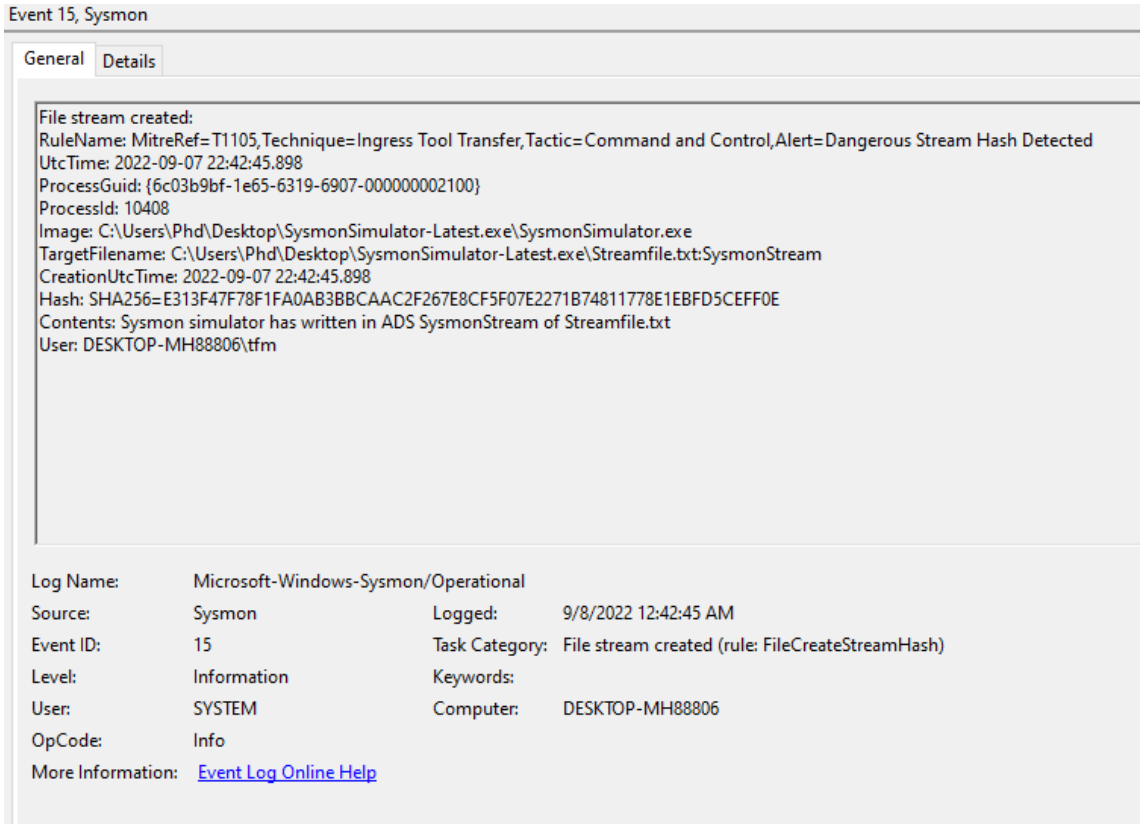

Ilustración 65. Detección de fichero malicioso a través de flujo de hash (parte 3)

6.2.12. MOVIMIENTO LATERAL MEDIANTE PIPES

Para esta sección, se tienen en cuenta dos subapartados: creación y conectividad de *pipes*. Éstas son usadas para la conexión entre interprocesos en sistemas operativos *Windows*, pero también soporta la comunicación entre dos procesos en máquinas separadas mediante el protocolo SMB.

6.2.12.1. CREACIÓN DE PIPES

Para esta prueba, se ha utilizado la ID 17 de *SysmonSimulator.* Se incluye a continuación el código correspondiente:

```
void pipeCreated17() {
    HANDLE hPipe = NULL;
    SECURITY_ATTRIBUTES secAttrib = { 0 };
    static LPCSTR lpName = "\\\\.\\pipe\\sysmontestnamedpipe";
    hPipe = CreateNamedPipeA(lpName,
        PIPE_ACCESS_DUPLEX | FILE_FLAG_OVERLAPPED,
        PIPE_TYPE_BYTE | PIPE_WAIT,
        10,
        2048,
        2048,
        0,
        &secAttrib);
    if (hPipe == INVALID_HANDLE_VALUE)
    {
        printf("CreateNamedPipeA(): FAILED.Error code is : %lu\n", GetLastError());
    }
    else {
        printf("[+] Successful   : Pipe %s has been created\n", lpName);
        LocalFree(hPipe);
        CloseHandle(hPipe);
    }
}
```

Ilustración 66. Movimiento lateral mediante pipes - Creación de pipes (parte 1)

A través del proceso asociado, se realiza la creación de una *pipe* bajo el nombre de *Sysmontestnamedpipe*, a la que se le asignan 10 instancias máximas, y los tamaños máximos de *buffer* de entrada y salida, siendo 2048 *bits*. Poco hay que señalar, se crea una *pipe* a través de un proceso no autorizado. Tras ser detectado, se matará el proceso padre e hijo y se reinicia el sistema. Además, se generan las alertas pertinentes para usuario *endpoint* y grupo de SOC.

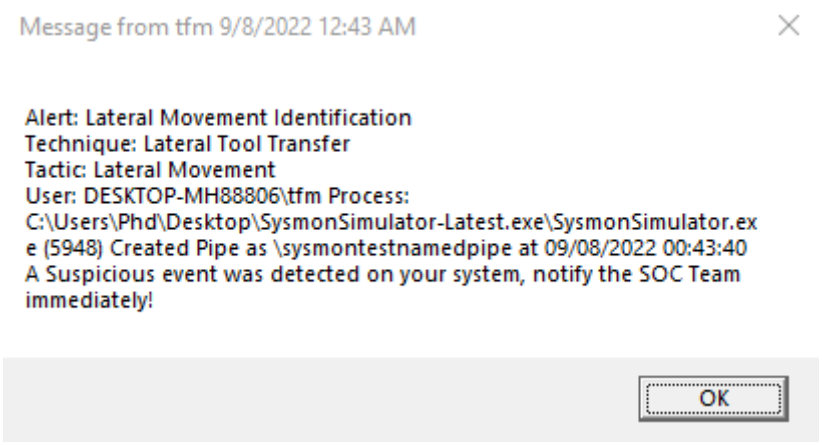

Ilustración 67. Movimiento lateral mediante pipes - Creación de pipes (parte 2)

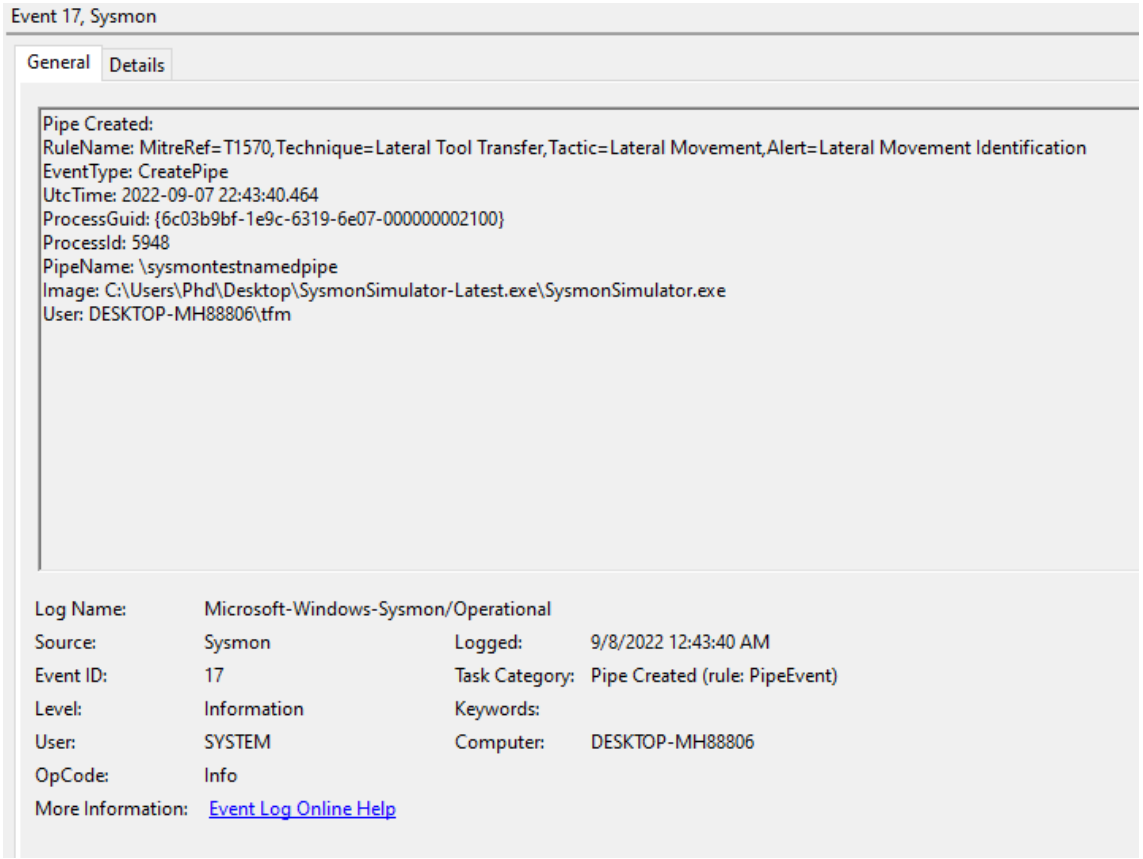

Ilustración 68. Movimiento lateral mediante pipes - Creación de pipes (parte 3)

6.2.12.2. CONECTIVIDAD DE PIPES

Para esta prueba, se ha utilizado la IP 18 de *SysmonSimulator*. Se incluye a continuación, el código relevante:

```
void pipeConnect18() {
    HANDLE hPipe = NULL;
    HANDLE hConnectPipe = NULL;
    SECURITY_ATTRIBUTES secAttrib = { 0 };
    LPCSTR lpName = "\\\\.\\pipe\\sysmontestconnectpipe";

    hPipe = CreateNamedPipeA(lpName,
        PIPE_ACCESS_DUPLEX | FILE_FLAG_OVERLAPPED,
        PIPE_TYPE_BYTE | PIPE_WAIT,
        10,
        2048,
        2048,
        0,
        &secAttrib);
    if (hPipe == INVALID_HANDLE_VALUE)
    {
        printf("CreateNamedPipeA(): FAILED.Error code is : %lu\n", GetLastError());
    }
    else {
        printf("[+] Successful   : Pipe connection event created for pipe %s \n", lpName);
    }
    hConnectPipe = CreateFileA(lpName, GENERIC_READ | GENERIC_WRITE | SYNCHRONIZE, 0, NULL, OPEN_EXISTING, FILE_FLAG_WRITE_THROUGH, NULL);
    if (hConnectPipe == INVALID_HANDLE_VALUE) {
        printf("Error: %lu\n", GetLastError());
    }
    else {
        if (hPipe != 0) {
            LocalFree(hPipe);
            CloseHandle(hPipe);
        }
        if (hConnectPipe != 0) {
            CloseHandle(hConnectPipe);
        }
    }
}
```

Ilustración 69. Movimiento lateral mediante pipes - Conectividad de pipes (parte 1)

Resumidamente ocurre lo siguiente: se crea una primera *pipe* unida al registro de *pipes* bajo el nombre *sysmontestconnectpipe* que permite el acceso a 10 instancias con un tamaño de *buffer* de entrada y salida de 2048 bits. A continuación se crea una segunda *pipe*, bajo el mismo nombre. Ambas *pipe* realizan una conexión a partir de un proceso no autorizado. Al igual que en el caso anterior, tras ser detectado, se matará el proceso padre e hijo y se reinicia el sistema. Además, se generan las alertas pertinentes para usuario *endpoint* y grupo de SOC.

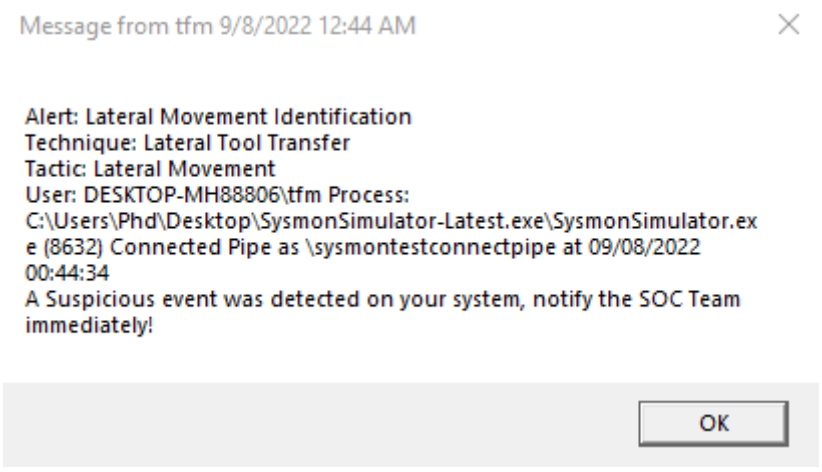

Ilustración 70. Movimiento lateral mediante pipes - Conectividad de pipes (parte 2)

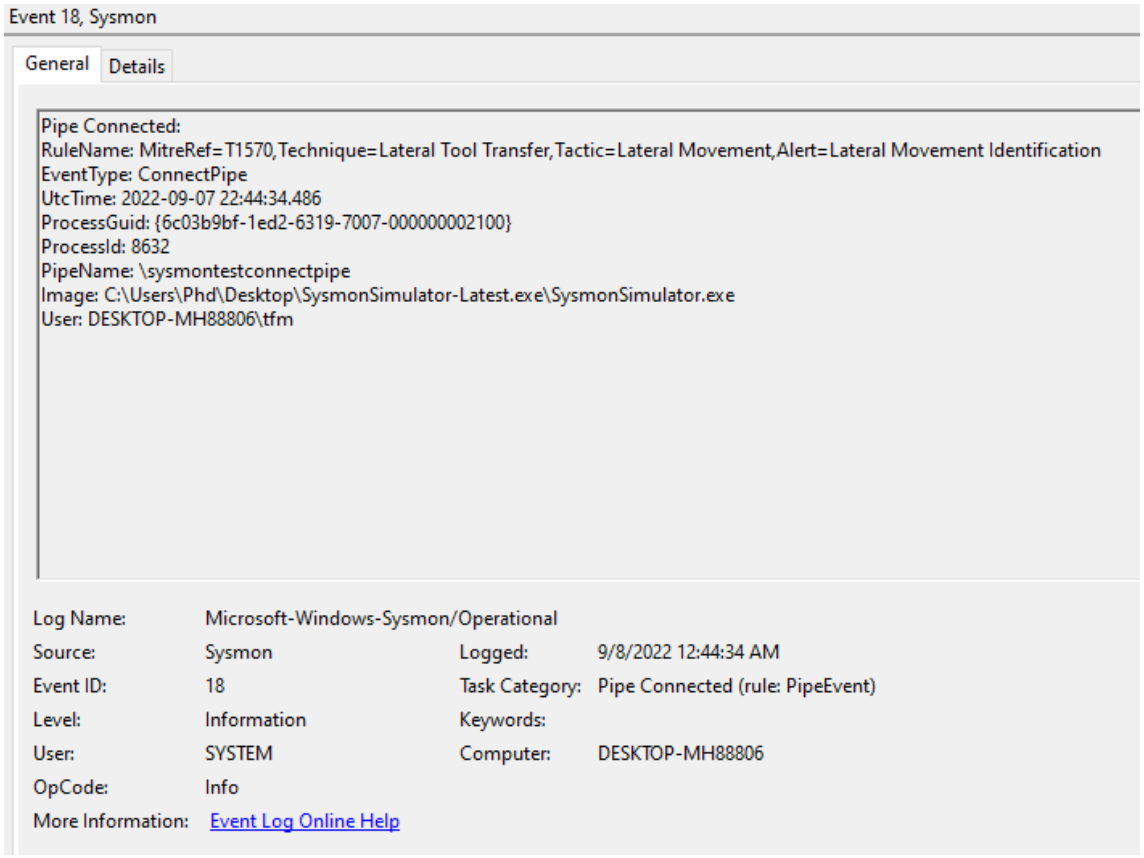

Ilustración 71. Movimiento lateral mediante pipes - Conectividad de pipes (parte 3)

6.2.13. CONSULTA DNS NO AUTORIZADA

Para la prueba, se utiliza la ID 22 de *SysmonSimulator*. Se incluye el código pertinente:

```
void dnsquery22() {
    DWORD response = 0;
    PDNS_RECORD base = NULL;
    DWORD options = DNS_QUERY_WIRE_ONLY;
    PIP4_ARRAY pSrvList = NULL;
    unsigned short wType = 0;

    response = DnsQuery_A("google.com", wType, options, pSrvList, &base, NULL);
    if (response) {
        printf("[+] Successful   : Performed DNS Lookup for 'google.com' \n");
    }
    else {
        printf("[+] Tried to perform lookup for domain 'google.com' but got an error \n");
        printf("[!] Error code is: %lu\n", GetLastError());
    }
}
```

Ilustración 72. Consulta DNS no autorizada (parte 1)

En este caso, el método intenta hacer su propia resolución de dominio, en lugar de utilizar la facilitada por la *Windows API* del sistema. Este comportamiento es sospechoso, ya que entre otras acciones, un atacante puede resolver a un dominio propio donde ejecutar herramientas maliciosas sin ser detectado. En cuanto la solución detecta este tipo de eventos; mata el proceso y además, aisla el sistema. Por último, genera las alerts pertinentes.

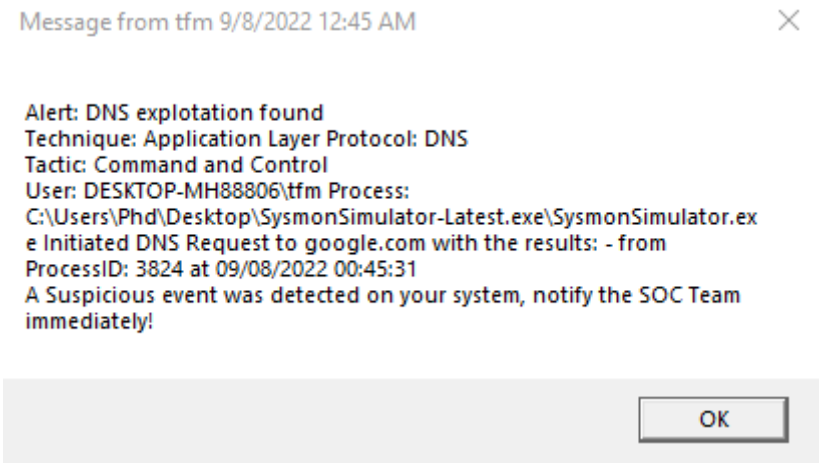

Ilustración 73. Consulta DNS no autorizada (parte 2)

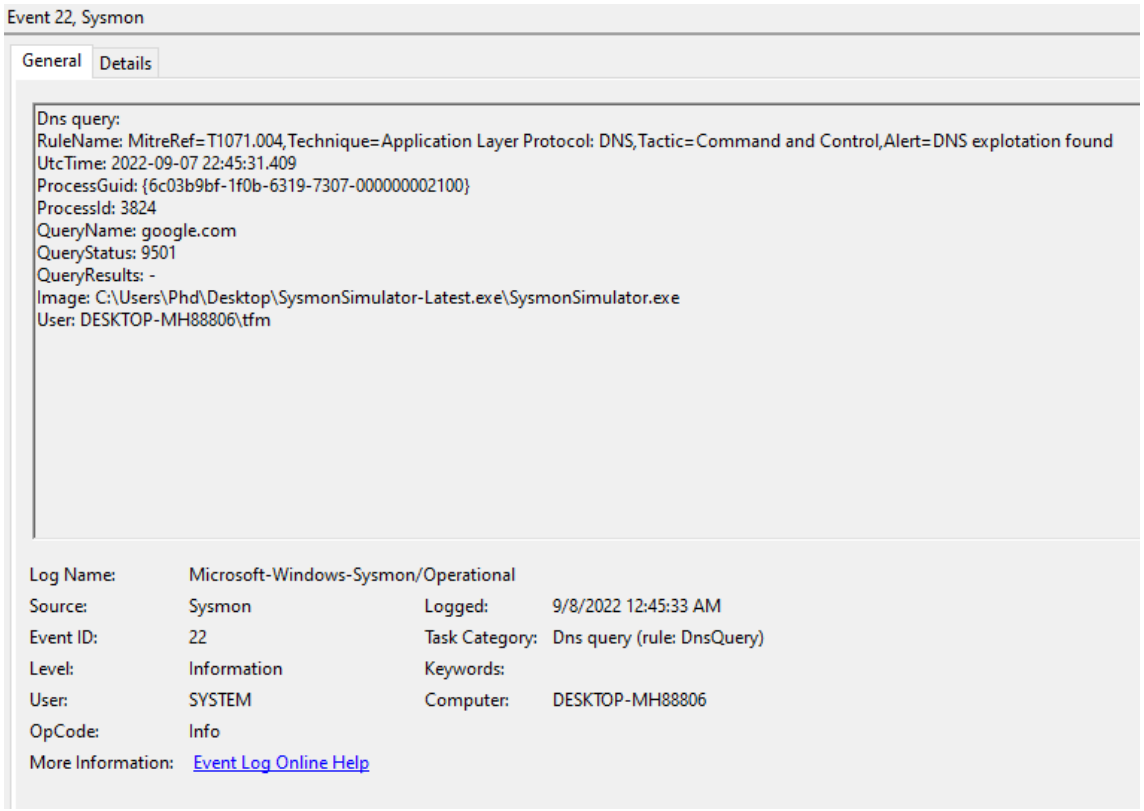

Ilustración 74. Consulta DNS no autorizada (parte 3)

6.2.14. ELIMINACIÓN DE FICHEROS - ALMACENAMIENTO EN CUARENTENA

Para esta prueba, se han utilizado comandos en la consola de *Powershell*, concretamente:

```
dir > test.txt

del test.txt
```

Que de forma muy simple, crea un fichero bajo el nombre de *test.txt* en el que se almacena la información del directorio, y posteriormente lo elimina utilizando el comando *del*, el cual invoca a *Remove-Item*. Esta funcionalidad elimina de manera no autorizada un fichero desde un proceso concreto (en este caso, *Powershell.exe*). El fichero eliminado es restaurado al sistema, se mata el proceso que ha realizado la eliminación y se generan las alertas correspondientes.

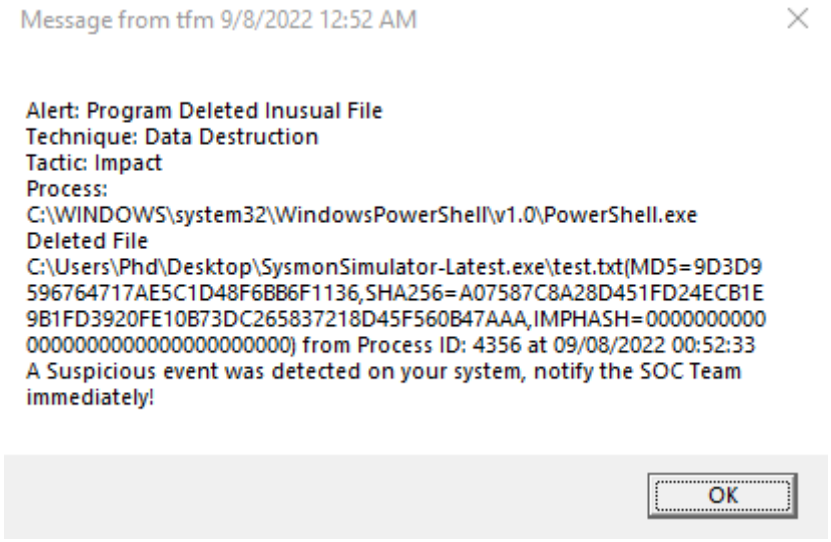

Ilustración 75. Eliminación de ficheros - almacenamiento en cuarentena (parte 1)

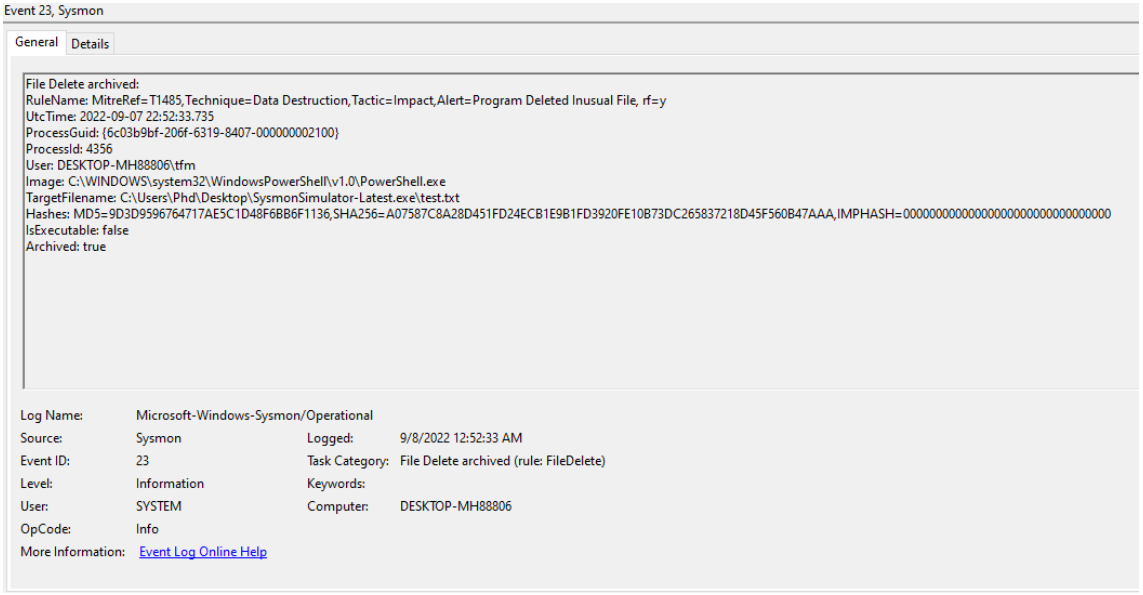

Ilustración 76. Eliminación de ficheros - almacenamiento en cuarentena (parte 2)

6.2.15. COPIA DE CONTRASEÑAS, USUARIOS, ETC. EN PORTAPAPELES

En este caso, no se genera alerta al usuario *endpoint*, pero si que crea el evento, y se envía la notificación al SOC. Se utiliza como medida adicional de monitorización por si, el propio usuario del equipo realizara actos ilegítimos con datos sensibles. Para estas pruebas, se ha copiado información directamente de un *Notepad++* que había abierto en el equipo. En caso de que el *ClientInfo* y *User* no coincidiesen, se aislaría el equipo.

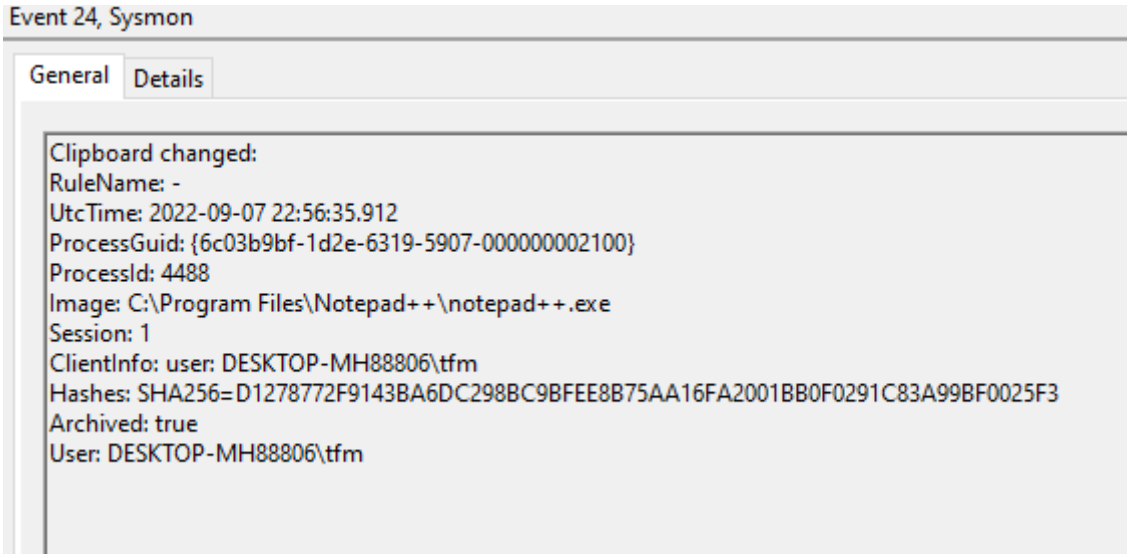

Ilustración 77. Copia de contraseñas, usuarios, etc. en portapapeles

6.2.16. TAMPER DE PROCESOS - HOLLOWING

Para esta prueba, se ha utilizado la ID 25 de *SysmonSimulator*. Se presenta a continuación, el código:

```
int processTampering25()
{
    PIMAGE_DOS_HEADER pDosH;
    PIMAGE_NT_HEADERS pNtH;
    PIMAGE_SECTION_HEADER pSecH;
    PVOID image, mem, base;
    DWORD i, read, nSizeOfFile;
    HANDLE hFile;

    STARTUPINFOA si;
    PROCESS_INFORMATION pi;
    CONTEXT ctx;

    ctx.ContextFlags = CONTEXT_FULL;

    memset(&si, 0, sizeof(si));
    memset(&pi, 0, sizeof(pi));

    LPSTR replacement = "c:\\windows\\system32\\cmd.exe";
    LPSTR targetExe = "c:\\Windows\\System32\\svchost.exe";

    if (!CreateProcessA(NULL, replacement, NULL, NULL, FALSE, CREATE_SUSPENDED, NULL, NULL, &si, &pi))
    {
        printf("\nNot able to run the target executable. Error code is : % lu\n", GetLastError());
        return 1;
    }

    hFile = CreateFileA(targetExe, GENERIC_READ, FILE_SHARE_READ, NULL, OPEN_EXISTING, 0, NULL);

    if (hFile == INVALID_HANDLE_VALUE)
    {
        printf("\nNot able to open the replacement executable. Error code is : % lu\n", GetLastError());
        NtTerminateProcess(pi.hProcess, 1);
        return 1;
    }

    nSizeOfFile = GetFileSize(hFile, NULL);
    image = VirtualAlloc(NULL, nSizeOfFile, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);

    if (!ReadFile(hFile, image, nSizeOfFile, &read, NULL))
    {
        printf("\nNot able to read the replacement executable. Error code is : % lu\n", GetLastError());
        NtTerminateProcess(pi.hProcess, 1);
        return 1;
    }

    NtClose(hFile);
    pDosH = (PIMAGE_DOS_HEADER)image;

    if (pDosH->e_magic != IMAGE_DOS_SIGNATURE)
    {
        printf("\nError: Invalid executable format.\n");
```

Ilustración 78. Process Tampering – Hollowing (parte 1)

```
        printf("\nError: Invalid executable format.\n");
        NtTerminateProcess(pi.hProcess, 1);
        return 1;
    }

    pNtH = (PIMAGE_NT_HEADERS)((LPBYTE)image + pDosH->e_lfanew);
    NtGetContextThread(pi.hThread, &ctx);

#ifdef _WIN64
    NtReadVirtualMemory(pi.hProcess, (PVOID)(ctx.Rdx + (sizeof(SIZE_T) * 2)), &base, sizeof(PVOID), NULL);
#endif

#ifdef _X86_
    NtReadVirtualMemory(pi.hProcess, (PVOID)(ctx.Ebx + 8), &base, sizeof(PVOID), NULL);
#endif
    if ((SIZE_T)base == pNtH->OptionalHeader.ImageBase)
    {
        printf("\nUnmapping original executable image from child process. Address: %#zx\n", (SIZE_T)base);
        NtUnmapViewOfSection(pi.hProcess, base);
    }

    mem = VirtualAllocEx(pi.hProcess, (PVOID)pNtH->OptionalHeader.ImageBase, pNtH->OptionalHeader.SizeOfImage, MEM_COMMIT | MEM_RESERVE, PAGE_EXECUTE_READWRITE);

    if (!mem)
    {
        printf("\nError: Unable to allocate memory in child process. VirtualAllocEx failed with error %lu\n", GetLastError());

        NtTerminateProcess(pi.hProcess, 1);
        return 1;
    }

    NtWriteVirtualMemory(pi.hProcess, mem, image, pNtH->OptionalHeader.SizeOfHeaders, NULL);

    for (i = 0; i < pNtH->FileHeader.NumberOfSections; i++)
    {
        pSecH = (PIMAGE_SECTION_HEADER)((LPBYTE)image + pDosH->e_lfanew + sizeof(IMAGE_NT_HEADERS) + (i * sizeof(IMAGE_SECTION_HEADER)));
        NtWriteVirtualMemory(pi.hProcess, (PVOID)((LPBYTE)mem + pSecH->VirtualAddress), (PVOID)((LPBYTE)image + pSecH->PointerToRawData), pSecH->SizeOfRawData, NULL);
    }


#ifdef _WIN64
    ctx.Rcx = (SIZE_T)((LPBYTE)mem + pNtH->OptionalHeader.AddressOfEntryPoint);
    NtWriteVirtualMemory(pi.hProcess, (PVOID)(ctx.Rdx + (sizeof(SIZE_T) * 2)), &pNtH->OptionalHeader.ImageBase, sizeof(PVOID), NULL);
#endif

#ifdef _X86_
    ctx.Eax = (SIZE_T)((LPBYTE)mem + pNtH->OptionalHeader.AddressOfEntryPoint);
    NtWriteVirtualMemory(pi.hProcess, (PVOID)(ctx.Ebx + (sizeof(SIZE_T) * 2)), &pNtH->OptionalHeader.ImageBase, sizeof(PVOID), NULL);
#endif

    NtSetContextThread(pi.hThread, &ctx);
    NtResumeThread(pi.hThread, NULL);
    NtWaitForSingleObject(pi.hProcess, FALSE, NULL);
```

Ilustración 79. Process Tampering – Hollowing (parte 2)

```
    NtSetContextThread(pi.hThread, &ctx);
    NtResumeThread(pi.hThread, NULL);
    NtWaitForSingleObject(pi.hProcess, FALSE, NULL);
    NtClose(pi.hThread);
    NtClose(pi.hProcess);
    if (image) {
        VirtualFree(image, 0, MEM_RELEASE);
        printf("[+] Successful\n");
    }
    else {
        printf("Error: %lu\n", GetLastError());
    }
    return 0;
}
```

Ilustración 80. Process Tampering – Hollowing (parte 3)

Los pasos que se realizan se pueden resumir en los siguientes puntos:

1) La aplicación objetivo comienza usando una instancia de *cmd.exe* en estado de suspensión.
2) El proceso abre el ejecutable a sustituir, en este caso, *svchost.exe*.
3) Obtiene el tamaño de este ejecutable por el que sustituir.
4) Aloja memoria para el ejecutable, utilizando *alloc*.
5) Lee el fichero ejecutable desde disco.
6) Desvincula la imagen ejecutable.
7) Escribe el ejecutable a reemplazar en la aplicación objetivo; usando escritura en memoria dinámica mediante la API nativa.
8) Escribe la nueva dirección del ejecutable en la pila, y marca el registro adecuado en el hilo principal para cuadrar con el del ejecutable a reemplazar.
9) Reinicia el hilo.

Este método es muy peligroso y permite ataques minuciosos. En este caso, la solución lo detecta; mata los procesos e inyecciones de hilos realizadas, crea una regla de *firewall* contra la imagen, realiza un *dump* de memoria del proceso y aisa el sistema. Adicionalmente, envía las alertas pertinentes.

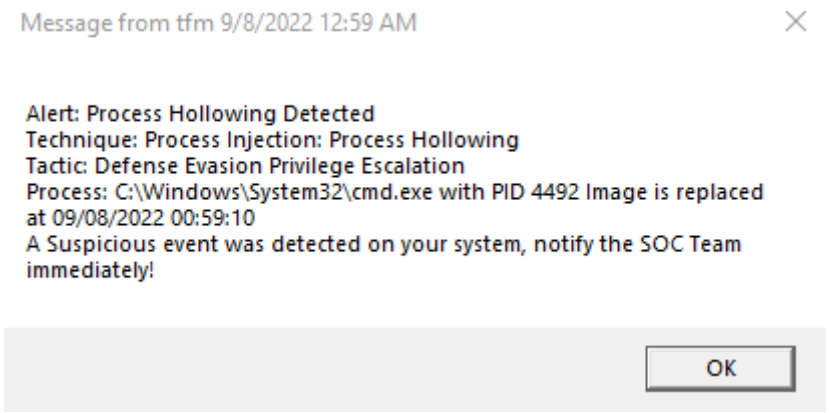

Ilustración 81. Process Tampering – Hollowing (parte 4)

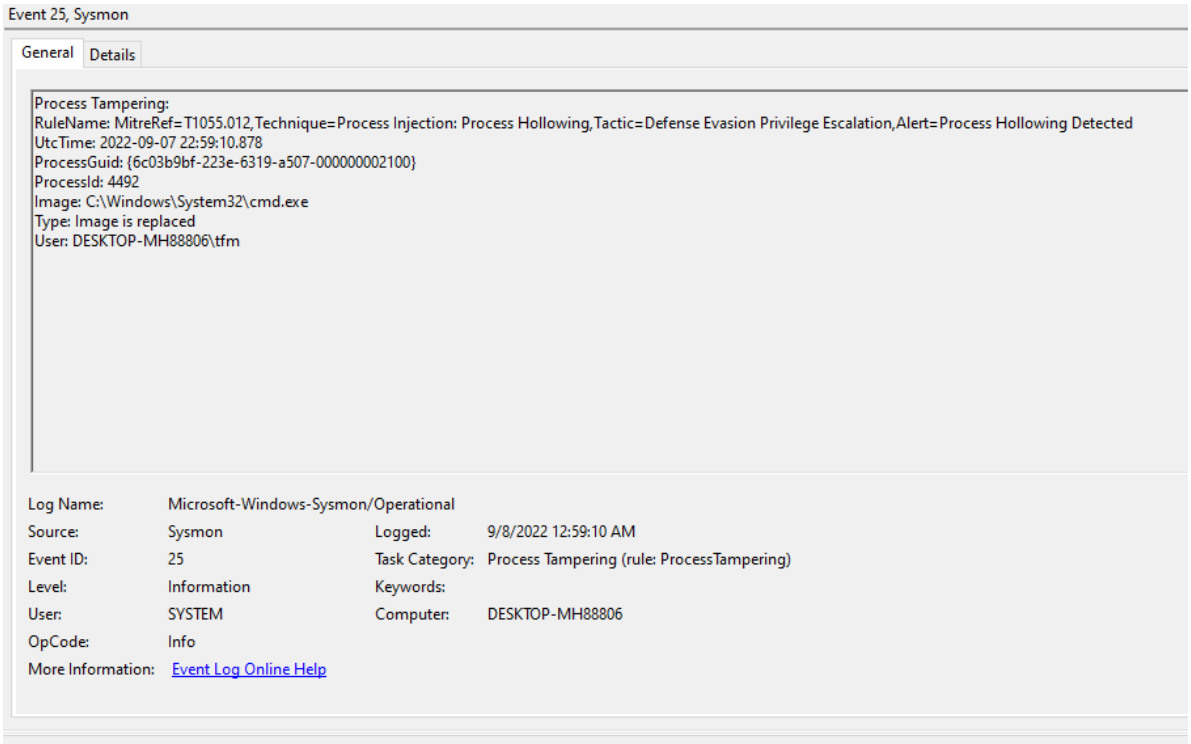

Ilustración 82. Process Tampering – Hollowing (parte 5)

6.2.17. ELIMINACIÓN DE FICHEROS - SIN ALMACENAMIENTO

Para esta prueba, se utilizó la ID 26 de *SysmonSimulator*, con el siguiente código:

```
void deleteFile26() {
    HANDLE hFile = CreateFile(
        L"NewFile.bat",
        GENERIC_WRITE,
        FILE_SHARE_READ,
        NULL,
        CREATE_ALWAYS,
        FILE_ATTRIBUTE_NORMAL,
        NULL);

    if (hFile == INVALID_HANDLE_VALUE) {
        printf("[!] Error code is: %lu\n", GetLastError());
    }
    else {
        CloseHandle(hFile);
    }

    if (!DeleteFile(L"NewFile.bat")) {
        printf("[-] Error deleting file: %lu\n", GetLastError());
    }
}
```

Ilustración 83. Eliminación de ficheros - sin almacenamiento (parte 1)

La explicación es directa: se crea un fichero bajo el nombre de *NewFile.bat* que posteriormente, será eliminado del sistema de manera permanente de manera no autorizada. Ante este comportamiento, la solución EDR eleva una regla de *firewall*, mata el proceso y genera las alertas pertinentes. Para el equipo de SOC, dispondrá entre muchos de sus datos, de tres *hashes*, que podrá utilizar en sus tareas de contrarrespuesta de manera adecuada.

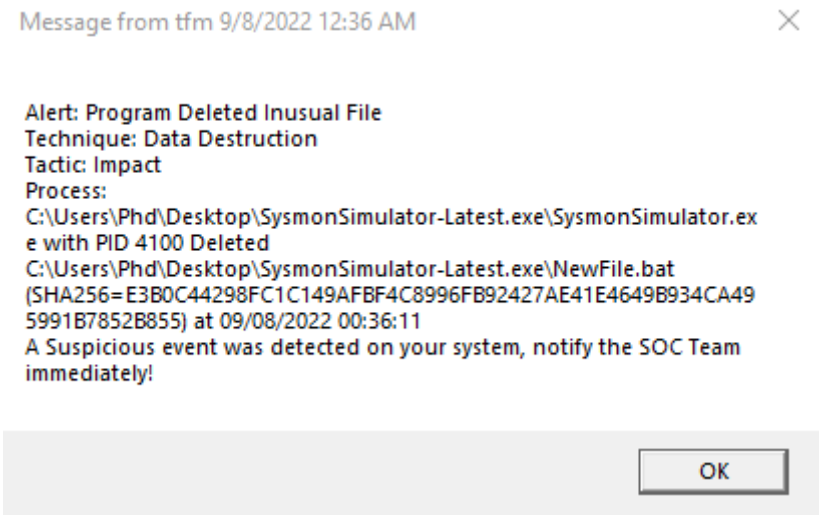

Ilustración 84. Eliminación de ficheros - sin almacenamiento (parte 2)

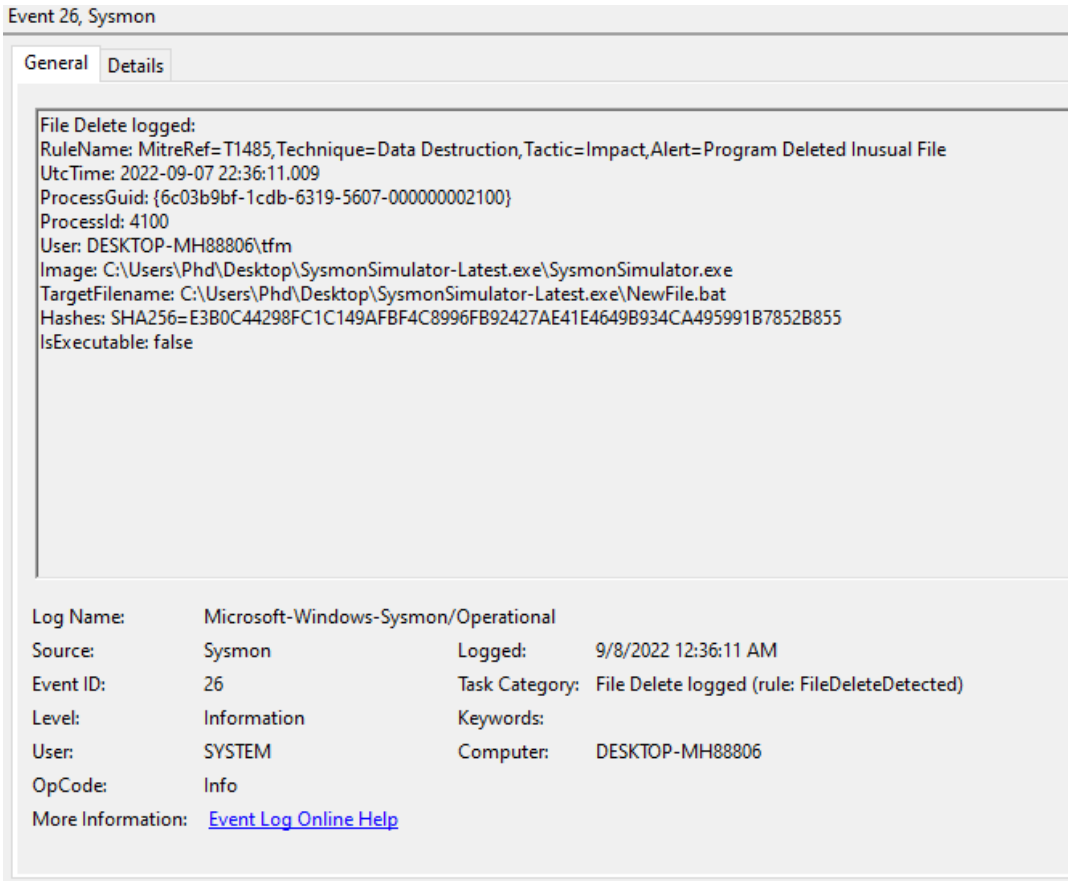

Ilustración 85. Eliminación de ficheros - sin almacenamiento (parte 3)

7. RESULTADOS

Tras la realización de las pruebas anteriores, los resultados obtenidos son satisfactorios. Se ha cumplido todos los objetivos específicos marcados, lo que, en resumidas cuentas, satisface el objetivo global del proyecto; la realización de una solución EDR funcional, gratuita, *open-source* y flexible**; lo que la hace una solución EDR al alcance de las PYME**.

Se ha podido comprobar que el tiempo de respuesta desde que se produce un evento en el equipo *endpoint* hasta que llega al equipo del SOC es de milisegundos. Además, toda la información llega *parseada* de manera adecuada para que pueda ser tratada con la máxima rigurosidad de la manera más rápida posible. Del mismo modo, el equipo de especialistas en ciberseguridad dispone de las herramientas suficientes para poder enriquecer aún más la información que les llega desde los eventos del sistema específico que está siendo vulnerado. Este enriquecimiento provee al equipo de medidas adicionales para el tratamiento del caso.

El agente utilizado ofrece una configuración perfecta; al transmitir únicamente aquellos datos de los eventos generados por *Sysmon*, y de nada más La indexación que se da es variante en función del tipo de evento gestionado, lo que favorece la clase de información que se notifica; avalorando más el peso de los datos de los eventos, a diferencia de si existiera un modelo genérico.

La configuración del fichero *Sysmon* ofrece de manera clara y concisa el funcionamiento de la arquitectura que es empleada a modo de solución EDR. Además, ofrece una plantilla a detalle que permite generar ficheros de configuración nuevos en base a éste que minorizan la posibilidad de errores por incompatibilidades de formato en relación con el *script* que ofrece respuesta a los eventos. Dicha plantilla presenta descripción del programa, banderas a utilizar, ejemplo de generación de regla y estructura pormenorizada para la facilidad de uso de nuevos usuarios.

La conexión entre los componentes de la arquitectura ofrece seguridad plena entre ellos, con la posibilidad de uso de protocolos como SSL/TLS o HTTPS, imposición de lista blanca y control mediante reglas establecidas. Estas medidas reducen enormemente la expectativa de posibles vulneraciones de los componentes, recopilación de datos en la transmisión de información entre éstos y escalada de privilegios de los sistemas.

El fichero que ofrece medidas de respuesta ante eventos dados ofrece la opción de modificar cada uno de los comportamientos dados dentro de estos identificadores de eventos. Esto

significa que una organización puede desde añadir o quitar funcionalidades dentro de los condicionales dados en el fichero; hasta crear comportamientos aún más complejos mediante la anidación de formatos anidados. Además, debido a que se basa en el uso de *Powershell*, ofrece un gigantesco abanico de posibilidades a la hora de ofrecer comportamientos para estos eventos: ejecución de programas, ejecutables, comandos, módulos u otros *scripts*, realizar conexiones remotas mediante RDP o WMI, manejo de reglas *firewall* en base a comportamientos, distribución de *logs* a equipos segmentados mediante *SSHpass*…, las posibilidades son infinitas.

Por último, y aunque no formase parte del objetivo global o específico inicialmente; toda la implementación ha sido basada en un estándar tan potente como es la ISO 27034-3:2018, y así se ve reflejado a efectos prácticos en las pruebas realizadas a modo de escenario reducido. Esta metodología valoriza enormemente el atractivo de este proyecto desde un punto de vista de negocio y científico; al ser un referente en el desarrollo de aplicaciones seguras actualmente, y un marco normativo de peso en el ámbito tecnológico y más concretamente, de la ciberseguridad.

8. CONCLUSIONES

Puede concluirse en que se ha evidenciado de manera teórico-práctica la creación, redacción, despliegue y uso esperado de una arquitectura que funciona perfectamente a modo de solución EDR, y que está al alcance de los sistemas y equipos de aquellas organizaciones o individuos que no podían permitirse soluciones de este tipo de pago ya existentes. Todo este proceso ha sido realizado mediante la unión, configuración y cohesión de un *script* realizado manualmente y de un alto número de tecnologías gratuitas que siempre estuvieron al alcance, pero que nunca se objetivaron con lo que se ha propuesto en este trabajo; a través de los métodos, procedimientos y técnicas especificados.

Del mismo modo, además de las propias funcionalidades que transcriben a un EDR, esta arquitectura también ofrece la alternativa de modificar de manera muy flexible cualquiera de los componentes que involucran a la monitorización, prevención, detección, análisis y respuesta frente a las amenazas dadas. Al mismo tiempo, y mediante un estudio analítico indicado sobre la estructura de la infraestructura, es posible añadir otro tipo de tecnologías que complementen la arquitectura base que se está empleando, potenciando las características de ciberdefensa ofrecidas. Esto lleva a concluir, que se tiene entre manos una herramienta capaz de ser integrada bajo unas pautas que pueden ser previamente preestablecidas por una organización.

En cuanto al ciclo de vida de esta solución, es concluyente que tenga un ciclo de vida extremadamente largo; debido a que el pilar principal sobre el que se soporta recae en el uso de la herramienta *Sysmon*, una herramienta propia de *Microsoft* que se encuentra continuamente actualizada mediante el arreglo de *bugs* que aparecen en el programa, u ofreciendo nuevas reglas para diferentes IDs de eventos del sistema, lo que transitivamente aporta aún más capacidad de respuesta para la solución EDR establecida.

Por último, puede afirmarse con certeza que la aplicación es lo suficientemente segura, al haber sido desarrollada bajo un estándar de normalización ampliamente conocido y establecido en muchas aplicaciones como es la ISO 27034-3:2018, lo que implica que su utilización bajo la supervisión de especialistas en el ámbito de la ciberseguridad promete una mejora en la protección de sus equipos, dominios y sistemas frente a otras soluciones más primitivas como pueden ser antivirus tradicionales.

Como conclusión global, se puede afirmar que la solución EDR propuesta en forma de arquitectura es funcional, segura, flexible, gratuita y longeva; según se expone en todo lo manifestado a lo largo de las hojas que constituyen el grueso de este documento.

9. TRABAJO FUTURO

Tras la realización del presente trabajo de investigación, se incluyen algunas mejoras o trabajos de innovación que tomen este documento como referencia:

- Generación de alertas adicionales desde *Elastalert* a un grupo de *Telegram* con los detalles de dicha alerta, de manera que, si el grupo de SOC no tiene su equipo principal disponible en un periodo en el que se eleve una alerta referente a una amenaza grave; puedan visualizarlo en sus dispositivos móviles y tener conocimiento lo antes posible de ésta.
- Implementación de la arquitectura propuesta en sistemas operativos *Linux* mediante el uso de la herramienta de *Sysinternals* conocida como *Sysmon For Linux*.
- Incorporación de una tecnología *sandbox,* por ejemplo Cuckoo, al diagrama de la arquitectura genérico (Cuckoo, 2019). Esta funcionalidad provee comportamientos de salida post ejecución de un fichero y/o ejecutable en un entorno seguro, los cuales pueden ser tratados por *Sysmon*.
- Adaptación de la arquitectura a modelos que utilizan inteligencia artificial, de manera que se adquiera capacidad de aprendizaje automático frente a amenazas que no han sido configuradas para ser detectadas en ninguno de los ficheros de configuración de todas las dependencias que constituyen el proyecto.
- Automatizar el proceso de compartición de *malware* con *MISP*, de manera que, una vez analizada una amenaza y producido un documento o caso por parte del equipo de SOC con la información específica que se ha querido incluir, el proceso se realice de manera automática y no manual. Esta implementación puede hacerse a través de un *script* del sistema, o bien, mediante la creación de un nuevo módulo que pueda ser utilizado en *TheHive*.
- Automatizar el proceso de generación de información adicional mediante los observable que llegan a *TheHive con Cortex*, una vez un caso nuevo de una amenaza es generado. De esta manera, los integrantes del equipo de SOC pueden ver toda la información profundizada sin necesidad de emplear las utilidades de manera manual. Esta mejora puede realizarse mediante la configuración de las herramientas que proporciona la propia plataforma, llegando incluso a poder establecer plantillas específicas para el análisis con *Cortex* en función del tipo de amenaza que se localiza en un caso en concreto.

- Integrar o transformar esta arquitectura en una solución de *eXtended Detection Response* (XDR), con capacidad de analizar los comportamientos derivados en la nube, redes; entre otros.
- Completar la arquitectura con herramientas como *Sysmon Tools* (compuesta por *Sysmon View, Sysmon Shell* y *Sysmon Box*), que aporten una visualización más amigable de los datos y ficheros de configuración referentes a esta dependencia (Shalabi, 2021). Se muestra un ejemplo de estas herramientas:

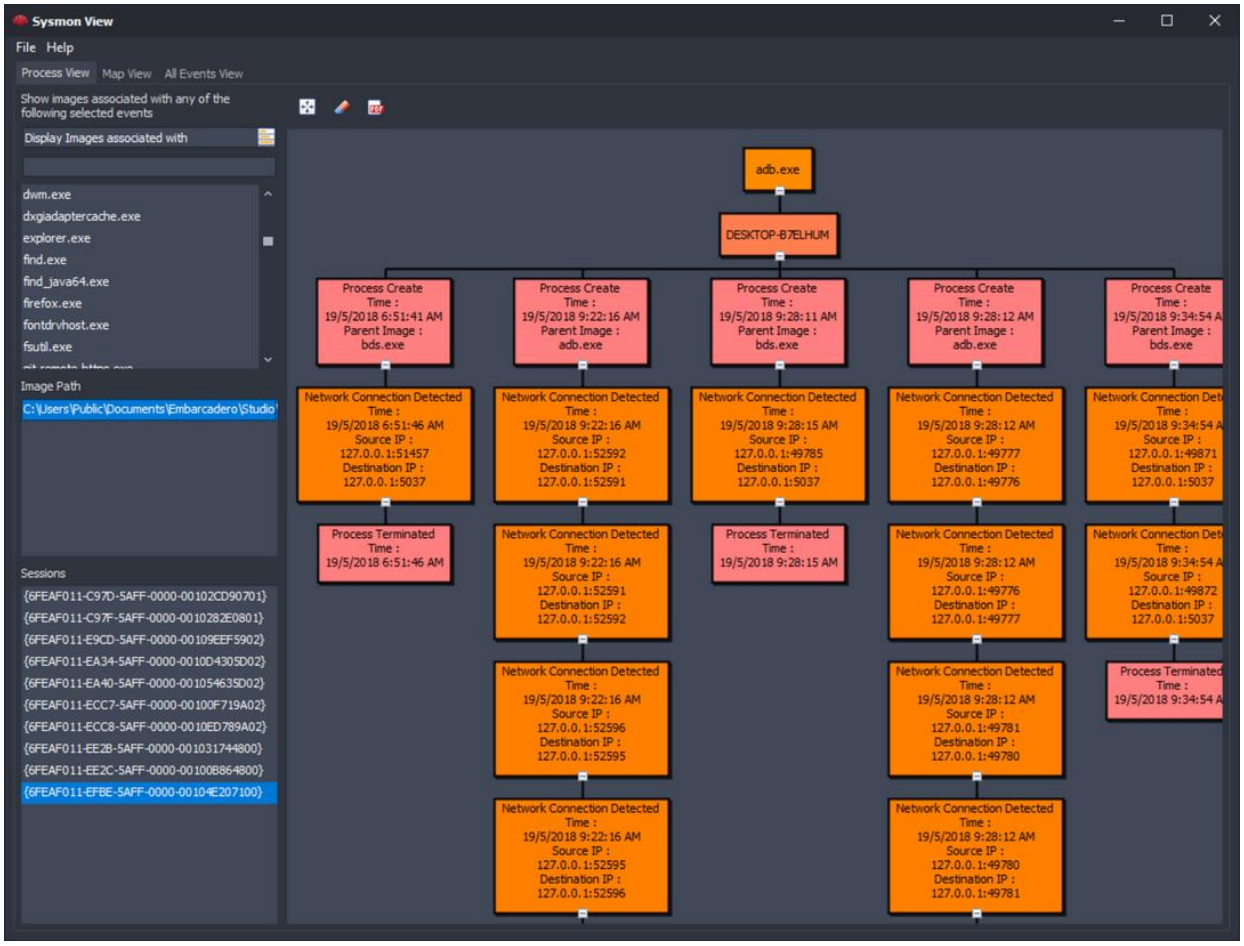

Ilustración 86. Ejemplo de Sysmon View (parte 1)

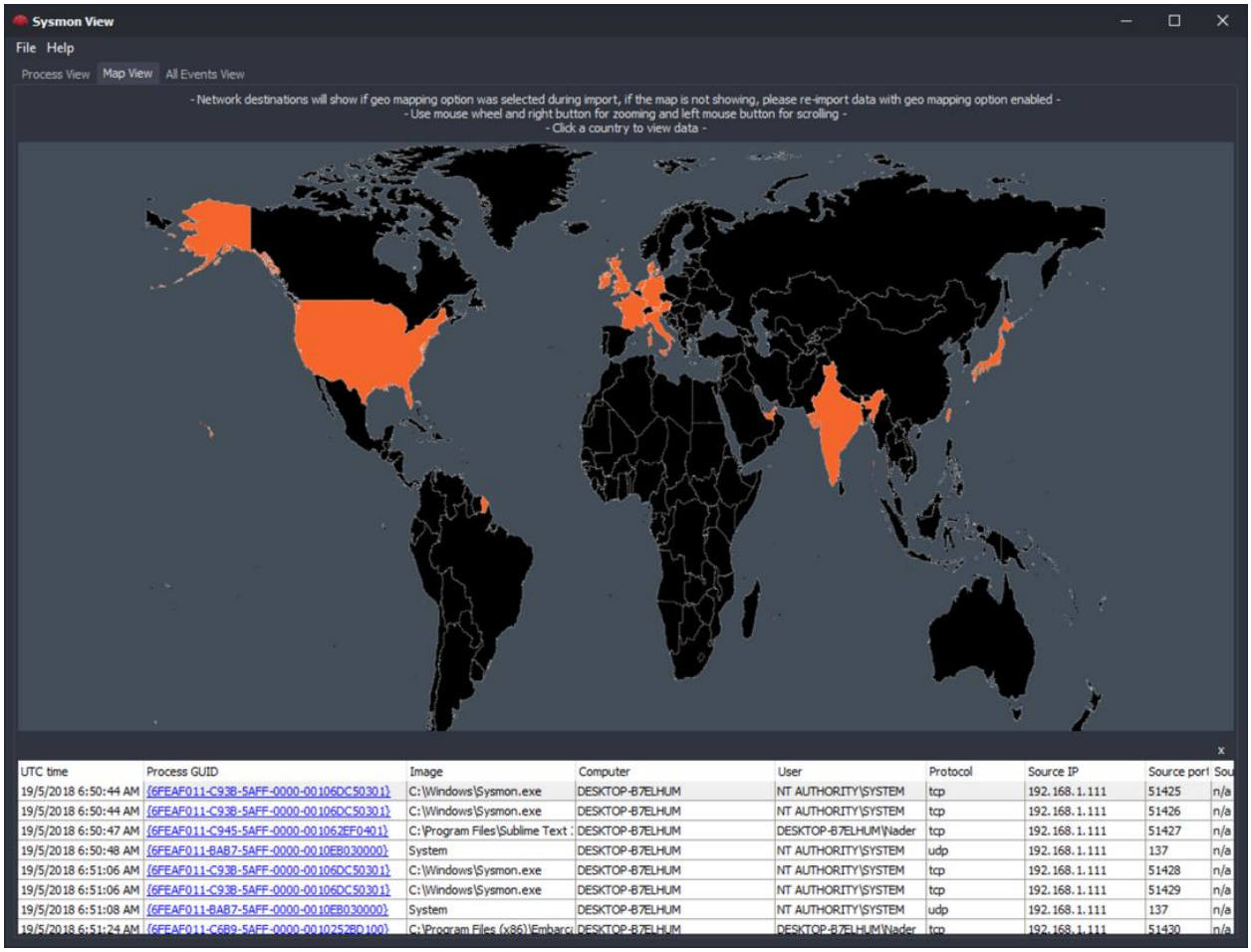

Ilustración 87. Ejemplo de Sysmon View (parte 2)

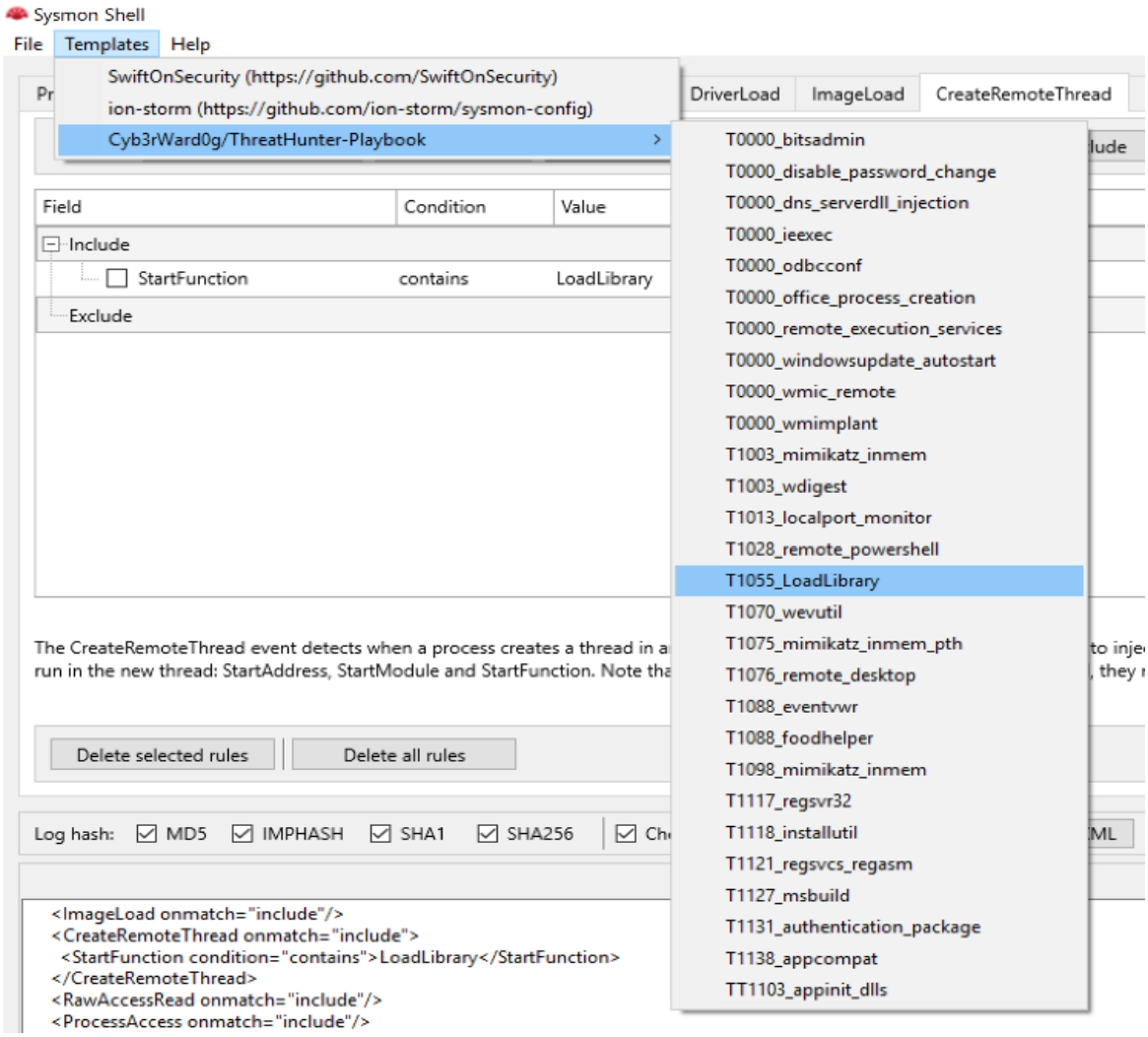

Ilustración 88. Ejemplo de Sysmon Shell

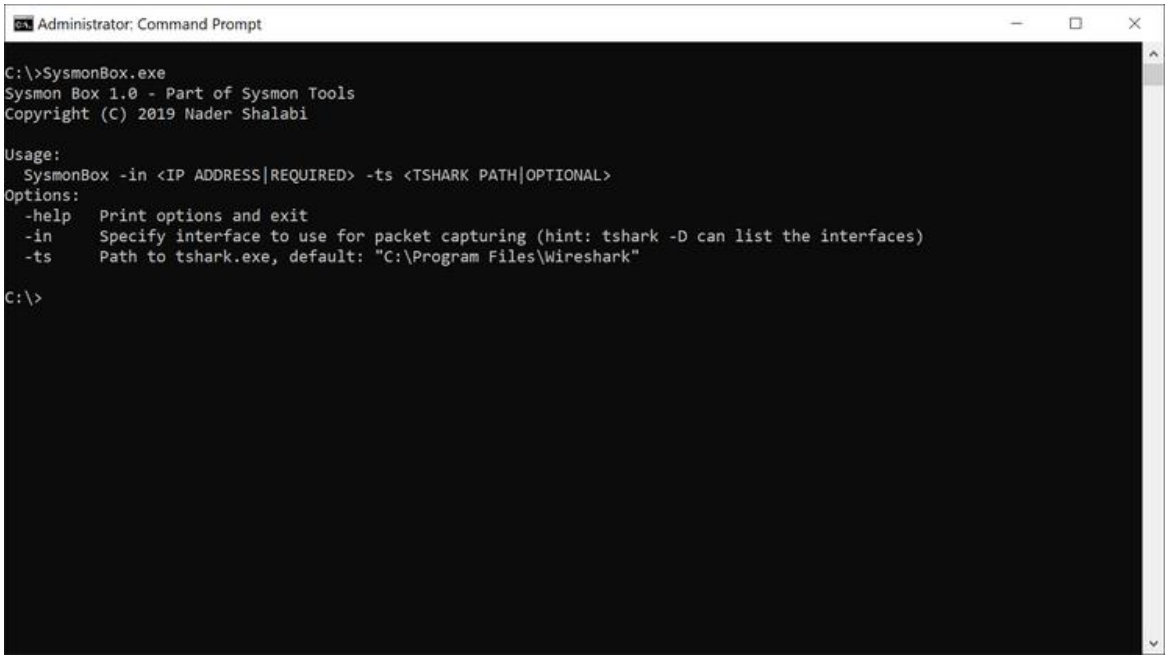

Ilustración 89. Ejemplo de Sysmon Box

APÉNDICES

BIBLIOGRAFÍA

Academy, B. T. (25 de Mayo de 2022). *Malware Analysis: How to use Yara rules to detect malware*. Recuperado el 29 de Julio de 2022, de https://www.blueteamsacademy.com/yara/

Adouani, N., Franco, T., Kadhi, S., Leonard, J., Co, D., & Kuhnert, N. (15 de Marzo de 2022). *TheHive.* Recuperado el 11 de Julio de 2022, de https://thehive-project.org/

Analyzing Document Macros with Yara. (27 de Marzo de 2019). Recuperado el 2022 de Julio de 10, de https://0xdf.gitlab.io/2019/03/27/analyzing-document-macros-with-yara.html

Arntz, P. (15 de Septiembre de 2017). *Explained: YARA rules.* Recuperado el 21 de Agosto de 2022, de https://www.malwarebytes.com/blog/news/2017/09/explained-yara-rules

AVINetworks. (1 de Septiembre de 2022). *Elliptic Curve Cryptography Definition*. Obtenido de https://avinetworks.com/glossary/elliptic-curve-cryptography/

Britannica. (1 de Septiembre de 2022). *HTTP*. Obtenido de HyperText Transfer Protocol: https://www.britannica.com/technology/HTTP

Burdova, C. (15 de Mayo de 2022). Recuperado el 9 de Junio de 2022, de https://www.avast.com/c-eternalblue

Burnham, Z. (18 de Noviembre de 2018). *Sending Logs to ELK with Winlogbeat and Sysmon*. Recuperado el 19 de Agosto de 2022, de https://burnhamforensics.com/2018/11/18/sending-logs-to-elk-with-winlogbeat-and-sysmon/

Catch All (MS Windows Event Logging XML - Sysmon). (1 de Febrero de 2022). Recuperado el 1 de Agosto de 2022, de https://docs.logrhythm.com/docs/devices/ms-windows-event-log-sources/ms-windows-event-logging-xml-sysmon-configuration-guide

CCN. (1 de Noviembre de 2013). *GUÍA/NORMA DE SEGURIDAD DE LAS TIC (CCN-STIC-470F/1): MANUAL DE USUARIO PILAR ANÁLISIS Y GESTIÓN DE RIESGOS.* Obtenido de https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/163-ccn-stic-470f1-manual-de-la-herramienta-de-analisis-de-riesgos-pilar-5-3/file.html

CCN. (1 de Abril de 2020). *Esquema Nacional de Seguridad. Gestión de ciberincidentes.* Obtenido de https://www.ccn-cert.cni.es/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/988-ccn-stic-817-gestion-de-ciberincidentes/file.html

CEC. (20 de Marzo de 2020). *ISO/IEC 27034: Estándar Internacional para la seguridad de las aplicaciones.* Recuperado el 30 de Agosto de 2022, de https://noticias.cec.es/index.php/2020/03/20/isoiec-27034-estandar-internacional-para-la-seguridad-de-las-aplicaciones/

Cloutier, R. (2 de Noviembre de 2021). *EDR: Endpoint Detection and Response*. Recuperado el 7 de Junio de 2022, de https://securitystudio.com/edr/

Cuckoo. (31 de Diciembre de 2019). *What is Cuckoo?* Recuperado el 14 de Julio de 2022, de https://cuckoosandbox.org/

Culafi, A. (5 de Julio de 2022). *Ransomware in 2022: Evolving threats, slow progress*. Recuperado el 22 de Julio de 2022, de https://www.techtarget.com/searchsecurity/news/252522369/Ransomware-Evolving-threats-slow-progress

Cyfirma. (12 de Mayo de 2022). *Onyx Ransomware Report*. Recuperado el 13 de Agosto de 2022, de https://www.cyfirma.com/outofband/onyx-ransomware-report/

Digicert. (1 de Septiembre de 2022). *The Ultimate Guide: What is SSL, TLS and HTTPS?* Recuperado el 14 de Julio de 2022, de https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https

Drysdale, J. (8 de Enero de 2021). *A Sysmon Event ID Breakdown*. Recuperado el 30 de Junio de 2022, de https://www.blackhillsinfosec.com/a-sysmon-event-id-breakdown/

Drysdale, J. (8 de Junio de 2021). *A Sysmon Event ID Breakdown – Now with Event ID 25!!* Obtenido de https://www.blackhillsinfosec.com/a-sysmon-event-id-breakdown/

Elastic. (7 de Julio de 2022). *What is Elasticsearch?* Recuperado el 20 de Agosto de 2022, de https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html

Encryption Consulting. (17 de Julio de 2022). *What is RSA? How does an RSA work?* Obtenido de https://www.encryptionconsulting.com/education-center/what-is-rsa/

GoDaddy. (28 de Julio de 2017). *YARA Rules for ProcFilter*. Recuperado el 4 de Agosto de 2022, de https://github.com/godaddy/yara-rules

Grossman, N. (29 de Septiembre de 2017). *EternalBlue – Everything There Is To Know*. Recuperado el 9 de Junio de 2022, de https://research.checkpoint.com/2017/eternalblue-everything-know/

Hartong, O. (12 de Junio de 2019). *Sysmon 10.0 - New features and changes*. Recuperado el 11 de Julio de 2022, de https://medium.com/@olafhartong/sysmon-10-0-new-features-and-changes-e82106f2e00

Hartong, O. (28 de Abril de 2020). *Sysmon 11 — DNS improvements and FileDelete events*. Recuperado el 30 de Junio de 2022, de https://medium.com/falconforce/sysmon-11-dns-improvements-and-filedelete-events-7a74f17ca842

ISO. (1 de Mayo de 2018). *International Standard: ISO/IEC 27034-3. Information Technology - Application security - Part 3: Application security management process.* Recuperado el 2 de Septiembre de 2022, de https://www.iso.org/standard/55583.html

ISO. (1 de Septiembre de 2022). *ISO - International Organization for Standardization*. Obtenido de https://www.iso.org

Kaspersky. (1 de Septiembre de 2022). *What is an IP Address – Definition and Explanation.* Obtenido de https://www.kaspersky.com/resource-center/definitions/what-is-an-ip-address

Liang, H., Shilpa, B., Mandalika, S., Czechowski, A., Matts, D., Coulter, D., . . . Vangel, D. (13 de Junio de 2022). *Generate a kernel or complete crash dump*. Recuperado el 24 de Agosto de 2022, de https://docs.microsoft.com/en-us/windows/client-management/generate-kernel-or-complete-crash-dump

Maayan, G. D. (7 de Abril de 2020). *A Brief History of EDR Security*. Recuperado el 8 de Junio de 2022, de https://dzone.com/articles/a-brief-history-of-edr-security

Malware Archaeology. (31 de Agosto de 2019). *WINDOWS SYSMON LOGGING CHEAT SHEET.* Obtenido de https://static1.squarespace.com/static/552092d5e4b0661088167e5c/t/5d5588b51fd81f0001471db4/1565886646582/Windows+Sysmon+Logging+Cheat+Sheet_Aug_2019.pdf

N-able. (30 de Junio de 2020). *A Short History of EDR*. Recuperado el 8 de Junio de 2022, de https://www.n-able.com/blog/short-history-of-edr

NIST. (1 de Septiembre de 2020). *NIST Special Publication 800-53: Revision 5 - Security and Privacy Controls for.* Recuperado el 2 de Septiembre de 2022, de https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf

NIST. (11 de Enero de 2022). Obtenido de https://www.nist.gov/about-nist

Official Elastic Community. (21 de Junio de 2021). *Ingest Pipelines - Daily Elastic Byte S02E11.* Recuperado el 26 de Agosto de 2022, de https://www.youtube.com/watch?v=-wOvyubrHIM

OpenSecure. (20 de Agosto de 2021). *ElastAlert Install - Automatically Forward Wazuh Alerts to TheHIVE!* Recuperado el 31 de Agosto de 2022, de https://www.youtube.com/watch?v=7zBGQxqf2G4

Perez, C. (9 de Noviembre de 2021). *SysmonCommunityGuide - Configuration*. Recuperado el 10 de Julio de 2022, de https://github.com/trustedsec/SysmonCommunityGuide/blob/master/chapters/configuration.md

Perez, C. (22 de Octubre de 2021). *SysmonCommunityGuide: File Delete*. Recuperado el 30 de Junio de 2022, de https://github.com/trustedsec/SysmonCommunityGuide/blob/master/chapters/file-delete.md

Praxis. (31 de Diciembre de 2019). *Evaluación de la probabilidad-impacto.* Recuperado el 25 de Agosto de 2022, de https://www.praxisframework.org/es/library/probability-impact-assessment

Praxis. (31 de Diciembre de 2019). *Reportes RAG.* Recuperado el 25 de Agosto de 2022, de https://www.praxisframework.org/es/library/rag-reports

Project, M. (3 de Septiembre de 2022). *MISP - Threat Intelligence Sharing Platform.* Recuperado el 11 de Julio de 2022, de https://github.com/MISP/MISP

Recovery, D. (6 de Junio de 2022). *Ransomware Mindware*. Recuperado el 10 de Julio de 2022, de https://digitalrecovery.com/es/recuperar-datos-ransomware-mindware/

Red Canary. (1 de Septiembre de 2022). *Windows Atomic Tests by ATT&CK Tactic & Technique defense-evasion.* Recuperado el 2 de Septiembre de 2022, de https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/Indexes/Indexes-Markdown/windows-index.md

RootDSE. (7 de Enero de 2022). *Understanding Sysmon Events using SysmonSimulator*. Recuperado el 28 de Agosto de 2022, de https://rootdse.org/posts/understanding-sysmon-events/

Rules, Y. (12 de Abril de 2022). *YARA Rules*. Recuperado el 28 de Agosto de 2022, de https://github.com/Yara-Rules/rules

Russinovich, M., & Garnier, T. (16 de Agosto de 2022). *Sysmon v14.0*. Recuperado el 18 de Agosto de 2022, de https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

Shalabi, N. (11 de Agosto de 2021). *Sysmon Tools*. Obtenido de https://github.com/nshalabi/SysmonTools

Simplilearn. (23 de Agosto de 2022). *Digital Signature Algorithm (DSA) in Cryptography: How It Works and Advantages*. Obtenido de https://www.simplilearn.com/tutorials/cryptography-tutorial/digital-signature-algorithm

Stackoverflow. (20 de Marzo de 2013). *Take a user dump using powershell*. Recuperado el 24 de Agosto de 2022, de https://stackoverflow.com/questions/15523460/how-can-i-take-a-user-dump-using-powershell

Stackoverflow. (11 de Noviembre de 2019). *How to convert powershell UTC datetime object to EST*. Recuperado el 2 de Julio de 2022, de https://stackoverflow.com/questions/58802973/how-to-convert-powershell-utc-datetime-object-to-est

StorMagic. (27 de Abril de 2021). *What is Data Compliance?* Obtenido de https://stormagic.com/resources/beginners-guides/data-compliance-a-beginners-guide/

Sysmon EDR Active Response Features. (1 de Mayo de 2021). Recuperado el 2 de Julio de 2022, de https://github.com/ion-storm/sysmon-edr

TheHive-Project. (22 de Junio de 2022). *Cortex.* Recuperado el 11 de Julio de 2022, de https://github.com/TheHive-Project/Cortex

Trellix. (29 de Junio de 2022). *What Is the MITRE ATT&CK Framework?* Recuperado el 15 de Julio de 2022, de https://www.trellix.com/en-us/security-awareness/cybersecurity/what-is-mitre-attack-framework.html

UNE. (1 de Mayo de 2020). *UNE-EN ISO/IEC 15408-3:2020 (Ratificada).* Recuperado el 3 de Agosto de 2022, de https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0063576

Vojinovic, I. (8 de Julio de 2022). *Ransomware Statistics in 2022: From Random Barrages to Targeted Hits*. Recuperado el 27 de Julio de 2022, de https://dataprot.net/statistics/ransomware-statistics/

Wang, F. (5 de Agosto de 2022). *Breaking the Mold: Halting a Hacker's Code ep. 4 – Black Basta*. Recuperado el 6 de Agosto de 2022, de https://www.hillstonenet.com/blog/breaking-the-mold-halting-a-hackers-code-ep-4-black-basta/

Wheeler, S., Vasin, S., & Wilson, C. (5 de Mayo de 2022). *How to use the PowerShell documentation*. Recuperado el 30 de Junio de 2022, de https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/get-date?view=powershell-7.2

whids. (5 de Septiembre de 2022). Recuperado el 2 de Julio de 2022, de https://github.com/0xrawsec/whids

Wikipedia. (27 de Marzo de 2022). *Endpoint detection and response*. Recuperado el 7 de Junio de 2022, de https://en.wikipedia.org/wiki/Endpoint_detection_and_response

Windows Management Instrumentation. (10 de Septiembre de 2021). Obtenido de https://docs.microsoft.com/en-us/windows/win32/wmisdk/wmi-start-page

Yelp. (15 de Abril de 2020). *ElastAlert - Easy & Flexible Alerting With Elasticsearch.* Recuperado el 29 de Agosto de 2022, de https://elastalert.readthedocs.io/en/latest/elastalert.html

A. DIARIO DE INVESTIGACIÓN

-1 de mayo de 2022: Se elige la temática e ideas del proyecto entre tutor y ponente. Se discuten algunas tecnologías a utilizar.

-4-7 de mayo de 2022: Investigación y viabilidad del proyecto en función de lo que existe y las tecnologías discutidas.

-8 de mayo de 2022: Comunicación al director del máster de la elección del tema del TFM.

-9-14 de mayo de 2022: Lectura sobre conceptos básicos que rodean al EDR. Apunte de primeros hipervínculos interesantes.

-15 de mayo de 2022: Comunicación por parte del director del máster de que la temática elegida es buena.

-16-27 de mayo de 2022: Realización de la propuesta del TFM en base a lo marcado en los documentos del campus.

-28 de mayo de 2022: Entrega del documento del TFM.

-29 de mayo de 2022: Investigación de soluciones EDR existentes y su funcionamiento: *Symantec, Crowdstrike, etc.*

-30 de mayo de 2022: Investigación de soluciones EDR en repositorios abiertos: *sysmon-edr, whids, etc.*

-**31 de mayo-2 de junio de 2022:** Estudio de contraste entre solución EDR respecto a AVs tradicionales y de próxima generación

-3-5 de junio de 2022: Lectura sobre antecedentes para llevar a cabo el desarrollo de soluciones EDR, y por qué es de las soluciones más usadas en las organizaciones actualmente. Investigación de tecnologías para implementar la solución EDR. Se llega a *Sysmon*, *ELK* y *TheHive.*

-6-9 de junio de 2022: Recopilación de datos sobre los ataques que más vulneran las soluciones AV y que sin embargo, son detenidas por los EDR. Se llega a que son los *ransomware*.

-10-12 de junio de 2022: Recopilación e investigación de los *ransomwares* más conocidos en los últimos años. Análisis de su repercusión reputacional y económica en las organizaciones.

-13-19 de junio de 2022: Desarrollo de la introducción, estado de la cuestión y descripción del problema.

-20-23 de junio de 2022: Implementación de máquinas virtuales: una *Windows 10* y dos *Ubuntu*. Instalación de dependencias para *endpoint: Sysmon, sysmon-edr* y *winlogbeat*. Investigación básica sobre las tecnologías a emplear.

-24 de junio de 2022: Entendimiento del fichero de configuración *Sysmon*.

-28-30 de junio de 2022: Entendimiento del fichero del *script* en *Powershell*. Lectura sobre reglas *YARA*.

-1-4 de julio de 2022: Configuración del fichero de *winlogbeat*. Generación de fichero de configuración de *Sysmon*, y de dos reglas de prueba en relación con la creación de procesos (ID 1 en *Sysmon*) para ver cómo se recogen los eventos y se produce la respuesta deseada.

-5-6 de julio de 2022: Incorporación de *Elasticsearch* y *Kibana* a uno de los sistemas operativos *Ubuntu*. Conexión entre ambos.

-11-13 de julio de 2022: Búsqueda de metodologías que poder emplear para la incorporación segura de la aplicación.

-14 de julio de 2022: Se elige la ISO-27314-3:2018 como guía a seguir para la correcta implementación de la aplicación. Comienzo del desarrollo de ésta.

-16 de julio de 2022: Conexión entre *winlogbeat* y *Elasticsearch*. Comprobación de que los eventos llegan correctamente a *Elasticsearch* y pueden ser visualizados en *Kibana.*

-17-18 de julio de 2022: Lectura sobre las *Ingest-pipeline*, para desfragmentar la información incorporada en la etiqueta *message* de cada amenaza encontrada. Prueba básica.

-20 de julio de 2022: Investigación sobre maneras de elevar alertas a la futura plataforma de SOC.

-21-22 de julio de 2022: Instalación e incorporación de los elementos de *TheHive Project* en la máquina *Ubuntu* restante.

-23 de julio de 2022: Se determina *Elastalert* como la herramienta óptima para generar alertas.

-24 de julio de 2022: Incorporación de *Elastalert* al ELK. Primeras pruebas de intento de generación de alertas sin éxito.

-26 de julio de 2022: Envío de alertas *parseadas* desde *Elastalert* a *TheHive* conseguido.

-27 de julio de 2022: Comunicación el tutor del TFM del estado del proyecto y la decisión de presentar en la próxima convocatoria.

-28 de julio-17 de agosto de 2022: Retoma de la metodología y desarrollo.

-18-21 de agosto de 2022: Búsqueda de batería de pruebas para *Sysmon*. Se llega a *SysmonSimulator*. Se continúa el desarrollo de la metodología.

-22-27 de agosto de 2022: Finalización de implementación de la arquitectura de manera práctica y del desarrollo de la metodología.

-28-30 de agosto de 2022: Realización de pruebas mediante *SysmonSimulator*. Descubrimiento de las pruebas de *red-atomic team* y redacción de pruebas y evidencias que las acompañan.

-31 de agosto-2 de septiembre de 2022: Escritura resultados, conclusiones y trabajo futuro.

-3 de septiembre de 2022: Redacción en formato APA de las referencias bibliográficas empleadas.

-4-7 de septiembre de 2022: Escritura de los anexos y cierre del documento. Envío al tutor del TFM para corrección de errores, en el caso de que los haya.

B. MANUAL DE INSTALACIÓN

La instalación de la arquitectura comienza por desplegar en tres máquinas diferentes los siguientes sistemas operativos:

- *Windows 10*, para el equipo *endpoint* que va a ser protegido por la arquitectura.
- Solución *Linux*, preferiblemente *Ubuntu*, para la instalación de la parte del ELK reducido para la recogida e indexación de los datos de los equipos *endpoint*.
- Solución *Linux,* preferiblemente *Ubuntu*, para la instalación de la parte de *TheHive Project* para el equipo de SOC.

Se procede primeramente con la instalación de los componentes de la máquina *endpoint*. Se provee seguidamente de las dependencias que van a ser instaladas y configuradas:

- *Sysmon.*
- *Winlogbeat*.
- *Reglas YARA.*
- *Script Powershell*.

Con estas herramientas en el equipo, el entorno de instalación estaría listo. Junto al fichero de configuración de *Sysmon* del proyecto, la instalación se realiza introduciendo la siguiente orden desde la ruta donde se encuentre *Sysmon*:

```
sysmon -accepteula -i c:\<ruta-del-fichero-xml-sysmon-del-proyecto> -a DeletedFiles
```

Este comando realizará la instalación de *Sysmon* con el fichero de configuración de *P-EDR Arch.* Adicionalmente, establece la carpeta de cuarentena en '*C:/DeletedFiles'* de manera oculta para todos los usuarios, protegida por ACL y solamente accesible mediante consola por *NT-System.*

Ilustración 90. Guía de instalación: Fichero de configuración de Sysmon para P-EDR Arch

Seguidamente, es necesario establecer momentáneamente las políticas de ejecución de *scripts* en el sistema como no restringidas; ya que si no, no se podrá instalar la solución EDR. Para ello, abrimos una consola de *Powershell* con privilegios de administrador y se introduce:

```
Set-ExecutionPolicy Unrestricted
```

o en su defecto:

```
powershell -ep bypass
```

Deshabilitando la restricción para políticas de ejecución. A continuación, se lanza el *script* de *Powershell* bajo el nombre, *install_PEdrArch.ps1*. Este fichero realizará la instalación del EDR en el sistema; incluyendo sistema de monitorización respecto a *Sysmon*, respuesta a eventos por ID, creación de tarea y registros para iniciar la solución EDR junto al inicio del sistema, y las reglas *YARA* pertinentes.

En este momento, el EDR ya está funcionando respecto a las reglas de *Sysmon*, y será capaz de responder frente a los eventos dados; pero no monitorizará nada al equipo de SOC.

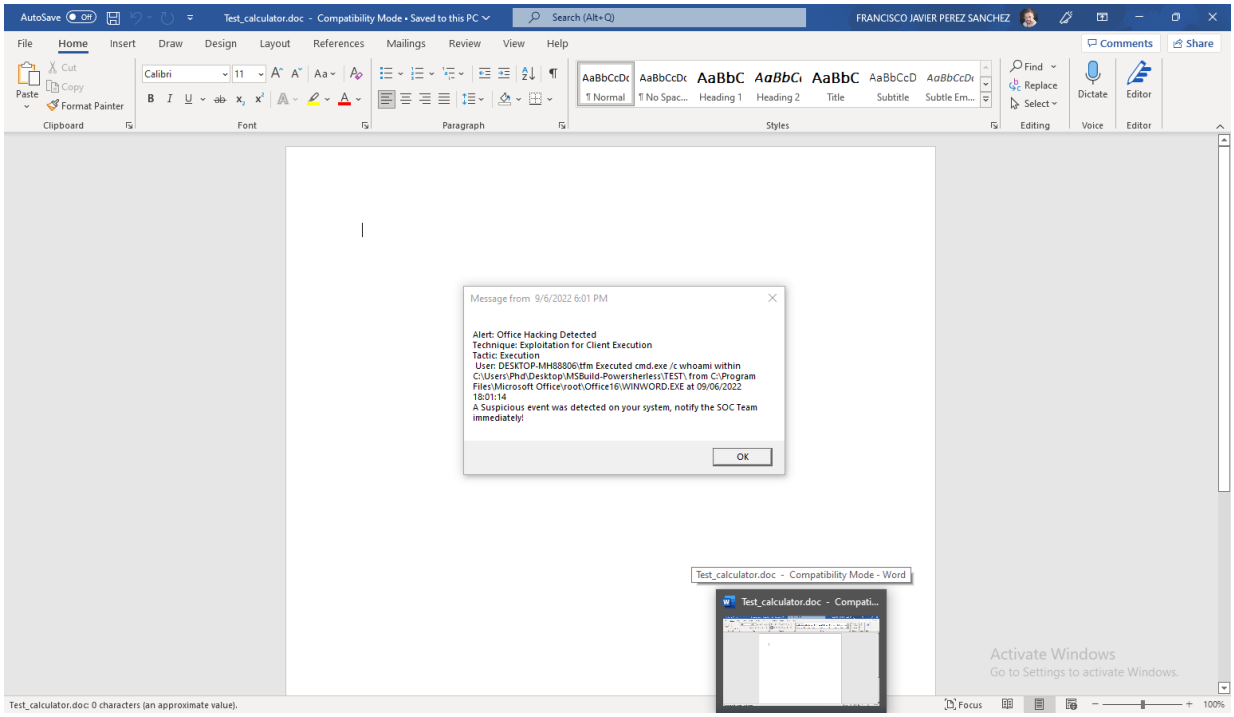

Ilustración 91. Guía de instalación: Funcionamiento de la solución EDR sin comunicación a SOC

Se instala *winlogbeat* en el sistema, accediendo al dominio de *Elastic* donde se ofrece dicha opción. Es importante señalar, que debe instalarse una versión de *winlogbeat* correspondiente a la versión de *Elasticsearch* que se use posteriormente, o inferior a éste.

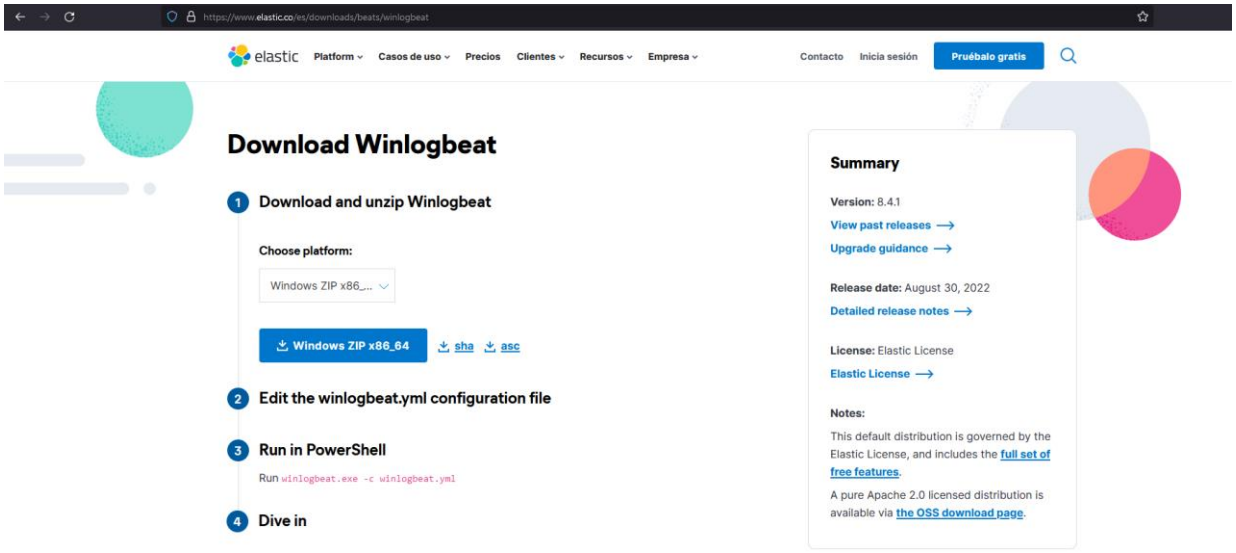

Ilustración 92. Guía de instalación: Instalación de winlogbeat (parte 1)

Una vez descargado *winlogbeat* para *Windows*, se procede a descomprimir en la ruta *'C/Archivos de Programa'*. Esto es debido a que es una ruta con privilegios donde el cortafuegos del sistema no pondrá inconvenientes a la hora de realizar la instalación de la dependencia.

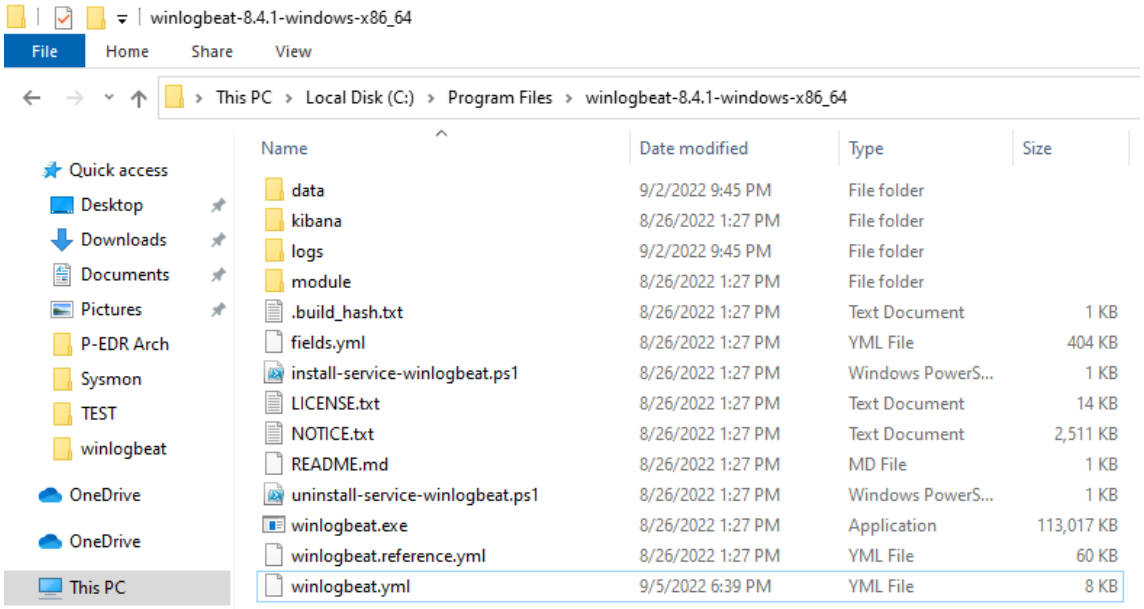

Ilustración 93. Guía de instalación: Instalación de winlogbeat (parte 2)

Antes de realizar el proceso de instalación, es obligatorio modificar el fichero de configuración para que solo incluya eventos de *Sysmon*, ya que si se deja por defecto, recogerá información proveniente de los *logs* del sistema, seguridad y *Powershell*, ofuscando la información importante que debe recibir el equipo de SOC.

El fichero de configuración de *winlogbeat* debería presentar el siguiente formato:

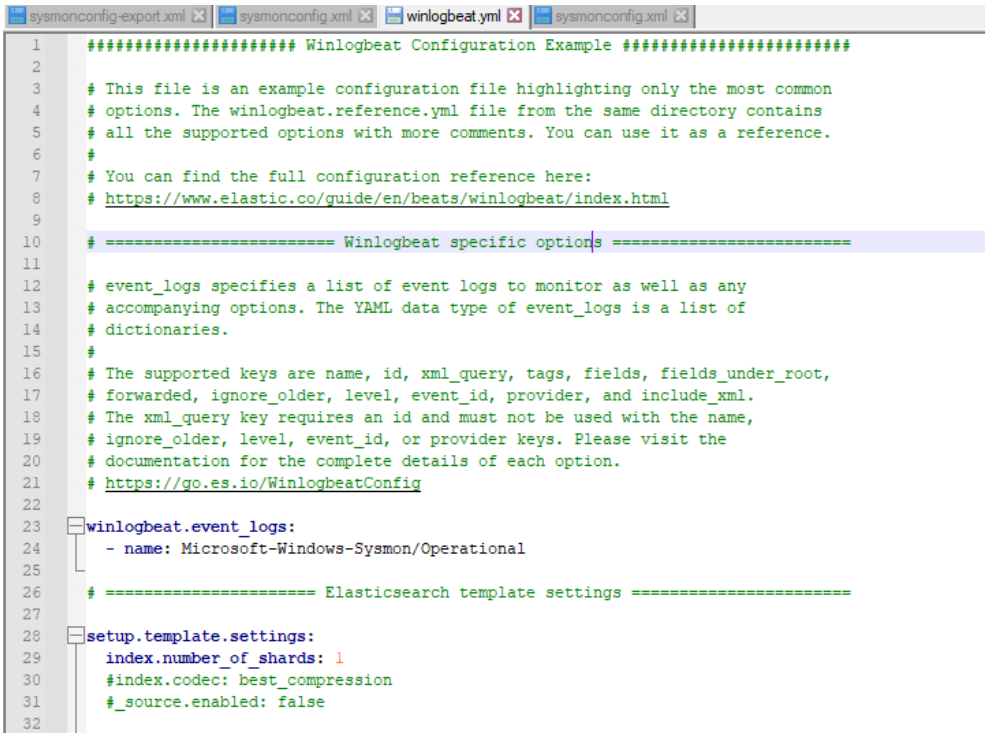

Ilustración 94. Guía de instalación: Instalación de winlogbeat (parte 3)

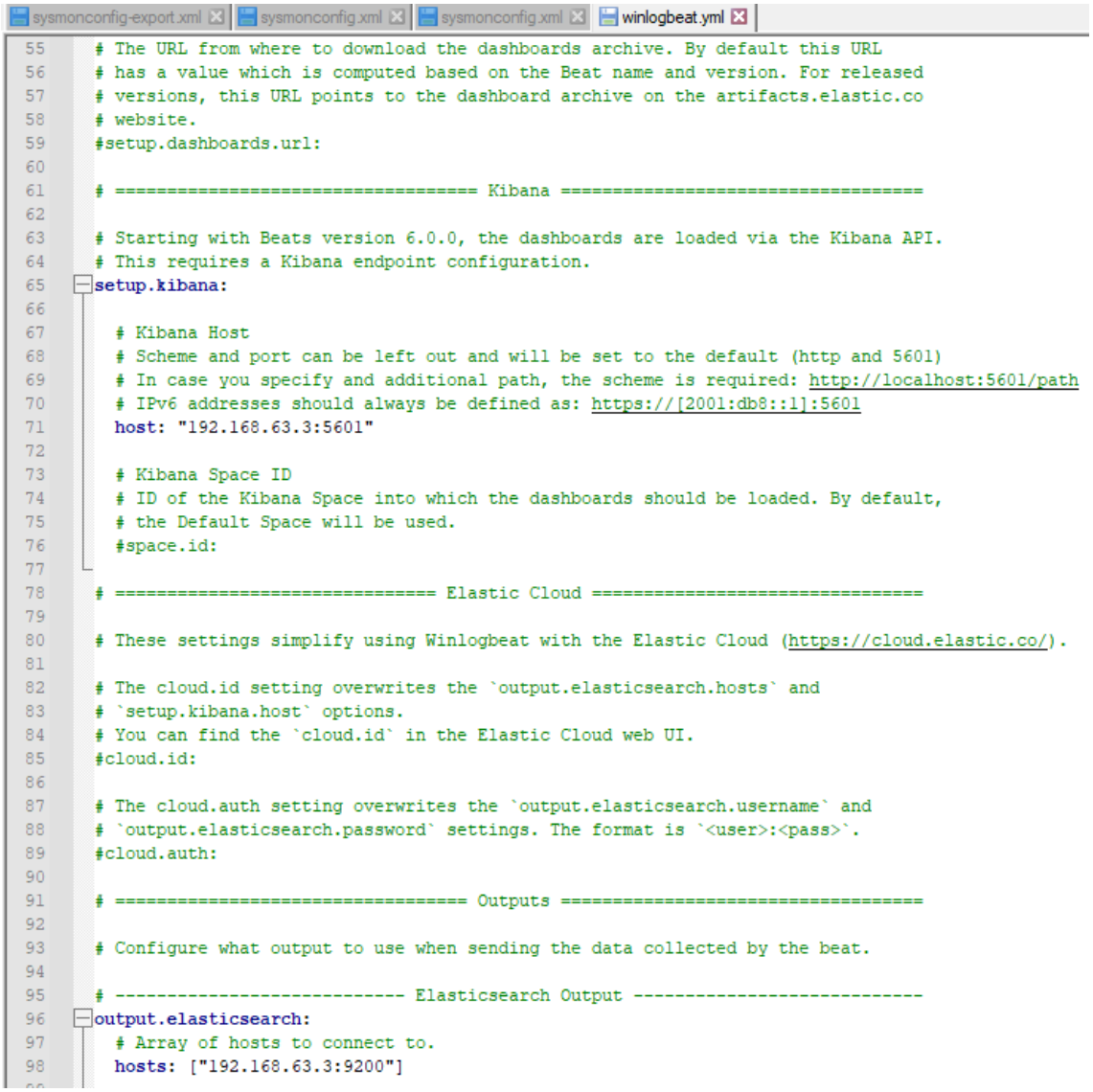

Ilustración 95. Guía de instalación: Instalación de winlogbeat (parte 4)

Finalmente, para la instalación de *winlogbeat,* comunicación con *Elasticsearch* y despliegue de su *indexer* en *Kibana;* es necesario la introducción de los comandos en una consola de *Powershell* con privilegios de administrador de la siguiente manera, en la ruta de la carpeta donde se localizan los ficheros de *winlogbeat*:

```
.\install-service-winlogbeat.ps1
```

```
.\winlogbeat.exe test config -c .\winlogbeat.yml -e
```

```
.\winlogbeat.exe setup -dashboards
```

Restaría iniciar el servicio creado en relación a *winlogbeat* en el sistema, ya que por defecto viene detenido.

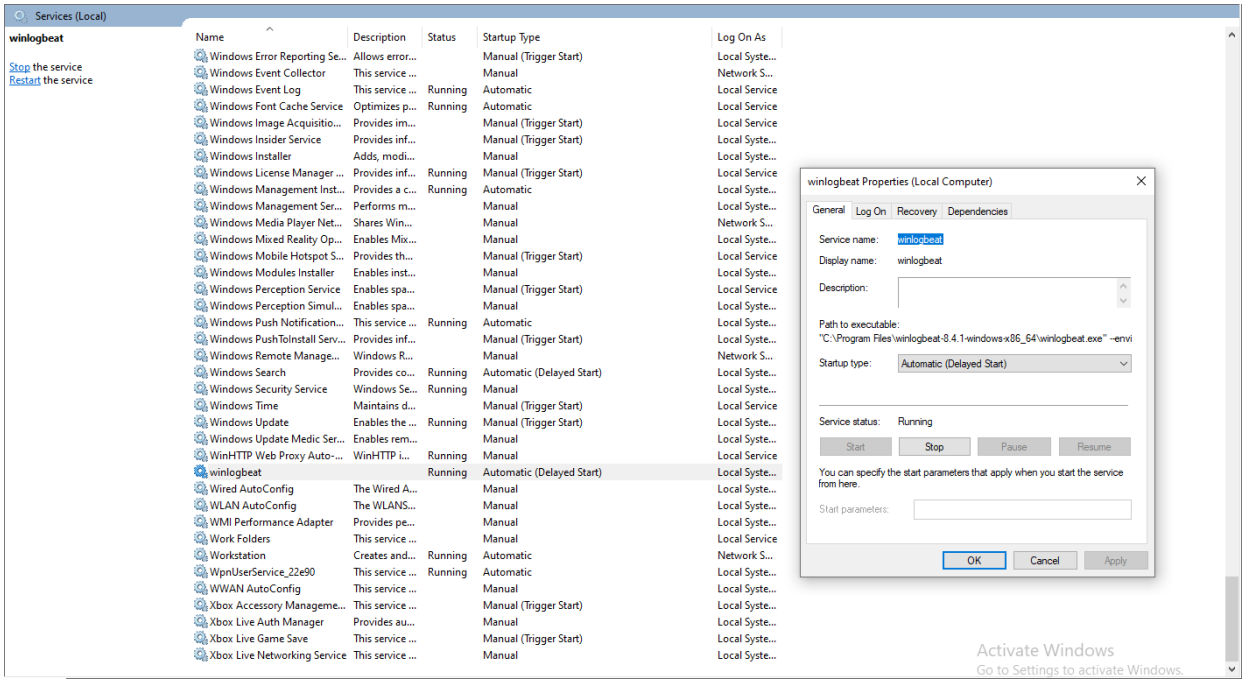

Ilustración 96. Guía de instalación. Instalación de winlogbeat (parte 5)

Con esto, queda configurada la parte de *endpoint*, y ya los datos se redirigen al sistema *Ubuntu* donde se encuentra el ELK reducido.

Respecto a la instalación de los componentes de este sistema, se necesita:

- *Curl*, para la descarga de las dependencias.
- *Elasticsearch.*
- *Kibana.*
- *Elastalert.*

Desde la máquina *Ubuntu* intermedia, se comienza por actualizar los paquetes del sistema; por seguir la práctica correcta y recomendada:

```
ubuntu-elk@ubuntu:/etc/netplan$ sudo apt-get update -y
Des:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88,7 kB]
Obj:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Des:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88,7 kB]
Des:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74,6 kB]
Descargados 252 kB en 1s (252 kB/s)
Leyendo lista de paquetes... Hecho
```

Ilustración 97. Guía de instalación: Instalación del ELK reducido (parte 1)

Con los paquetes actualizados, es necesario obtener la herramienta que permita descargar de manera sencilla todo lo necesario en el sistema. Para ello, se hace uso de *curl*:

```
ubuntu-elk@ubuntu:/etc/netplan$ sudo apt-get install curl -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libcurl4
Se instalarán los siguientes paquetes NUEVOS:
  curl libcurl4
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 379 kB de archivos.
Se utilizarán 1.053 kB de espacio de disco adicional después de esta operación.
Des:1 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcurl4 amd64 7.58.0-2ubuntu3.20 [220 kB]
Des:2 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 curl amd64 7.58.0-2ubuntu3.20 [159 kB]
Descargados 379 kB en 1s (305 kB/s)
Seleccionando el paquete libcurl4:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 166384 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libcurl4_7.58.0-2ubuntu3.20_amd64.deb ...
Desempaquetando libcurl4:amd64 (7.58.0-2ubuntu3.20) ...
Seleccionando el paquete curl previamente no seleccionado.
Preparando para desempaquetar .../curl_7.58.0-2ubuntu3.20_amd64.deb ...
Desempaquetando curl (7.58.0-2ubuntu3.20) ...
Configurando libcurl4:amd64 (7.58.0-2ubuntu3.20) ...
Configurando curl (7.58.0-2ubuntu3.20) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
Procesando disparadores para libc-bin (2.27-3ubuntu1.6) ...
```

Ilustración 98. Guía de instalación: Instalación del ELK reducido (parte 2)

A continuación, se descarga y se instala *Elasticsearch*. Para ello, se usará la versión comprimida para paquetes *Debian* y la dependencia que viene por defecto en los sistemas *Ubuntu* para descompresión e instalación de este tipo de paquetes conocida como *dpkg;* además de la herramienta recién descargada.

```
ubuntu-elk@ubuntu:~/Desktop/tools$ sudo curl -L -O https://artifacts.elastic.co/downloads/elastic
search/elasticsearch-7.15.2-amd64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  325M  100  325M    0     0  8742k      0  0:00:38  0:00:38 --:--:-- 9627k
ubuntu-elk@ubuntu:~/Desktop/tools$ sudo dpkg -i elasticsearch-7.15.2-amd64.deb
Seleccionando el paquete elasticsearch previamente no seleccionado.
(Leyendo la base de datos ... 166397 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar elasticsearch-7.15.2-amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Desempaquetando elasticsearch (7.15.2) ...
Configurando elasticsearch (7.15.2) ...
### NOT starting on installation, please execute the following statements to configure elasticsea
rch service to start automatically using systemd
 sudo systemctl daemon-reload
 sudo systemctl enable elasticsearch.service
### You can start elasticsearch service by executing
 sudo systemctl start elasticsearch.service
Created elasticsearch keystore in /etc/elasticsearch/elasticsearch.keystore
Procesando disparadores para systemd (237-3ubuntu10.53) ...
Procesando disparadores para ureadahead (0.100.0-21) ...
```

Ilustración 99. Guía de instalación: Instalación del ELK reducido (parte 3)

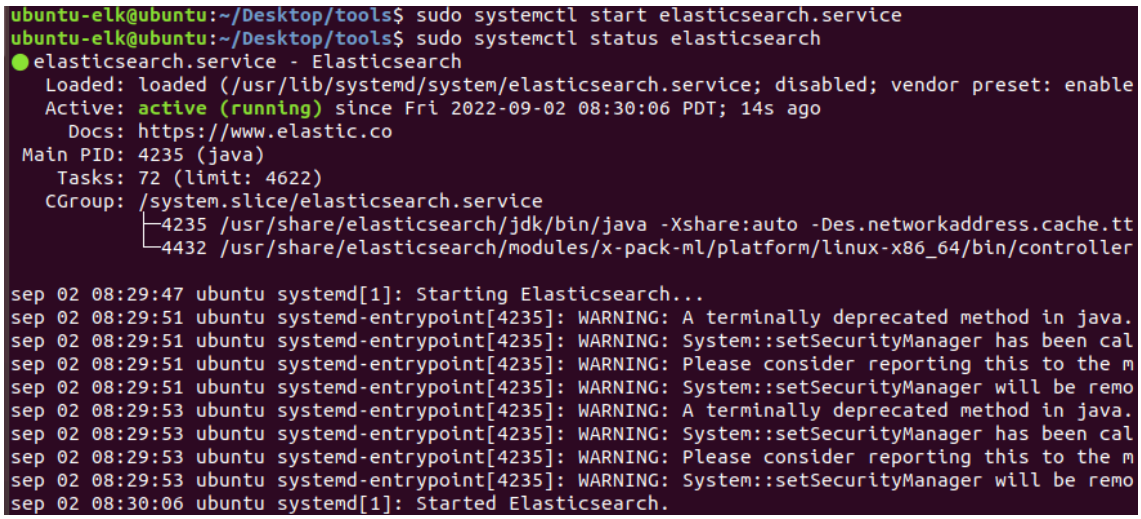

Ilustración 100. Guía de instalación: Instalación del ELK reducido (parte 4)

Como puede verse en la última captura, se ha iniciado el servicio y comprobado que está funcionando; lo cuál es correcto. *Elasticsearch* es configurado por defecto de manera que solo está al alcance a nivel local. Para que pueda ser observado por otros equipos, es necesario modificar su fichero de configuración. Por sencillez de la guía, y de la demo técnica que será realizada; se ha indicado en el primer campo que *Elasticsearch* está al alcance de cualquier red o subred que pueda comunicarse con el equipo donde se aloja esta herramienta, pero se recomienda establecerla únicamente para la red de la organización. El segundo campo, indica la IP de la máquina.

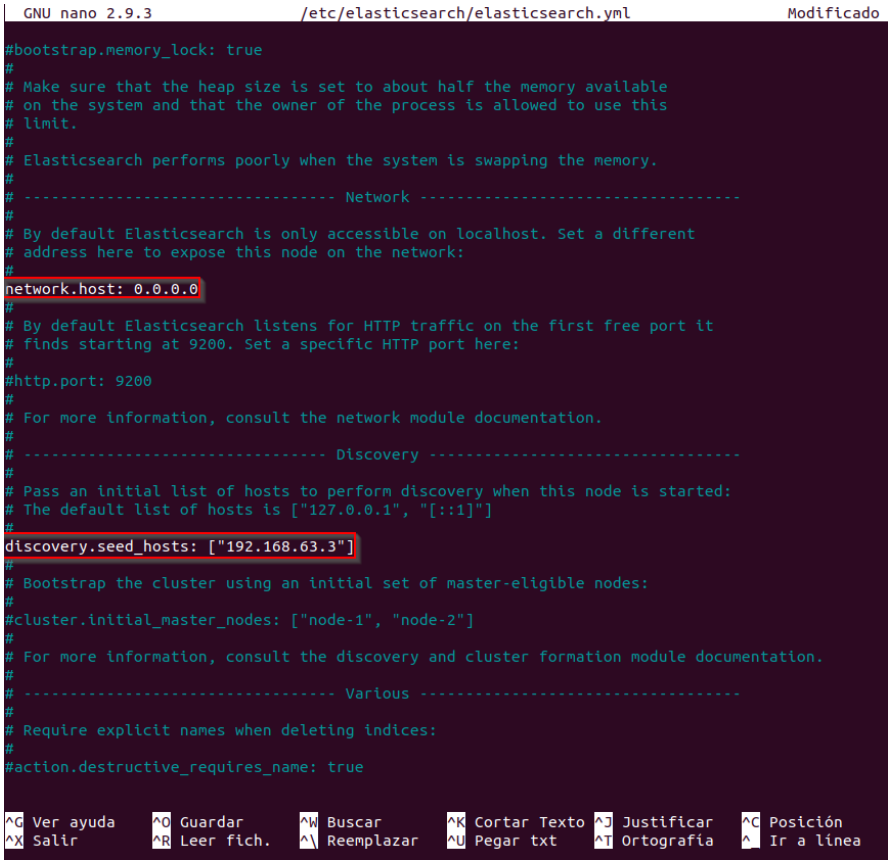

Ilustración 101. Guía de instalación: Instalación de ELK reducido (parte 5)

Se comprueba que *Elasticsearch* es visible desde la máquina *Windows*.

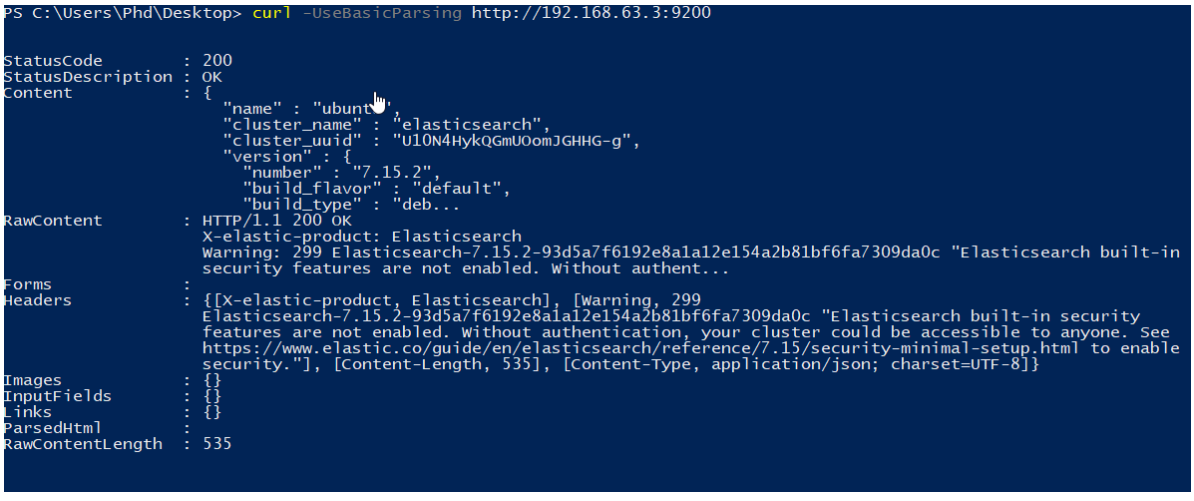

Ilustración 102. Guía de instalación: Instalación de ELK reducido (parte 6)

En esta etapa, *winlogbeat* ya puede enviar datos, pero no pueden visualizarse ni tratarse la manera adecuada para generar las alertas que lleguen a *TheHive Project*. Se procede a la instalación de *Kibana*, que solventará este problema. Nuevamente, se utilizará *curl* para la descarga y *dpkg* para la descompresión e instalación de la herramienta:

```
ubuntu-elk@ubuntu:~/Desktop/tools$ curl -L -O https://artifacts.elastic.co/downloads/kibana/kibana-7.15.2-amd64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  274M  100  274M    0     0  9601k      0  0:00:29  0:00:29 --:--:-- 9776k
ubuntu-elk@ubuntu:~/Desktop/tools$ sudo dpkh -i kibana-7.15.2-amd64.deb
[sudo] contraseña para ubuntu-elk:
sudo: dpkh: orden no encontrada
ubuntu-elk@ubuntu:~/Desktop/tools$ sudo dpkg -i kibana-7.15.2-amd64.deb
Seleccionando el paquete kibana previamente no seleccionado.
(Leyendo la base de datos ... 167520 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar kibana-7.15.2-amd64.deb ...
Desempaquetando kibana (7.15.2) ...
Configurando kibana (7.15.2) ...
Creating kibana group... OK
Creating kibana user... OK
Created Kibana keystore in /etc/kibana/kibana.keystore
Procesando disparadores para systemd (237-3ubuntu10.53) ...
Procesando disparadores para ureadahead (0.100.0-21) ...
```

Ilustración 103. Guía de instalación: Instalación de ELK reducido (parte 7)

Y se inicia el proceso:

```
ubuntu-elk@ubuntu:~/Desktop/tools$ systemctl start kibana
ubuntu-elk@ubuntu:~/Desktop/tools$ systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; disabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-09-02 08:53:05 PDT; 4s ago
     Docs: https://www.elastic.co
 Main PID: 4940 (node)
    Tasks: 11 (limit: 4622)
   CGroup: /system.slice/kibana.service
           └─4940 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli/dist --logging.dest=/var/log/kibana/kibana.log --pid.file=/run/kibana/kibana.pid

sep 02 08:53:05 ubuntu systemd[1]: Started Kibana.
```

Ilustración 104. Guía de instalación: Instalación de ELK reducido (parte 8)

Si se accede al puerto de *Kibana* por defecto, con la dirección IP de *loopback*; puede observarse que el proceso ya está funcional:

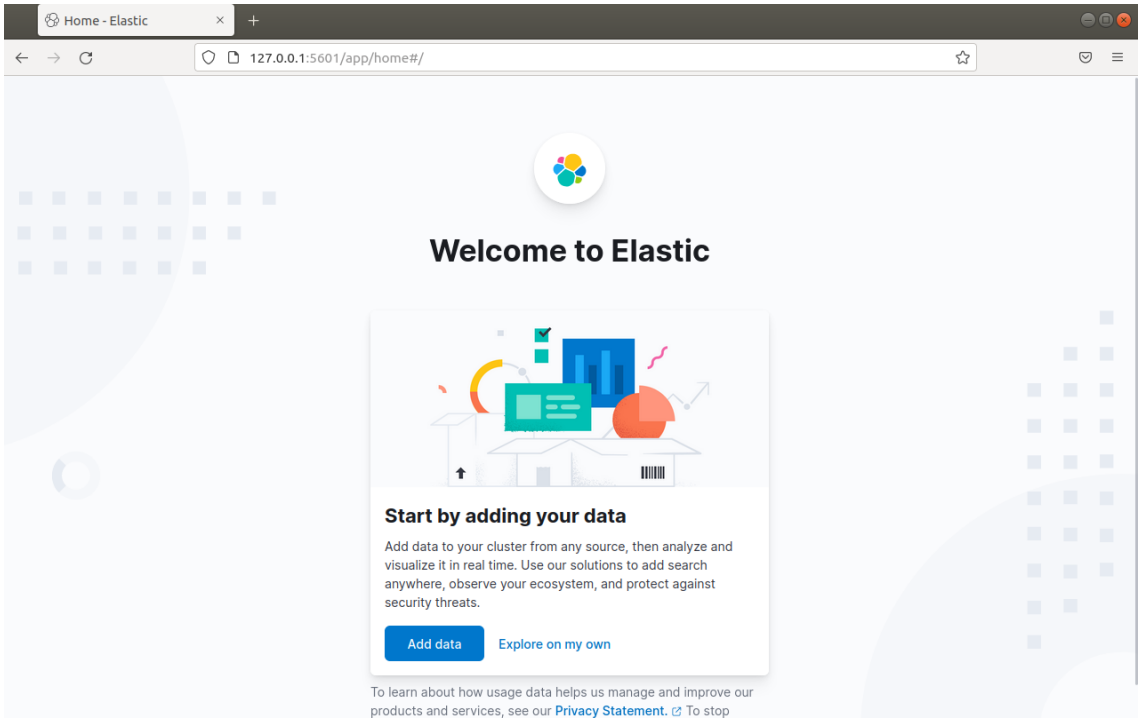

Ilustración 105. Guía de instalación: Instalación de ELK reducido (parte 9)

Restaría configurar su fichero de configuración para que la aplicación fuese accesible desde el resto de la red, y para indicar las instancias de *Elasticsearch* que van a ir relacionadas a esta dependencia de *Kibana*:

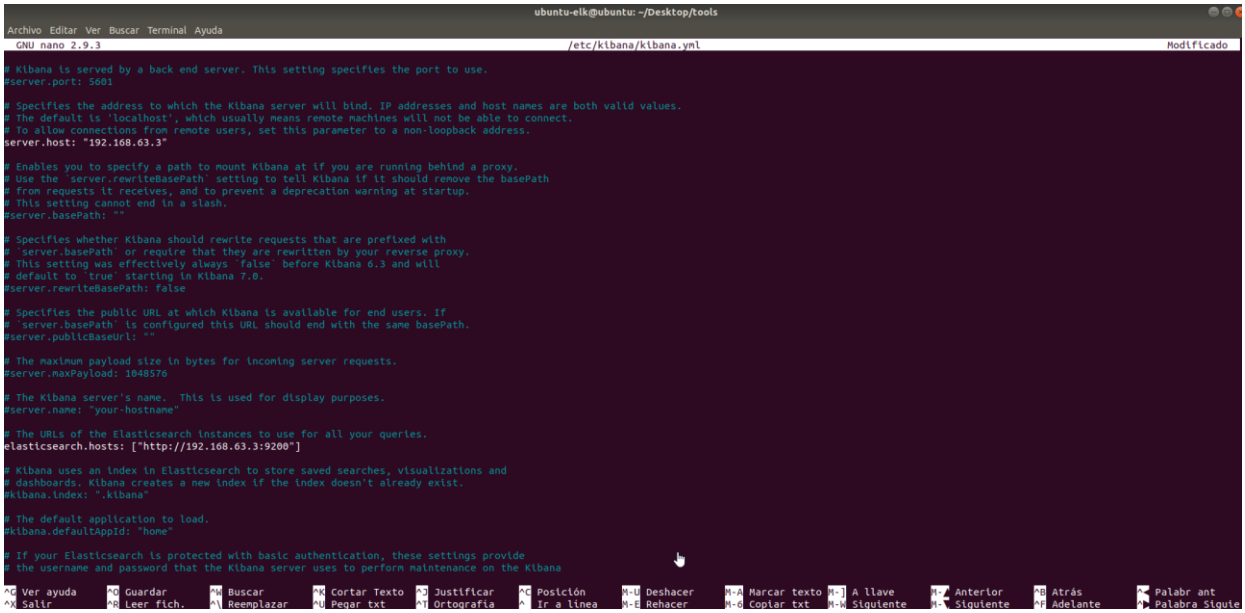

Ilustración 106. Guía de instalación: Instalación de ELK reducido (parte 10)

La información pionera de *Sysmon*, ya empieza a mostrarse en *Kibana*:

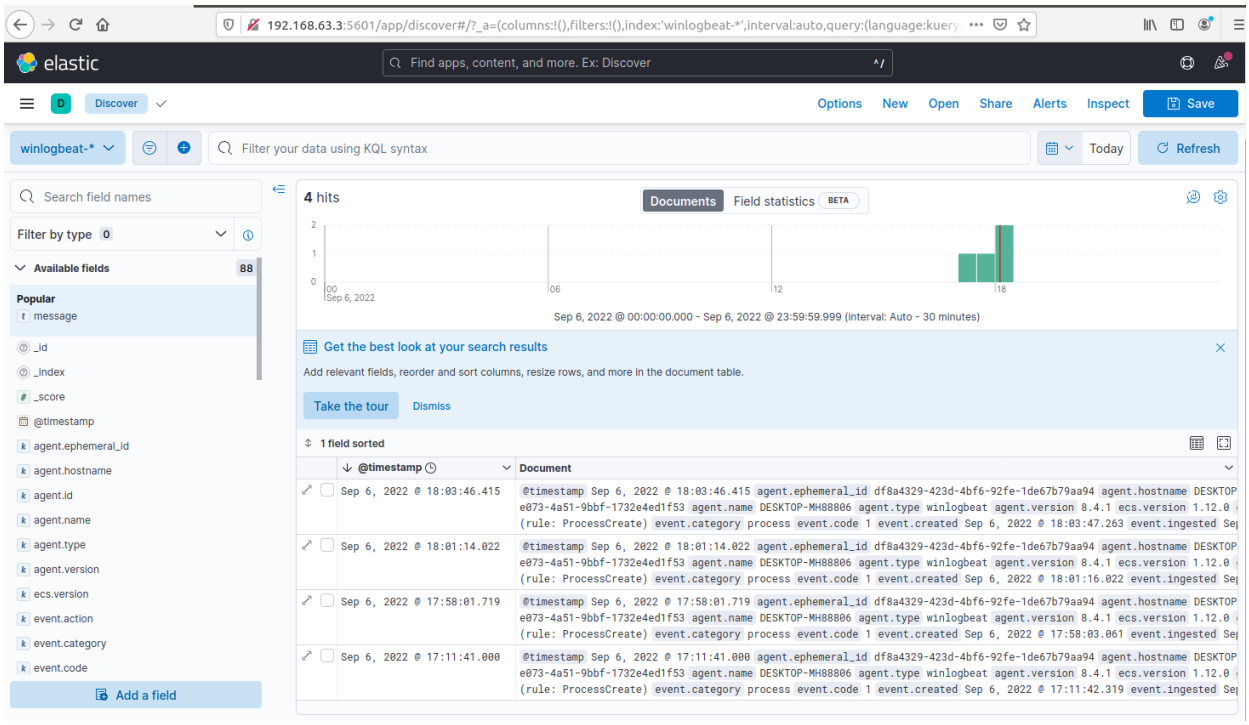

Ilustración 107. Guía de instalación: Instalación de ELK reducido (parte 11)

	process.command_line	C:\Windows\Microsoft.NET\Framework\v4.0.30319\msbuild.exe C:\Users\Phd\Desktop\MSBuild-Powersherless\TEST\readme.txt
	process.entity_id	{6c03b9bf-6f62-6317-9804-000000002100}
	process.executable	C:\Windows\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe
	process.hash.md5	8fdf47e0ff70c40ed3a17014aeea4232
	process.hash.sha256	ed9884bac608c06b7057037cc91d90e4ae5f74dd2dbce2af476699c6d4492d82
	process.name	MSBuild.exe
	process.parent.args	[C:\Program Files\Microsoft Office\Root\Office16\EXCEL.EXE, C:\Users\Phd\Desktop\MSBuild-Powersherless\TEST\TestAPT1.xls]
	process.parent.args_count	2
	process.parent.command_line	"C:\Program Files\Microsoft Office\Root\Office16\EXCEL.EXE" "C:\Users\Phd\Desktop\MSBuild-Powersherless\TEST\TestAPT1.xls"
	process.parent.entity_id	{6c03b9bf-6f60-6317-9404-000000002100}
	process.parent.executable	C:\Program Files\Microsoft Office\root\Office16\EXCEL.EXE

Ilustración 108. Guía de instalación: Instalación de ELK reducido (parte 12)

Por último, es necesario instalar *Elastalert* para la generación de alertas a *TheHive*. Primeramente, es necesario instalar *python3* para obtener todo lo necesario. Si no se encuentra instalado ya en el sistema, se lanza la siguiente orden:

```
sudo apt install python3
```

```
sudo apt install python3-pip
```

o en su defecto:

```
yam install python3
```

Seguidamente, se lanza el comando:

```
pip3 install elastalert
```

Una vez finalizada la descarga, se accede a la ruta *'/opt/elastalert'*. Creamos una nueva carpeta llamada *rules* mediante *mkdir* y se configura el fichero *'config.yaml'* de manera que todos los campos queden comentados menos los que se indican:

- **rules_folder:** rules
- **run_every:** minutes: 1
- **buffer_time:** minutes: 15
- **es_host:** 192.168.63.3
- **es_port:** 9200
- **use_ssl:** False
- **verify_certs:** False
- **writeback_index:** elastalert_status
- **writeback_alias:** elastalert_alerts
- **alert_time_limit:** days: 2

A continuación, se crean los índices en *Elasticsearch* mediante *'elastalert-create-index'*. Estos serán mostrados en el gestor de índices de *Kibana:*

Ilustración 109. Guía de instalación: Instalación de ELK reducido (parte 13)

Accediendo a la carpeta de *'examples_rules'* se muestran una infinidad de reglas que pueden ser aplicadas al generador de alertas. Entre ellas se incluye una para un agente *Wazuh*, muy parecido a lo que necesitamos para pasar nuestras alertas a *TheHive*, y será el que copiemos como ejemplo en la carpeta *'rules'* que se creó anteriormente. Hay que tener en cuenta que deben generarse 24 reglas diferentes; una por cada ID de evento que genera *Sysmon* y que transmite *winlogbeat*:

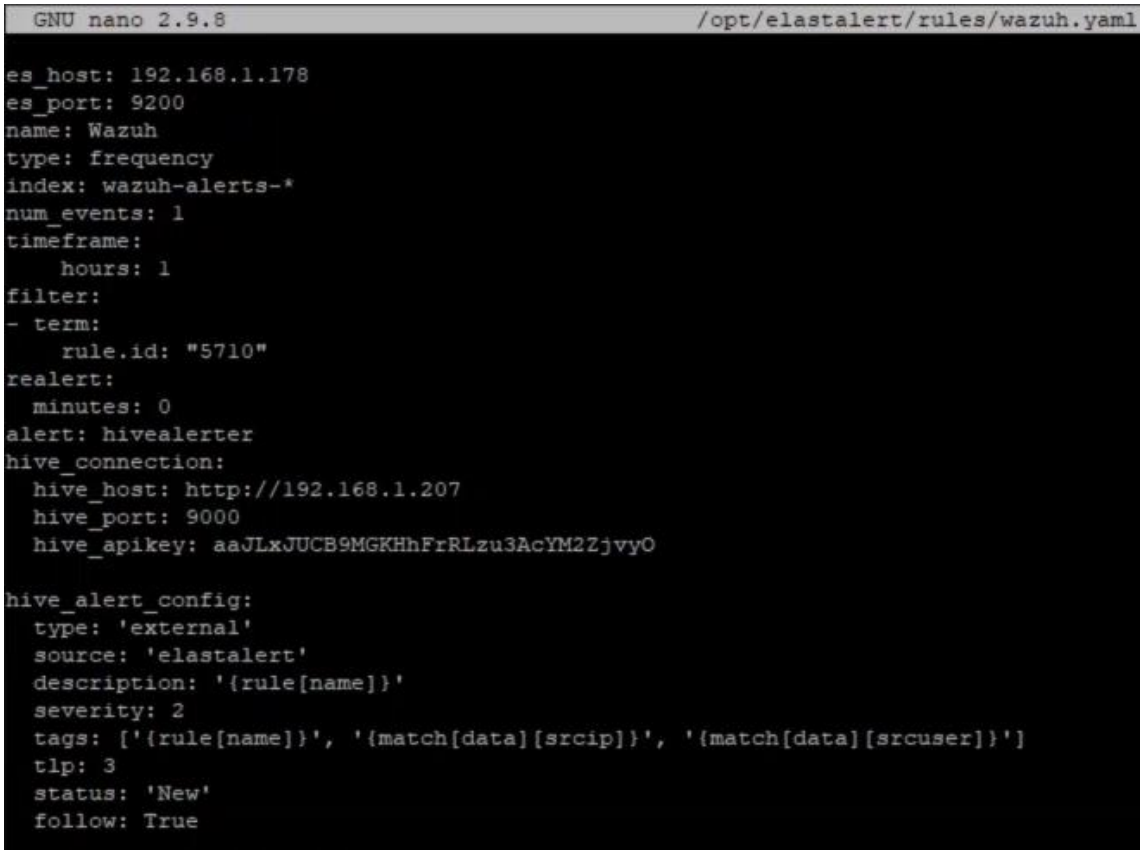

Ilustración 110. Guía de instalación: Instalación de ELK reducido (parte 14)

Los campos que deberían modificarse son:

- **es_host:** IP de *Elasticsearch.*
- **es_port:** Puerto de *Elasticsearch.*
- **name:** Winlogbeat
- **index:** winlogbeat-*
- **timeframe:** minute: 30
- **filter:** event_id: “1”
- **hive_connection:** hive_host: IP de equipo de SOC y hive_apikey: APIkey de
- **hive_alert_config:** Adaptar los *tags* a lo que se necesite según la regla tratada.

De ahora en adelante, cada vez que se genere un evento en sistema, será generada una alerta en función del tipo de regla realizada y con la información que se precise para el tratamiento en *TheHive*.

Se procede a la instalación de los componentes de *TheHive Project* en el equipo restante. Para esta instalación se necesitará:

- Docker.

- Dockstation.
- TheHive.
- MISP.
- Cortex.

Se incluye la lista de comandos que deben introducirse para la correcta implementación de *Docker* así como de *Dockstation*, programas que serán usados para una posterior obtención de las tecnologías necesarias del SIRP de una forma fácil. Los comandos son:

```
sudo apt-get update

sudo apt-get upgrade -y

sudo update-initramfs -u -k -all

sudo apt-get install git apt-transport-https ca-certificates curl
software-properties-common -y

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -

sudo add-apt-repository "deb[arch=amd64]
https://download.docker.com/linux/ubuntu focal stable" -y

sudo apt-cache policy docker-ce

sudo apt install docker docker-ce -y

sudo usermod -aG docker Ubuntu

sudo curl
"https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose
```

A continuación; descargamos *Dockstation* del *github* indicado (https://github.com/DockStation/dockstation/releases/download/v1.5.1/dockstation_1.5.1_amd64.deb), y lo instalamos en */opt*:

```
cd /opt

sudo dpkg -i dockstation*.deb

sudo apt-get install -f
```

Ya tenemos *Docker* y *Dockstation* funcionales. Con introducir 'dockstation' en el *shell* deberíamos ver que es así:

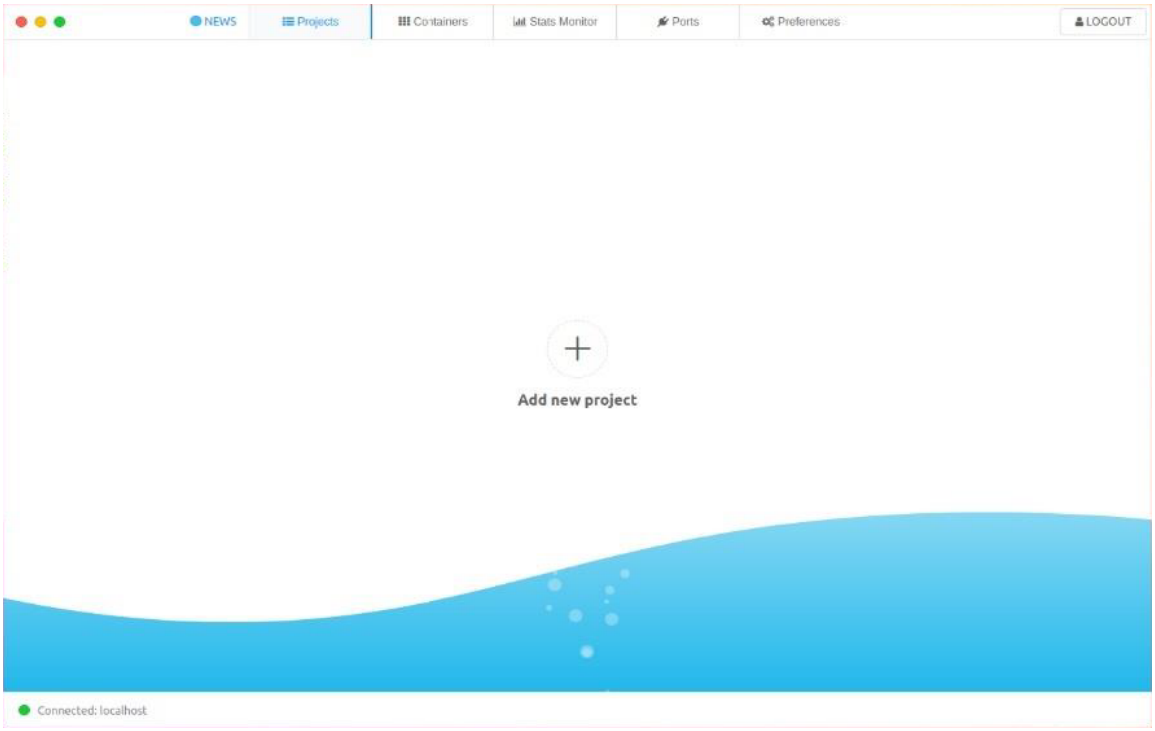

Ilustración 111. Guía de instalación: Instalación de TheHive Project (parte 1)

A la hora de agregar un nuevo proyecto, se nos pide un fichero *.yml*; propio de los de configuración en **Docker.** Volviendo a la carpeta creada con anterioridad; creamos el fichero dentro de ésta:

```
cd /opt/mthc

nano docker-compose.yml
```

Y dejamos el fichero de configuración de la siguiente manera:

```
  GNU nano 2.9.3                                  docker-compose.yml

version: '3.7'
services:
  elasticsearch:
    container_name: elasticsearch
    image: 'docker.elastic.co/elasticsearch/elasticsearch:7.12.0'
    restart: unless-stopped
    ports:
      - '0.0.0.0:9200:9200'
    environment:
      - http.host=0.0.0.0
      - xpack.security.enabled=false
      - cluster.name=hive
      - bootstrap.memory_lock=true
      - script.allowed_types=inline
      - thread_pool.search.queue_size=100000
      - thread_pool.write.queue_size=10000
      - discovery.type=single-node
      - ES_JAVA_OPTS=-Xms256m -Xmx256m
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - './elasticsearch/data:/usr/share/elasticsearch/data'
      - './elasticsearch/logs:/usr/share/elasticsearch/logs'
    networks:
      - Hive
  cortex:
    image: 'thehiveproject/cortex:latest'
    container_name: cortex
    restart: unless-stopped
    depends_on:
      - elasticsearch
    environment:
      - JOB_DIRECTORY=/opt/cortex/jobs
    ports:
      - '0.0.0.0:9001:9001'
    volumes:
      - './cortex/application.conf:/etc/cortex/application.conf'
      - '/opt/cortex/jobs:/opt/cortex/jobs'
      - '/var/run/docker.sock:/var/run/docker.sock'
      - './cortex/log/:/var/log/cortex'
      - '/tmp:/tmp'
    command: '--no-config --no-config-secret'
```

Ilustración 112. Guía de instalación: Instalación de TheHive Project (parte 2)

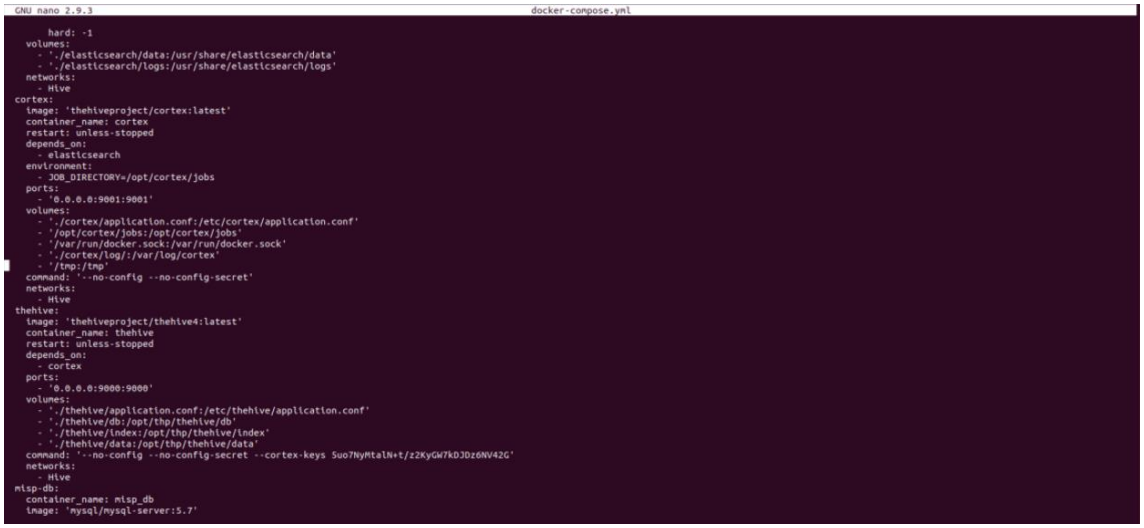

Ilustración 113. Guía de instalación: Instalación de TheHive Project (parte 3)

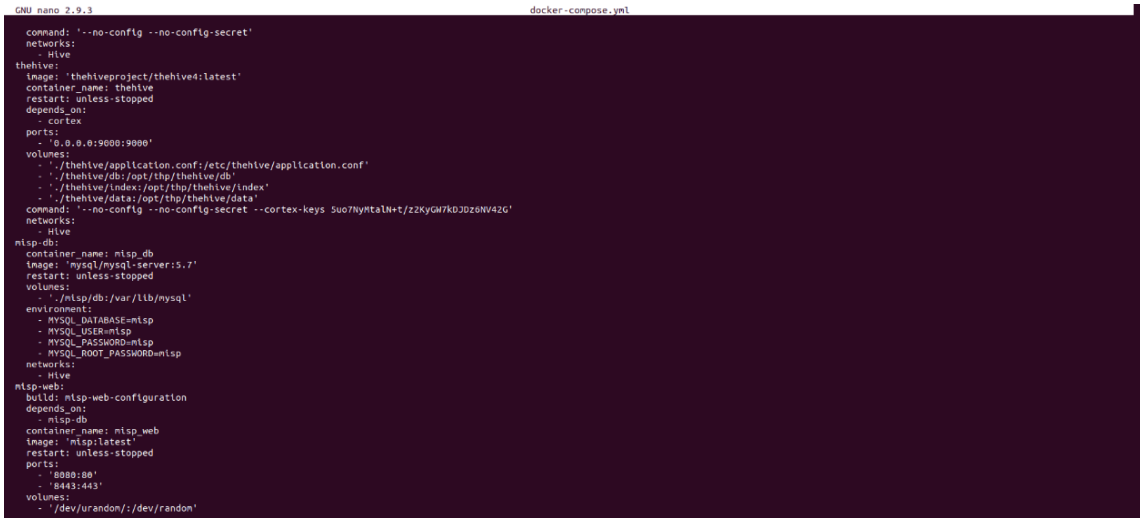

Ilustración 114. Guía de instalación: Instalación de TheHive Project (parte 4)

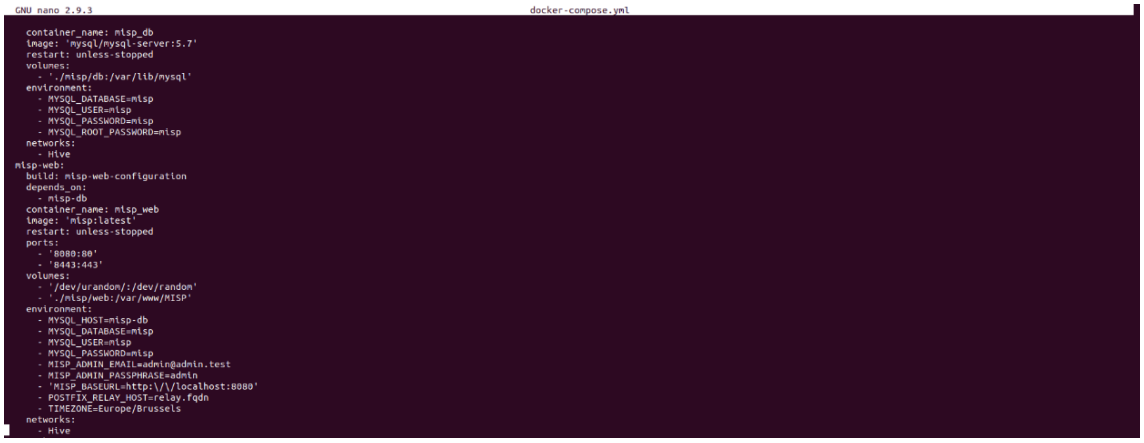

Ilustración 115. Guía de instalación: Instalación de TheHive Project (parte 5)

Guardamos los cambios en el fichero; y volvemos a *DockStation*, donde añadiremos el fichero y lo *buildearemos*. Esto instalará *TheHive*, *MISP*, *Cortex*, *Elasticsearch* y las bases de datos

correspondientes. Cuando el proceso finalice, no lo lanzamos aún. Es completamente necesario la configuración de archivos de configuración de *TheHive* y *Cortex* para la correcta integración de las tecnologías.

```
cd /opt/mthc/thehive

nano application.conf
```

Dejamos el archivo *.conf* tal y como se presenta:

Ilustración 116. Guía de instalación: Instalación de TheHive Project (parte 6)

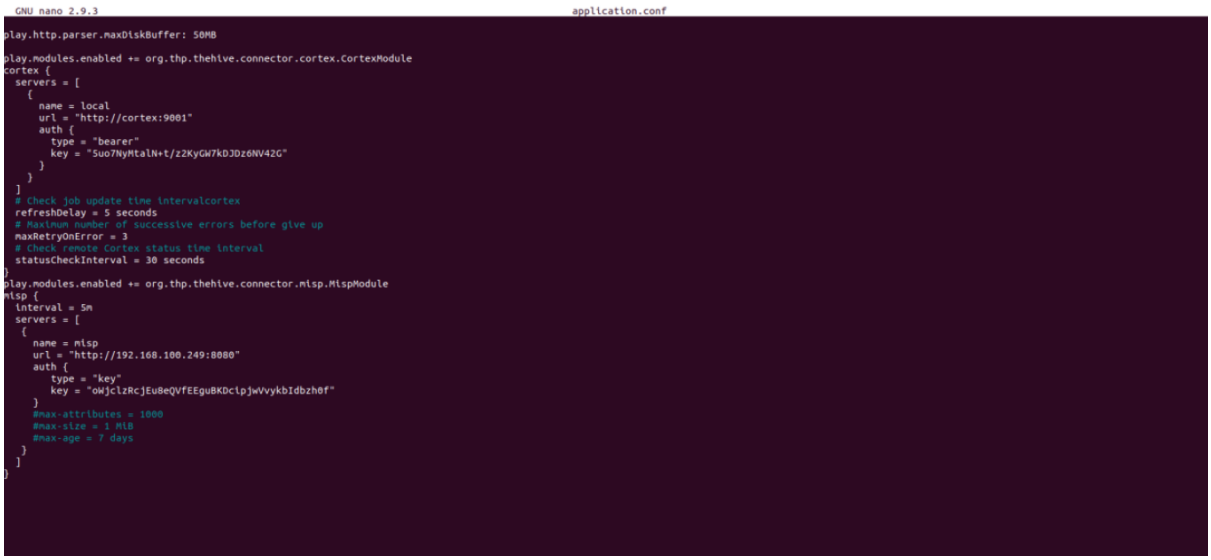

Ilustración 117. Guía de instalación: Instalación de TheHive Project (parte 7)

Hacemos lo propio con Cortex:

```
cd /opt/mthc/cortex

nano application.conf
```

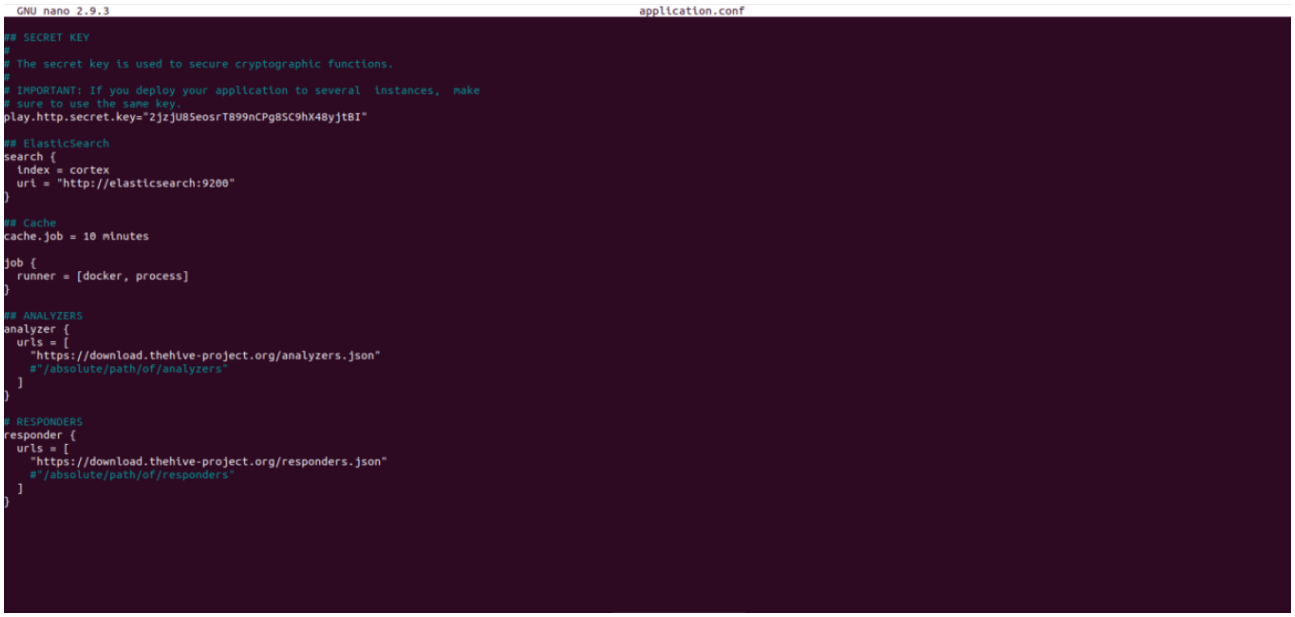

Ilustración 118. Guía de instalación: Instalación de TheHive Project (parte 8)

En lo respectivo a la *secret.key* debe ser la misma para todos los archivos de configuración; para que funcionen las distintas instancias. Por parte de las *keys* indicadas en los módulos de integración del archivo de configuración de *TheHive*; es necesario el lanzamiento de las tecnologías, acceso a *Cortex* y *MISP*, perfil de usuario y obtener la *API key* a usar desde allí. Para el acceso inicial a *MISP*, el usuario es **admin@admin.test** y la contraseña es **admin.** En el caso de *Cortex*, usuario y contraseña son **admin.** En las imágenes se muestra exactamente donde se encuentra la API key una vez se accede:

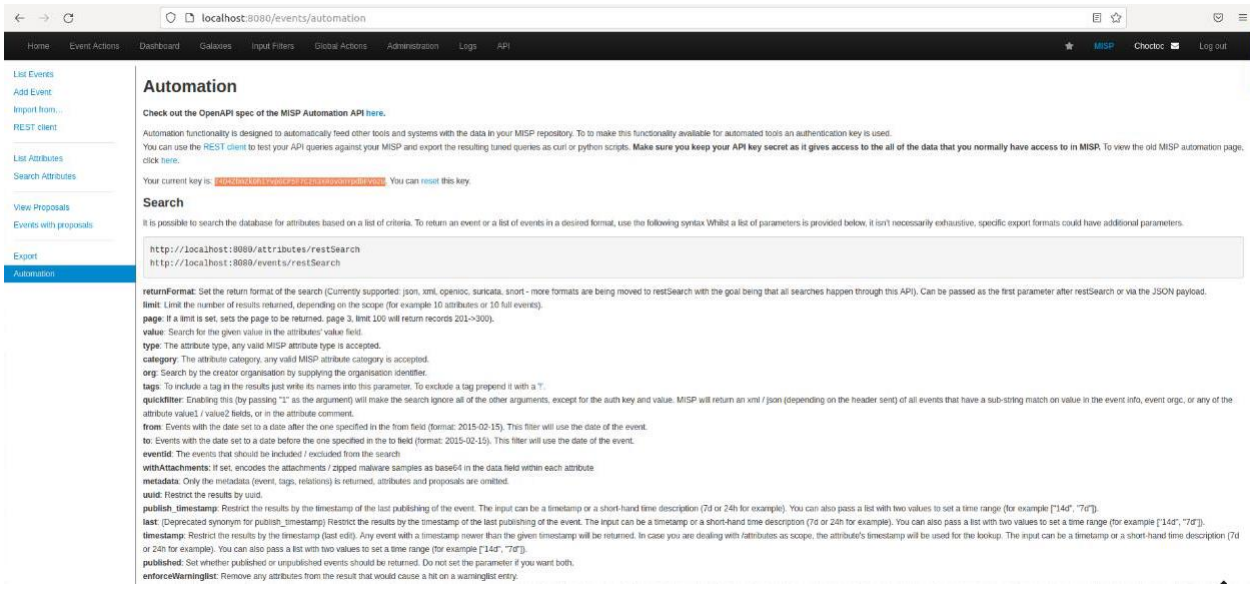

Ilustración 119. Guía de instalación: Instalación de TheHive Project (parte 9)

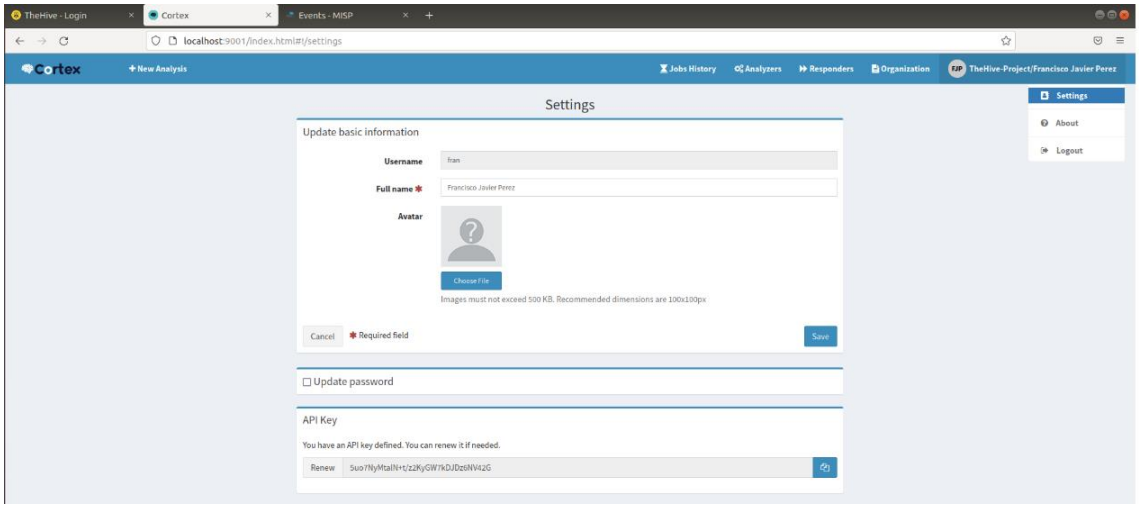

Ilustración 120. Guía de instalación: Instalación de TheHive Project (parte 10)

Si la integración ha sido exitosa, al acceder a *TheHive*; se nos mostrarán los dos iconos pertenecientes a los módulos en la esquina inferior derecha de la pantalla, rodeados por un círculo verde. Si no aparece, o aparecer rodeados por un círculo rojo; la configuración ha fallado en algún punto, y debe *debuggearse*.

Ilustración 121. Guía de instalación: Instalación de TheHive Project (parte 11)

Con todo esto, la completitud de la arquitectura queda implementada; permitiendo la utilización de la solución EDR esperada con todas sus funciones.

C. MANUAL DE USUARIO

El manual de usuario se resume de una manera muy breve con las funciones más importantes que pueden realizarse:

- **Configuración del fichero *Sysmon*:** Se accede al fichero de configuración de *Sysmon* proporcionado en el proyecto y se sigue la guía que viene indicada. Como normativa general, debe añadirse una regla en el apartado de inclusión o exclusión del ID que se trate de la siguiente manera:

```
Full example: <Rule name="MitreRef=T1203,Technique=Exploitation for Client Execution,Tactic=Execution,Alert=Office Hacking
                                    Detected,kpp=y,kp=y" groupRelation="and">
```

Ilustración 122. Manual de usuario - Añadiendo reglas Sysmon

- **Configuración del *script* EDR:** Se accede al fichero *script* EDR proporcionado y se añade una *flag* utilizando el lenguaje .NET propio de *Powershell*. No existe una guía detallada de este paso, ya que requiere conocimientos de dicho lenguaje; sin embargo, si que es necesario que el condicional que indique el nuevo funcionamiento, vaya precedido de un guión. Por ejemplo: "-newfunc".
- **Añadir reglas YARA:** Se accede a la carpeta de *'yara-rules'* de la solución *EDR* y se crean las reglas YARA requeridas. Es necesario conocimiento del lenguaje que manejan los ficheros *.yar*.
- **Añadir reglas *Elastalert:*** Indicado en la sección 'Guía de instalación'. Deberían añadirse 24 reglas en total, una por regla de *Sysmon* que interactúa con la solución EDR.
- **Consultar casos *TheHive*:**
- **Analizar mediante *Cortex*:** Elegir uno de los observables en *TheHive.* Si uno de estos observables tiene un analizador disponible, podrá pulsarse de manera interactiva. Al cabo de un rato, se devolverá información adicional sobre dicho observable, como tipo de *malware* o localización geográfica de una IP.
- **Compartir información en *MISP*:** Elegir el caso en *TheHive* que quiere compartirse. Elevar la alerta a *MISP*, donde debe existir una cuenta creada previamente. La alerta será compartida y podrá ser visualizada por otros usuarios con instancia de *MISP* en su sistema.

D. GUÍA DE SOLUCIONADO DE ERRORES

Se exponen a continuación algunas de las soluciones a los posibles errores que pueden aparecer durante la instalación y en la configuración de la aplicación:

- **"¿Por qué no se está almacenando el *hash* correctamente en la carpeta de cuarentena?".**

  Hay que tener en cuenta que la manera en la que el EDR maneja los elementos eliminados que son trasladados a la carpeta de cuarentena viene dada por la forma en la que se constituye el nombre del *path* generado para ese fichero. Dicho nombre está constituido por los *hashes* en el siguiente orden: MD5, SHA256 e IMPHASH. Es decir, cualquier otra manera de ordenar dichos *hashes* resultará en un error del sistema. Del mismo modo, es recomendable añadir una regla que excluya los accesos directos generados por creación de ficheros en el escritorio, ya que el EDR eliminará los ficheros maliciosos que se encuentren en el escritorio y el acceso directo asociado a este; provocando un error en el sistema cuando la solución EDR trate de analizar el acceso directo utilizando el *hash* del fichero original.

- **"Mi instancia de *winlogbeat* recoge más información además de la de *Sysmon*…"**
  La configuración de *winlogbeat* es incorrecta. Debería presentar el siguiente formato:

```
##################### Winlogbeat Configuration Example #########################

# This file is an example configuration file highlighting only the most common
# options. The winlogbeat.reference.yml file from the same directory contains
# all the supported options with more comments. You can use it as a reference.
#
# You can find the full configuration reference here:
# https://www.elastic.co/guide/en/beats/winlogbeat/index.html

# ======================== Winlogbeat specific options ========================

# event_logs specifies a list of event logs to monitor as well as any
# accompanying options. The YAML data type of event_logs is a list of
# dictionaries.
#
# The supported keys are name, id, xml_query, tags, fields, fields_under_root,
# forwarded, ignore_older, level, event_id, provider, and include_xml.
# The xml_query key requires an id and must not be used with the name,
# ignore_older, level, event_id, or provider keys. Please visit the
# documentation for the complete details of each option.
# https://go.es.io/WinlogbeatConfig

winlogbeat.event_logs:
  - name: Microsoft-Windows-Sysmon/Operational

# ====================== Elasticsearch template settings ======================

setup.template.settings:
  index.number_of_shards: 1
  #index.codec: best_compression
  #_source.enabled: false


# ================================== General ===================================

# The name of the shipper that publishes the network data. It can be used to group
# all the transactions sent by a single shipper in the web interface.
#name:

# The tags of the shipper are included in their own field with each
# transaction published.
#tags: ["service-X", "web-tier"]

# Optional fields that you can specify to add additional information to the
# output.
#fields:
#  env: staging
```

Ilustración 123. Solucionado de errores - configuración winlogbeat

- **"No se están indexando datos desde *winlogbeat* a *Elasticsearch*".**

  La versión de *winlogbeat* es superior a la *Elasticsearch*, en la mayoría de los casos. Se recomienda o bien, actualizar la versión de *Elasticsearch*, o alternativamente, descargar una versión equivalente o inferior a la de *Elasticsearch* para que el envío de datos se produzca sin problemas. Además, se recomienda revisar que no existan filtros añadidos en *Kibana* o que los intervalos de tiempo para muestreo de eventos sean los esperados.

- **"Mi instancia de *Sysmon* está incluyendo toda la información de los procesos del evento del sistema <ID-evento>".**

  Este comportamiento se da cuando se define un grupo de reglas como de exclusión sin incluir nada dentro. Por funcionamiento de *Sysmon*, este formato indica que deben incluirse todos los procesos en los eventos del sistema de la ID respectiva.

- **"Las notificaciones del EDR frente a amenazas se están produciendo varias veces en el equipo *endpoint*".**

  Hay más de una instancia abierta del EDR en el sistema. Hay que denotar que el EDR se inicializa con el sistema como servicio a nivel de *Kernel*, de manera que no es necesario volverlo a ejecutar. Se recomienda encarecidamente, crear una regla en *Sysmon* que indique al propio EDR de matar el nuevo proceso EDR si ya hay uno en ejecución.

- ***"Kibana* no recibe información de *Elasticsearch".***

  Es muy posible que la capacidad de almacenamiento de *Elasticsearch* haya llegado a su límite. Se recomienda hacer una copia de seguridad de los *logs* y reiniciar las herramientas. En su defecto si esto no soluciona el problema, deben revisarse los ficheros de configuración de las dependencias.

- **"Las alertas no se están generando de la manera adecuada/no llegan a *TheHive".***

  La configuración de *Elasticalert* no es correcta, o la capacidad de almacenamiento de casos por *TheHive* ha llegado a su límite. Se recomienda realizar copias de seguridad y vaciar los *logs*.

- **"Mi equipo de SOC no puede analizar una dirección IP, *hash*, etc. mediante *Cortex*".**

  El *parseo* realizado por parte de *Elasticalert* a observables de *TheHive* no es correcto. Para *TheHive* es indispensable que los datos se encuentren correctamente tipificados, de manera que *Cortex* pueda entender con que tipo de información se está tratando para analizarla.

- **"Mi instancia de *TheHive* no está utilizando *Cortex* y *MISP".***

  Sí la utiliza, pero la configuración es incorrecta. Hay que asegurarse que las *apikeys* empleadas para asociar las dependencias son correctas: misma organización creada, correcta redacción de éstas, etc.