BUSINESS Y DIGITAL SCHOOL

Trabajo de fin de Carrera titulado:

"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SIEM CON WAZUH PARA LA DETECCIÓN PROACTIVA DE AMENAZAS CIBERNÉTICAS EN LA EMPRESA ECOILPET S.A."

Realizado por:

NAKLEH SAID ZEIDAN SILVA

Director del proyecto:

MSC. PATRICIO MOREJÓN HIDALGO

Requisito para la obtención del título de:

INGENIERO EN SISTEMAS DE INFORMACIÓN CON ÉNFASIS EN CONTENIDOS INTERACTIVOS

Quito, Agosto 2025

DECLARATORIA

El presente Trabajo de investigación titulado:

"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SIEM CON WAZUH PARA LA DETECCIÓN PROACTIVA DE AMENAZAS CIBERNÉTICAS EN LA EMPRESA ECOILPET S.A."

Realizado por:

NAKLEH SAID ZEIDAN SILVA

Como requisito para la obtención del título:

INGENIERO EN SISTEMAS DE INFORMACIÓN CON ÉNFASIS EN CONTENIDOS INTERACTIVOS

Ha sido dirigido por el profesor:

MSC. PATRICIO MOREJÓN HIDALGO

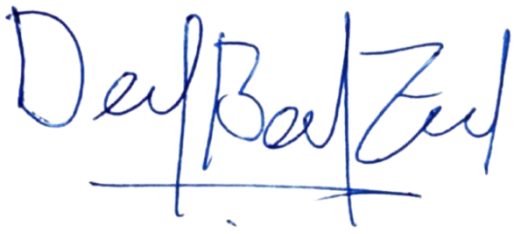
Quien considera que constituye un trabajo original de su autor

MSC. PATRICIO MOREJÓN HIDALGO

PROFESOR

DECLARACION JURAMENTADA

Por medio de la presente, yo, NAKLEH SAID ZEIDAN SILVA, con cédula de ciudadanía Nº 1718814690, declaro bajo juramento que el trabajo aquí presentado es de mi autoría, que no ha sido previamente entregado para la obtención de ningún grado académico o calificación profesional, y que todas las fuentes consultadas han sido debidamente citadas y referenciadas. Asimismo, cedo los derechos de propiedad intelectual derivados de este trabajo a la Universidad Internacional SEK (UISEK), conforme a la Ley de Propiedad Intelectual, su reglamento y la normativa institucional vigente.



Nakleh Said Zeidan Silva

C.C: 171881469-0

LOS PROFESORES INFORMANTES:

PhD. VIVIANA ELIZABETH CAJAS CAJAS

MSc. PATRICIO FERNANDO ALARCON LARREA

Después de revisar el trabajo presentado lo han calificado como apto para su defensa oral ante el tribunal examinador.

PhD. Viviana Cajas

MSc. Patricio Alarcón

DEDICATORIA

Dedico este trabajo con profundo cariño y gratitud a mi familia, por su apoyo incondicional en cada etapa de este camino. A mis padres, por enseñarme el valor del esfuerzo y la perseverancia; a mis seres queridos, por su paciencia, motivación y comprensión en los momentos difíciles. También dedico esta tesis a todos aquellos que creyeron en mí cuando más lo necesitaba. Este logro es tanto mío como de ustedes.

AGRADECIMIENTO

Extiendo mi sincero agradecimiento a la Universidad Internacional SEK y a todos los docentes que durante estos años me brindaron sus conocimientos y guía. A mis amigos y compañeros, con quienes compartí no sólo clases, sino también aprendizajes, desafíos y grandes momentos. Finalmente, gracias a mi familia por su apoyo constante, por creer en mí y por motivarme cada día a seguir adelante.

RESUMEN

Esta tesis presenta el diseño e implementación de un sistema SIEM basado en Wazuh para mejorar la ciberseguridad en ECOILPET S.A. La solución centraliza registros, correlaciona eventos y genera alertas en tiempo real, integrando agentes en Windows y monitoreo agentless en dispositivos de red. Se validó su eficacia mediante simulaciones de ataque con MITRE Caldera y se evaluó el cumplimiento de ISO/IEC 27001 y la Ley Orgánica de Protección de Datos Personales del Ecuador (LOPDP). Los resultados muestran que Wazuh es una opción viable, escalable y de bajo costo para proteger entornos empresariales de tamaño medio, fortaleciendo la defensa de activos digitales y el cumplimiento normativo.

Palabras clave: *Ciberseguridad, detección de amenazas, ISO 27001, SIEM, Wazuh.*

ABSTRACT

This thesis presents the design and implementation of a Security Information and Event Management (SIEM) system using Wazuh to enhance cybersecurity at ECOILPET S.A. The solution centralizes logs, correlates events, and generates real-time alerts, combining Windows agents with agentless monitoring for network devices. Its effectiveness was validated through controlled attack simulations using MITRE Caldera, and compliance with ISO/IEC 27001 and Ecuador's Organic Law on Personal Data Protection (LOPDP) was assessed. Results show that Wazuh is a viable, scalable, and cost-effective option for protecting medium sized business environments, strengthening digital asset defense and regulatory compliance.

Keywords*:* *Cybersecurity, ISO 27001, SIEM, threat detection, Wazuh.*

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

1. CAPÍTULO I

INTRODUCCIÓN

1.1 Planteamiento del problema

En el contexto actual de transformación digital, las empresas dependen cada vez más de sus sistemas informáticos para operar con eficiencia y continuidad. Sin embargo, esta dependencia tecnológica también las expone a riesgos significativos relacionados con la ciberseguridad, como accesos no autorizados, pérdida de información, sabotajes internos, y ataques externos como "*ransomware*" o "*phishing*" (Von Solms & Niekerk, 2013).

La empresa ECOILPET S.A., dedicada a "Prestación de servicios de asesoramiento, orientación y asistencia operativa a empresas y administraciones públicas en: diseño de métodos contables, programas de contabilidad de costos y procedimientos de control presupuestario." (Superintendencia, 2025), cuenta con una infraestructura informática en crecimiento, pero carece de un sistema centralizado y automatizado de monitoreo y respuesta activa ante incidentes de seguridad. Actualmente, no existe una solución implementada para la correlación de eventos de seguridad, detección proactiva de amenazas ni alertas en tiempo real. Los registros (*logs*) de los sistemas no se gestionan de manera eficiente, y la empresa depende de una vigilancia manual que es limitada, reactiva y propensa a errores.

Esta falta de visibilidad y control incrementa la probabilidad de que ataques cibernéticos pasen desapercibidos, comprometiendo datos sensibles de la organización y sus clientes. Además, pone en riesgo el cumplimiento de normativas como la Ley Orgánica de Protección de Datos Personales (Asamblea Nacional del Ecuador., 2021) y los principios de la norma ISO/IEC 27001 (Humphreys, 2008), afectando su reputación, operatividad y sostenibilidad.

Ante esta situación se hace necesario implementar una solución tecnológica eficaz y asequible que permita a ECOILPET S.A. identificar, registrar, analizar y responder a amenazas de seguridad de manera inmediata. Wazuh, como sistema SIEM (Security Information and Event Management) de código abierto, se presenta como una alternativa viable para cubrir esta brecha, permitiendo a la empresa contar con "*dashboards"* interactivos, reglas de correlaciones personalizadas y capacidades de respuesta inmediata ante eventos anómalos.

1.2 Justificación

La presente propuesta responde a una necesidad concreta y crítica para ECOILPET S.A.: fortalecer su infraestructura de seguridad informática mediante la implementación de una solución capaz de detectar y responder proactivamente a las amenazas cibernéticas. En un contexto en el que la información se convierte en uno de los activos más valiosos, disponer de herramientas que garanticen su protección es una prioridad estratégica. La elección de Wazuh como solución tecnológica se fundamenta en su carácter de código abierto, compatibilidad multiplataforma y capacidad para generar alertas, correlacionar eventos y visualizar incidentes en tiempo real mediante "*dashboards"* personalizados.

Ecuador ha mostrado avances significativos en materia de ciberseguridad. En septiembre de 2024, el país alcanzó 87,18 puntos sobre 100 en el Índice Global de Ciberseguridad de la Unión Internacional de Telecomunicaciones, mejorando su posición en la región y ubicándose en el grupo "*T2 Advancing*" junto a naciones como Canadá y México (Ministerio de Telecomunicaciones y de la Sociedad de la Información del Ecuador, 2024)

Asimismo, en enero y marzo de 2023 se llevaron a cabo talleres presenciales sobre ciberhigiene y buenas prácticas de ciberseguridad dirigidos a personal de instituciones como la Corte Nacional de Justicia, el Ministerio de Educación y el Ministerio de Inclusión Económica, con más de 273 participantes. En mayo del mismo año, se realizó un taller virtual sobre la

plataforma "*Cyber Hygiene e-Learning*" con 90 asistentes, fortaleciendo las capacidades del personal público en materia de concienciación digital (Ministerio de Telecomunicaciones y de la Sociedad de la Información del Ecuador., 2023) Sin embargo, según Kaspersky, (2023) en el año 2023 Ecuador se posiciona en el segundo lugar de los países mas atacados en América Latina, sólo por detrás de Brasil.

En el ámbito normativo, Ecuador aprobó en agosto de 2022 su primera Estrategia Nacional de Ciberseguridad (ENC), que abarca gobernanza, resiliencia cibernética, ciberdefensa, prevención del delito digital, formación y cooperación internacional. Esta estrategia fue respaldada por un comité interdisciplinar y es el marco rector para fortalecer la protección digital del país (Ministerio de Telecomunicaciones y de la Sociedad de la Información del Ecuador., 2022)

Frente a este escenario de creciente vulnerabilidad y evolución institucional, se justifica el uso de Wazuh como una solución SIEM eficiente y accesible. Su naturaleza de código abierto evita los altos costos de licencias comerciales, lo cual es crucial para empresas medianas. Además, su gran capacidad para centralizar registros, detectar anomalías en tiempo real y generar "*dashboards"* personalizados fortalece la postura defensiva de ECOILPET, al alinearse con la Ley Orgánica de Protección de Datos Personales y los estándares internacionales ISO/IEC 27001. La propuesta contempla también simulaciones controladas de ataques para medir la efectividad del sistema y capacitar al personal técnico, consolidando una cultura de seguridad consciente en la organización.

Por último, la solución es altamente replicable y escalable, lo que permite adaptar la implementación a otras empresas de servicios, cooperativas u organizaciones ecuatorianas en condiciones similares. De esta forma, ECOILPET no solo mejora su competitividad y

seguridad, sino que también contribuye al posicionamiento de la ciberseguridad como una práctica corporativa central en el país.

1.3 Metodología

El proyecto combina investigación descriptiva-explicativa y un enfoque de ingeniería de sistemas aplicado. Desde la perspectiva investigativa, se caracterizó la situación actual de ciberseguridad en ECOILPET S.A. y se analizaron las funcionalidades de herramientas SIEM. Esto permitió identificar cómo la implementación de Wazuh puede mitigar brechas de seguridad, planteando soluciones aplicadas a un problema real. En paralelo, la ingeniería de sistemas brindó un marco diseñado de manera estructurada para abarcar el desarrollo, implementación y evaluación del sistema SIEM piloto.

Para el ciclo de desarrollo de software se adoptó un enfoque "*DevSecOps"* para el desarrollo del SIEM, combinando ciclos iterativos de implementación con validaciones continuas de seguridad. Esto permitió integrar controles desde las fases iniciales (como la creación de reglas personalizadas) y probar su efectividad mediante emulación de ataques con MITRE Caldera antes del despliegue en producción.

1.3.1 Tipo de enfoque de la investigación

La investigación es de tipo descriptiva explicativa y aplicada. Es descriptiva porque documenta el estado actual de la ciberseguridad en ECOILPET S.A., incluyendo activos tecnológicos, políticas y procesos. Es explicativa al analizar las causas de las brechas y mostrar cómo el SIEM propuesto puede mitigarlas. Es aplicada porque el objetivo es desarrollar una solución parcial funcional que sirva de base para una implementación más amplia en el futuro.

1.3.2 Fases metodológicas

La implementación de este trabajo de titulación se llevó a cabo siguiendo un proceso estructurado en fases, lo que permitió un desarrollo sistemático y una gestión eficiente de los recursos. Cada fase se concibió para construir sobre la anterior, asegurando que los hallazgos y decisiones de una etapa informaran y validaran la siguiente. Inicialmente, el proyecto transitó por una fase de investigación preliminar y marco conceptual. Esta etapa fue crucial para sentar las bases teóricas y técnicas, comenzando con una investigación documental exhaustiva que incluyó una revisión crítica de la bibliografía científica y técnica sobre ciberseguridad, gestión de eventos e información de seguridad y la arquitectura específica de Wazuh.

Este proceso implicó el estudio de artículos académicos, informes técnicos, y la consulta de la documentación oficial de Wazuh, junto con el análisis de experiencias prácticas de implementación en organizaciones con características similares a ECOILPET S.A. Simultáneamente, se realizó un análisis profundo de la normativa vigente, enfocándose en la Ley Orgánica de Protección de Datos Personales del Ecuador (LOPDP) para asegurar la adhesión a los requisitos legales de protección de datos, y en la norma internacional ISO/IEC 27001, que sirvió como un marco que brinda las buenas prácticas para gestionar la seguridad de la información. La sistematización de todos estos hallazgos teóricos fue fundamental para construir un marco conceptual robusto que no solo guiara las decisiones técnicas, sino que también contextualizara la problemática y la solución propuesta.

Con el marco conceptual establecido, la siguiente etapa fue la fase de diseño y planificación. Esta se centró en la planificación detallada y el diseño de la solución SIEM adaptada a ECOILPET S.A. Se inició con un análisis de requerimientos técnicos y del entorno operacional actual de la empresa, lo que implicó la identificación de los activos tecnológicos que serían objeto de monitoreo, incluyendo servidores, estaciones de trabajo y dispositivos de red como

"*switches*" y "*routers*", así como la evaluación de su criticidad dentro de la infraestructura. Adicionalmente, se definieron con precisión los requerimientos de hardware (especificaciones del servidor para Wazuh Manager, recursos de almacenamiento y procesamiento necesarios), conectividad de red y el software específico (versiones de sistemas operativos, de Wazuh y de Elastic Stack).

Como parte de la selección de herramientas, se evaluó el ecosistema de Wazuh, comprendiendo a fondo el funcionamiento de sus componentes (Manager, Agentes, Reglas, Decodificadores y su integración con Elastic Stack). Para las pruebas de validación, se seleccionó MITRE Caldera como la herramienta principal de simulación de ataques, justificando su elección por su capacidad avanzada de emulación de adversarios y su alineación directa con el marco de trabajo MITRE ATT&CK. Finalmente, esta fase culminó con el diseño detallado de la arquitectura lógica y física del sistema SIEM propuesto para ECOILPET, delineando cómo se integrarían todos los componentes.

Una vez definidos el diseño y la planificación, la fase de implementación y configuración se dedicó a la materialización del sistema SIEM. Este proceso comenzó con la instalación del Wazuh Manager en el entorno predefinido, seguido de su integración con Elastic Stack, lo que incluyó la configuración de Elasticsearch para el almacenamiento de registros y de Kibana para análisis y visualización de datos. Posteriormente, se procedió con el despliegue y la configuración de los agentes Wazuh en los servidores Windows, estaciones de trabajo y otros puntos críticos de la infraestructura de ECOILPET, a la vez que se configuró el monitoreo "*agentless*" para aquellos dispositivos de red que no permiten la instalación de software, utilizando protocolos estándar como "*Syslog*".

Un aspecto clave fue la creación y personalización de reglas de correlación y decodificadores específicos adaptados al contexto y las actividades de la empresa, lo que permitiría una

detección precisa de incidentes. Complementariamente, se diseñaron y crearon "*dashboards*" personalizados en Kibana para ofrecer una visualización interactiva y significativa de los eventos de seguridad. Para preparar la etapa de pruebas, se configuró MITRE Caldera, estableciendo los agentes "*Sandcat*" necesarios para emular las técnicas de ataque que se validarían posteriormente. El plugin "*Sandcat*" proporciona a Caldera su agente predeterminado, Sandcat, el cual está desarrollado en GoLang para ser compatible con múltiples plataformas y puede compilarse actualmente para Windows, Linux y MacOS (Caldera, 2025)

La fase de pruebas, validación y evaluación fue fundamental para verificar la eficacia y el correcto funcionamiento del sistema SIEM implementado. Se inició con el diseño de pruebas rigurosas y la definición de casos de uso específicos, basados en técnicas del framework MITRE ATT&CK, que eran relevantes para los riesgos de ECOILPET S.A. Estos escenarios guiaron la ejecución controlada de ataques simulados utilizando MITRE Caldera, lo que permitió validar la capacidad del SIEM para detectar comportamientos anómalos en tiempo real. Durante estas pruebas, se realizó un registro detallado de los resultados, documentando cada alerta generada por Wazuh, incluyendo su precisión (verdaderos positivos, falsos positivos y falsos negativos) y el tiempo de detección de cada incidente.

Paralelamente, se llevó a cabo un análisis y evaluación del desempeño del sistema, cuantificando métricas clave como la tasa de detección de ataques y la mejora en la cobertura de monitoreo de activos. Finalmente, se realizó una comparación entre el estado de ciberseguridad inicial de ECOILPET S.A. y el estado posterior a la implementación del SIEM, evaluando cómo el sistema contribuía al cumplimiento normativo de la ISO/IEC 27001 y la LOPDP, al proporcionar mecanismos robustos para la gestión y el registro de eventos de seguridad.

Finalmente, la fase de conclusiones y recomendaciones consistió en la síntesis de los hallazgos y la propuesta de directrices para el futuro. Se elaboraron las conclusiones generales del proyecto, donde se validó el cumplimiento de los objetivos planteados y se resumieron los resultados clave de la implementación y las pruebas de validación, destacando la viabilidad y efectividad de Wazuh como solución SIEM para ECOILPET. Asimismo, se detallaron los aportes tangibles a la empresa ECOILPET S.A., explicando cómo el sistema implementado fortalece su postura de seguridad, mejora la visibilidad de eventos y facilita el cumplimiento normativo.

A partir de la experiencia adquirida, se formularon recomendaciones claras para futuras implementaciones o expansiones del SIEM, y se propuso un plan para la sostenibilidad y el mantenimiento a largo plazo del sistema, incluyendo políticas de revisión periódica y actualización de componentes. Finalmente, se identificaron y delinearon líneas futuras de investigación, sugiriendo áreas para profundización académica que surgieron a lo largo del desarrollo del proyecto, y se diseñó una propuesta de plan de capacitación técnica para el equipo de TI de ECOILPET, asegurando la continuidad y eficiencia en la gestión del SIEM.

1.4 Objetivo de la implementación

1.4.1 Objetivo general

Diseñar e implementar un sistema SIEM utilizando Wazuh en un conjunto limitado de endpoints y dispositivos críticos de ECOILPET S.A.

1.4.2 Objetivos específicos

Analizar la situación actual de ciberseguridad y los requisitos normativos aplicables (LOPDP e ISO/IEC 27001) para delimitar el alcance del proyecto piloto.

Diseñar e implementar la arquitectura del SIEM con Wazuh, incluyendo la configuración de agentes y monitoreo agentless en los endpoints seleccionados.

Validar el sistema mediante simulaciones controladas de ataques.

2. CAPÍTULO 2

MARCO TEÓRICO DE CIBERSEGURIDAD Y SIEM

2.1 Ciberseguridad y gestión de eventos

La ciberseguridad busca salvaguardar los activos digitales de las organizaciones contra posibles amenazas. Para ello, es fundamental la gestión de eventos de seguridad, que consiste en la recolección, correlación y análisis continuo de logs y sucesos de redes, sistemas y aplicaciones. En este enfoque, todos los eventos relevantes (intentos de acceso, cambios en archivos, actividad de red, etc.) se centralizan para detectar patrones anómalos o incidentes en tiempo real. Sistemas como SIEM (Security Information and Event Management) implementan esta estrategia, facilitando a los equipos de seguridad un monitoreo holístico de la infraestructura y una respuesta automatizada ante alertas (Kent & Souppaya, 2006). Además, la automatización basada en eventos permite definir respuestas preprogramadas, por ejemplo, aislar un host o aplicar un parche que se ejecutan automáticamente cuando ocurre un evento de seguridad específico, reforzando así la capacidad de reacción de la organización (Kent & Souppaya, 2006)

2.2 Principales amenazas cibernéticas

Las organizaciones actuales enfrentan diversas amenazas cibernéticas en constante evolución. Entre las más relevantes se encuentran las *"Advanced Persistent Threats"* (APT), el *malware*, los ataques a la cadena de suministro y las amenazas dirigidas a infraestructuras críticas. Las APT son ataques prolongados ejecutados de forma sigilosa por adversarios sofisticados y bien financiados, cuyo objetivo suele ser el robo de información o el sabotaje de infraestructuras estratégicas (Chen et al., 2014). El término malware engloba diferentes tipos de software

malicioso virus, troyanos, gusanos, etc. capaces de comprometer la integridad, disponibilidad o confidencialidad de los sistemas corporativos.

Por su parte, los ataques a la cadena de suministro comprometen algún componente del software durante las fases de desarrollo o distribución, lo que permite a los atacantes desplegar código malicioso mediante actualizaciones legítimas. Según Ghanbari y Koskinen (2024), este tipo de ataques ha aumentado significativamente y constituye uno de los vectores con mayor proyección en los próximos años. Finalmente, los ataques contra infraestructuras críticas representan un riesgo elevado para la sociedad, ya que pueden provocar la interrupción de servicios esenciales como la energía, el transporte o las finanzas. En este sentido, Shawe y McAndrew (2023) destacan que los ciberataques dirigidos a este tipo de infraestructura «constituyen un riesgo directo para la seguridad nacional» de los Estados.

2.2 Seguridad basada en eventos

La seguridad basada en eventos se fundamenta en el análisis continuo de los registros generados por redes, sistemas y aplicaciones con el fin de identificar comportamientos anómalos. Para ello, las organizaciones emplean plataformas SIEM (Security Information and Event Management), que permiten recolectar, correlacionar y analizar eventos procedentes de distintas fuentes en tiempo real. González-Granadillo, González-Zarzosa y Díaz (2021) afirman que los SIEM «constituyen el núcleo principal de los centros de operaciones de seguridad», ya que permiten centralizar el monitoreo de la infraestructura y generar alertas ante incidentes. Asimismo, estas plataformas facilitan la automatización de respuestas a partir de políticas predefinidas, lo que contribuye a reducir significativamente el tiempo de reacción ante un incidente. En este sentido, Sheeraz et al. (2024) sostienen que un modelo de seguridad basado en eventos permite obtener una visión integral del estado de la red y responder de forma más eficiente a las amenazas emergentes.

2.3 Sistemas SIEM

El SIEM (Security Information and Event Management) es una tecnología clave en los SOC (Centros de Operaciones de Seguridad). Integra funciones de gestión de información de seguridad (SIM) y gestión de eventos de seguridad (SEM) para proporcionar análisis en tiempo real y archivado histórico de sucesos. Según IBM (2023), "SIEM es una solución de seguridad que ayuda a las organizaciones a reconocer y abordar posibles amenazas antes de que afecten las operaciones". Un SIEM moderno recoge datos de fuentes diversas de la infraestructura (servidores, dispositivos de red, aplicaciones, usuarios, nubes, etc.) (IBM, 2023), correlaciona esos eventos aplicando reglas predefinidas o inteligencia artificial, y genera alertas para el equipo de seguridad. Entre sus funcionalidades principales figuran:

- Gestión de registros y normalización: Agrega eventos de todo tipo (registros de sistema, tráfico de red, alertas de IDS/IPS, aplicaciones, etc.) en una base de datos centralizada. Luego normaliza los formatos heterogéneos a un esquema común para facilitar el análisis.
- Correlación de eventos y análisis: Emplea motores de reglas o ML para detectar patrones complejos en los datos. Por ejemplo, relaciona un acceso inusual (código de evento de login remoto) con una modificación de archivos críticos en un corto lapso, lo que desencadena una alerta de posible intrusión. Esto mejora significativamente el MTTR (Mean Time to Detect/Respond) al reducir alertas falsas y concentrar atención en incidentes reales (IBM, 2023).
- Alertas en tiempo real: Consolida los resultados del análisis en paneles de control centralizados. En un mismo dashboard, los analistas monitorean la actividad global, visualizan tendencias (picos de tráfico o ataques simultáneos) y filtran eventos según riesgos. Un SIEM alerta inmediatamente cuando se cumple una regla crítica (por

ejemplo, ejecución de comando sospechoso). A demás incluye visualizaciones gráficas para facilitar el entendimiento de los datos (IBM, 2023).

- Cumplimiento y reportes: Automatiza la generación de informes auditables para reglamentaciones (PCI-DSS, GDPR, HIPAA, ISO 27001, etc.). Al recopilar y conservar evidencias de seguridad de toda la red, el SIEM simplifica las auditorías de cumplimiento. Muchas soluciones comerciales ofrecen plantillas de reportes predefinidos para normas específicas, reduciendo la carga de documentación (IBM, 2023).
- Inteligencia de amenazas y respuesta: Muchos SIEM integran feeds de inteligencia de amenazas externas (listas de IP maliciosas, indicadores de compromiso). Esto permite comparar internamente la actividad con amenazas conocidas. Asimismo, los SIEM pueden combinarse con sistemas SOAR (Security Orchestration, Automation and Response) para automatizar pasos de mitigación (bloquear un IP, aislar un host) según las alertas recibidas (IBM, 2023).

2.4 Arquitectura de un SIEM

Aunque existen variadas implementaciones, la arquitectura típica de un SIEM es modular. En la capa inicial se ubican los recolectores de datos: agentes instalados en los endpoints, conectores a APIs, sistemas de syslog o espías de red que capturan eventos en tiempo real. Estos componentes envían los logs al SIEM. Luego existe una capa de procesamiento/transporte, que puede incluir colas o brokers de mensajería, donde los eventos se transmiten al motor central. El módulo de normalización convierte todos los formatos distintos a un esquema unificado. A continuación, el motor de correlación y análisis aplica reglas lógicas y analíticas al flujo de eventos, detectando amenazas complejas. Simultáneamente, los datos se almacenan en una base

indexada (por ejemplo, bases NoSQL o motores de búsqueda de texto completo) que permite consultas rápidas sobre los historiales de eventos (IBM, 2023).

Finalmente, la capa de presentación ofrece el dashboard central y mecanismos de notificación (emails, SIEM interno, etc.) para que los analistas supervisen alertas y respondan a incidentes en tiempo real. Cada módulo recolección, procesamiento, almacenamiento, visualización cumple tareas definidas. Por ejemplo, (Sheeraz, M., et al, 2024) destacan en la literatura la importancia de una arquitectura modular bien definida, donde cada componente (ingesta, normalización, correlación, archivado, etc.) opera de forma independiente. De este modo, la solución puede escalar según demanda (más agentes, más nodos de almacenamiento) y adaptarse a futuros requerimientos.

2.5 Comparativa de soluciones SIEM open source vs comerciales

Las soluciones SIEM se pueden clasificar principalmente en dos grandes grupos: de código abierto (open source) y comerciales. Cada tipo presenta ventajas y desventajas que dependen del contexto organizacional, los recursos disponibles y los objetivos de seguridad.

- Costo y licenciamiento: Las soluciones open source como Wazuh, OSSIM o Graylog eliminan costos de licencia iniciales, lo que puede parecer atractivo para organizaciones con presupuesto limitado. Sin embargo, según Kaspersky Lab., (2025) este ahorro se ve contrarrestado por la necesidad de contar con personal altamente especializado para implementar, configurar y mantener la plataforma. Por lo tanto, el costo total de propiedad (TCO) puede ser alto debido a la inversión en recursos humanos. En cambio, los SIEM comerciales (Splunk, IBM QRadar, ArcSight) tienen costos de licencia y mantenimiento, pero incluyen soporte técnico profesional y actualizaciones automáticas.

- Implementación y despliegue: Los SIEM comerciales suelen ofrecer despliegues más rápidos, ya que incluyen configuraciones y reglas predefinidas que aceleran la puesta en marcha. En promedio, un SIEM comercial puede implementarse correctamente en alrededor de seis meses, mientras que una solución open source puede tardar el doble o más debido a la necesidad de personalizar reglas y adaptar la arquitectura.
- Actualizaciones y contenido: Las suites comerciales gestionan manualmente las reglas, listas negras y motores de correlación, permitiendo detectar nuevas amenazas más rápidamente. En contraste, el SIEM open source depende de la comunidad o equipos internos para actualizar las reglas, lo que puede crear brechas temporales y aumentar falsos positivos si no se mantiene un expertise constante.
- Cumplimiento y reportes: Los SIEM comerciales incluyen plantillas preconfiguradas para normativas como PCI-DSS, GDPR, HIPAA, facilitando auditorías y generación de reportes automáticos. Los SIEM open source requieren que la organización desarrolle estas funcionalidades, lo cual consume recursos adicionales.
- Flexibilidad y dependencia: El software open source ofrece gran flexibilidad para personalizar y extender funcionalidades, evitando depender del roadmap de un proveedor. Sin embargo, esta ventaja puede generar dependencia en el talento interno, pues la salida de personal clave puede afectar el mantenimiento del sistema. Por otro lado, los SIEM comerciales implican dependencia del proveedor para soporte y actualizaciones, aunque garantizan continuidad operativa.
- Escalabilidad y soporte: Ambas soluciones deben escalar su infraestructura según el volumen de datos, pero las comerciales suelen tener arquitecturas más probadas y soporte para dimensionamiento óptimo, mientras que las open source corren el riesgo de configuraciones subóptimas, aumentando costos operativos.

En síntesis, un SIEM open source es adecuado para organizaciones con equipos técnicos fuertes y necesidades de personalización específicas, mientras que las soluciones comerciales favorecen a organizaciones que buscan rapidez, soporte profesional y cumplimiento regulatorio simplificado.

2.6 La plataforma Wazuh

Aunque existen variadas implementaciones, la arquitectura típica de un SIEM es modular. En la capa inicial se ubican los recolectores de datos: agentes instalados en los endpoints, conectores a APIs, sistemas de syslog o espías de red que capturan eventos en tiempo real. Estos componentes envían los logs al SIEM. Luego existe una capa de procesamiento/transporte, que puede incluir colas o brokers de mensajería, donde los eventos se transmiten al motor central. El módulo de normalización convierte todos los formatos distintos a un esquema unificado. A continuación, el motor de correlación y análisis aplica reglas lógicas y analíticas al flujo de eventos, detectando amenazas complejas. Simultáneamente, los datos se almacenan en una base indexada (por ejemplo, bases NoSQL o motores de búsqueda de texto completo) que permite consultas rápidas sobre los historiales de eventos (IBM, 2023). Finalmente, la capa de presentación ofrece el dashboard central y mecanismos de notificación como envio de correos electrónicos para que los analistas supervisen alertas y respondan a incidentes en tiempo real.

2.7 Los componentes de Wazuh

La arquitectura de Wazuh se compone de varios módulos clave que trabajan de forma conjunta brindando una solución SIEM completa. El agente Wazuh es ligero, multiplataforma y se instala en los endpoints para realizar monitoreo local, recolectando eventos que luego son enviados al servidor central. El servidor Wazuh (manager) aplica reglas de detección y correlación sobre los datos recibidos, generando alertas de seguridad y siendo capaz de coordinar miles de agentes simultáneamente. El indexador Wazuh, basado en Elasticsearch, almacena estas alertas en

índices optimizados, permitiendo búsquedas rápidas y análisis histórico eficiente. Por su parte, el dashboard Wazuh, basado en Kibana, proporciona una interfaz visual con paneles interactivos que facilitan el análisis de amenazas y la gestión centralizada del sistema. Además, Wazuh incorpora capacidades de monitoreo agentless, diseñadas para aquellos dispositivos donde no es viable instalar agentes, recolectando logs a través de protocolos como Syslog, SSH o APIs, ampliando así su alcance en entornos híbridos (Wazuh I. , Getting Started – Components. Wazuh Documentation., 2025).

2.8 Integración con elasticsearch

Wazuh se integra nativamente con Elastic Stack (Elasticsearch, Logstash, Kibana). El servidor Wazuh genera alertas en formato JSON que son procesadas por Logstash y enviadas a Elasticsearch para su indexación. Kibana provee dashboards para visualizar alertas, métricas y tendencias, facilitando la correlación avanzada y búsqueda interactiva en grandes volúmenes de datos (Wazuh I. , Elastic Stack integration. Wazuh Documentation. , 2024) Esta integración permite una solución SIEM completa, escalable y con capacidad analítica avanzada.

2.9 Herramientas de simulación de ataques

Para validar y evaluar la efectividad de un SIEM, se utilizan herramientas de simulación de ciberataques, también conocidas como frameworks de red teaming. Estas herramientas reproducen técnicas y tácticas de adversarios reales para probar la detección y respuesta de la plataforma.

2.9.1 Metasploit

Metasploit es un framework open source para pruebas de penetración, ampliamente utilizado para explotar vulnerabilidades en redes, sistemas y aplicaciones. Proporciona una extensa biblioteca de “*exploits*”, “*payloads*” y herramientas auxiliares que permiten simular ataques

reales (Vectra AI, 2024). Es multiplataforma y se integra con distribuciones de seguridad como Kali Linux. Su uso permite verificar que el SIEM detecte correctamente eventos maliciosos generados por ataques conocidos, como ransomware o inyección de código.

2.9.2 Atomic red team

Atomic Red Team (ART) es un proyecto open source mantenido por Red Canary que ofrece una biblioteca de pruebas atómicas alineadas con el framework MITRE ATT&CK. Cada prueba es un "átomo" que simula una técnica específica de ataque (por ejemplo, volcado de credenciales o movimiento lateral) de forma sencilla y "*ART"* facilita la validación de detecciones SIEM mediante la generación controlada de actividad maliciosa específica, actualizándose continuamente con nuevas tácticas reflejadas en MITRE ATT&CK (Red-Orbita, 2022).

2.9.3 MITRE Caldera

MITRE Caldera es una plataforma open source para la emulación avanzada de adversarios, diseñada para automatizar campañas de red teaming y evaluar el enfoque de seguridad. Basada en el marco de trabajo MITRE ATT&CK, Caldera ejecuta ataques simulados que replican tácticas y técnicas reales, usando un agente "*Sandcat*" como se detalló anteriormente para interactuar con los sistemas comprometidos. Su motor de decisiones permite adaptar la campaña en función del estado de la red, automatizando cadenas complejas de ataque.

Se decidió usar exclusivamente MITRE Caldera en este proyecto por su capacidad para simular escenarios realistas y completos, desde reconocimiento hasta exfiltración, sin intervención manual constante. Como afirma (MITRE, Caldera, 2023), Caldera permite "ver la red a través de los ojos de un adversario APT bajo demanda", maximizando la detección de intrusiones avanzadas y validando la respuesta del SIEM Wazuh ante ataques sofisticados.

2.9.4 Monitoreo con agente y sin agente

Wazuh ofrece dos modos complementarios para la recolección de datos en entornos de seguridad: "*agent-based*" y "*agentless*" es decir, basado en agente y sin agente. En el modo agent-based, se instalan agentes directamente en los endpoints (como servidores, estaciones de trabajo o contenedores), los cuales envían datos de manera continua al servidor Wazuh, habilitando funciones avanzadas como detección de rootkits, escaneo de integridad y respuesta activa ante amenazas (Wazuh I. , Getting Started – Components. Wazuh Documentation., 2025).

Por otro lado, el modo agentless está diseñado para dispositivos que no permiten la instalación de software adicional, como algunos firewalls, routers o servidores heredados; en estos casos, Wazuh captura los logs utilizando protocolos estándar como Syslog, SSH o APIs, y los procesa del mismo modo que los datos provenientes de agentes, aunque con menor profundidad analítica (Wazuh I. , Getting Started – Components. Wazuh Documentation., 2025). Este enfoque dual amplía significativamente la cobertura del sistema, permitiendo una supervisión eficaz tanto en infraestructuras modernas como en entornos heterogéneos o con restricciones técnicas.

2.10. Normativas y estándares aplicables

En el ámbito de la ciberseguridad y la gestión de eventos, es fundamental considerar el cumplimiento de normas internacionales y nacionales. ISO/IEC 27001:2022 constituye el estándar global para establecer, implementar, mantener y mejorar un Sistema de Gestión de Seguridad de la Información (SGSI) mediante un enfoque de evaluación continua de riesgos y controles integrales (Organización Internacional de Normalización y Electrotécnica [ISO/IEC], 2022).

En Ecuador, la Ley Orgánica de Protección de Datos Personales (LOPDP), vigente desde mayo de 2021, garantiza el derecho constitucional a la privacidad, estableciendo principios como licitud, finalidad, proporcionalidad y seguridad en el tratamiento de datos personales (Protección de Datos Personales - Tecnologia, 2025). Ambas normativas exigen registrar eventos, definir controles de acceso, capacidad de auditoría e implementación de medidas técnicas y organizacionales, siendo un SIEM como Wazuh una herramienta idónea para apoyar su cumplimiento.

2.10.1 ISO/IEC 27001

La norma ISO/IEC 27001 es el referente internacional para los Sistemas de Gestión de la Seguridad de la Información (SGSI) (ISO, 2022). Publicada originalmente en 2005 y actualizada en 2013 y 2022, establece los requisitos para implementar, mantener y mejorar continuamente un SGSI basado en el ciclo de mejora continua "*Plan-Do-Check-Act*" (PDCA) (ISO, 2022; IBM, 2023). Su objetivo es ayudar a las organizaciones a identificar riesgos relacionados con la información, seleccionar controles adecuados del Anexo A y establecer un marco de gestión que garantice protección, integridad y disponibilidad de los activos. Entre los beneficios principales se incluyen la reducción de incidentes de seguridad, mejora en la confianza de las partes interesadas y cumplimiento con regulaciones nacionales e internacionales. Una certificación ISO/IEC 27001 agrade a clientes y auditores, y fortalece la postura competitiva de empresas que manejan información sensible.

2.10.2. Ley orgánica de protección de datos personales del ecuador

La Ley organica de protección de datos personales del ecuador o por sus siglas LOPDP, promulgada el 26 de mayo de 2021, regula el tratamiento de datos personales en Ecuador con el propósito de garantizar el derecho constitucional a la privacidad y al acceso a la información. Aplica tanto en soporte físico como digital y establece obligaciones claras para responsables y

encargados del tratamiento. Entre los principios fundamentales se destacan la legalidad, finalidad, proporcionalidad, calidad, seguridad y responsabilidad proactiva. La norma también regula derechos de los titulares (acceso, rectificación, oposición, portabilidad, borrado) y define la creación de la Autoridad de Protección de Datos personales, junto a su registro público y régimen sancionatorio por incumplimiento.

3. CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DEL SISTEMA SIEM CON WAZUH EN ECOILPET

3.1 Ingeniería de software del sistema SIEM

3.1.2 Metodología de Desarrollo

El proyecto se desarrolló empleando una metodología ágil adaptada, combinando "*Scrum"* con prácticas "*DevSecOps*" para garantizar la entrega incremental de funcionalidades. Se organizó en cinco "*sprints*" de dos semanas cada uno, donde el primer sprint se dedicó al análisis de requisitos y configuración del entorno base. Los "*sprints*" 2 y 3 implementaron el núcleo de Wazuh y su integración con Elastic Stack, mientras que los "*sprints*" finales se enfocaron en personalizar reglas de seguridad y "*dashboards*". Cada iteración incluyó pruebas automatizadas con MITRE Caldera y revisiones con el equipo de seguridad de ECOILPET, permitiendo ajustar continuamente los componentes según las necesidades operativas reales detectadas.

3.1.3 Arquitectura del Sistema

El sistema sigue una arquitectura en tres capas claramente definidas. La capa de recolección integra tanto agentes Wazuh instalados en servidores Windows y Linux como dispositivos de red monitoreados mediante "*Syslog*". La capa de procesamiento, alojada en un servidor Ubuntu, incluye el Wazuh Manager con reglas personalizadas para detectar amenazas específicas del entorno ECOILPET. Finalmente, la capa de visualización utiliza Elasticsearch para almacenamiento y Kibana para presentación de datos, con "*dashboards*" adaptados a diferentes perfiles de usuario.

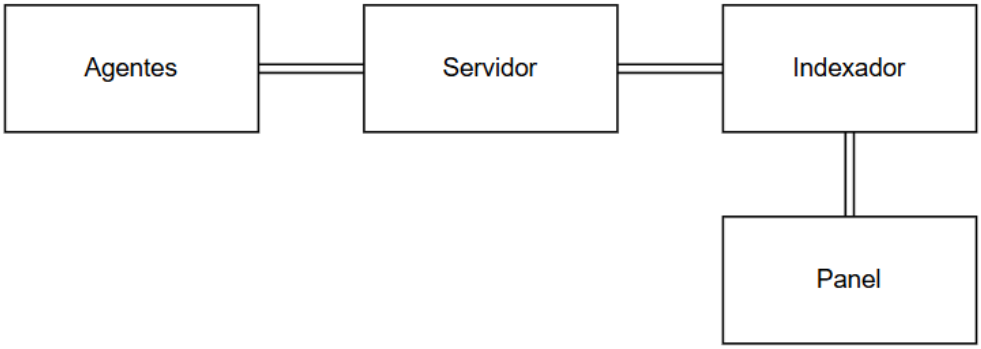


Figura 1. Arquitectura SIEM.

3.1.4 Casos de Uso Principales

El sistema implementa siete casos de uso clave que cubren todo el ciclo de gestión de seguridad. Estos incluyen la ingesta y normalización de "*logs*", correlación de eventos, investigación de alertas y la administración del sistema. Los administradores pueden gestionar reglas y usuarios, mientras los analistas investigan alertas mediante herramientas de correlación avanzada y la monitorean el estado general del sistema, y los dispositivos envían datos de forma automatizada.

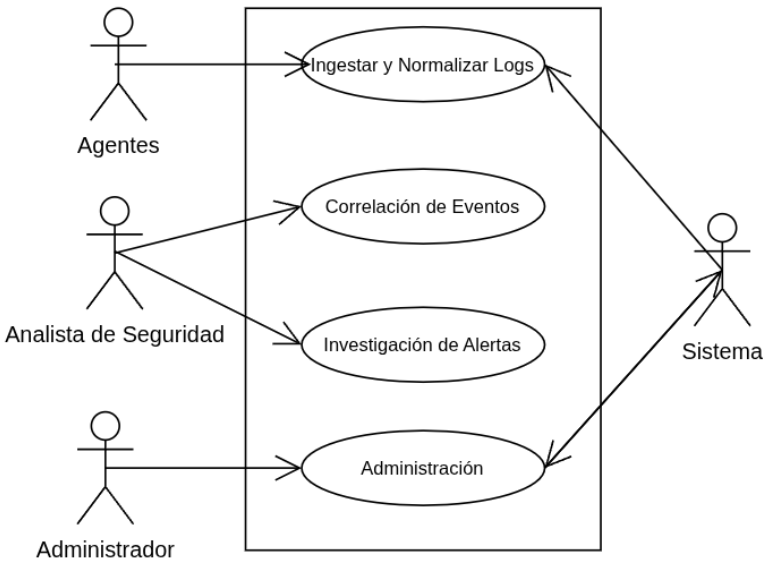


Figura 2. Diagrama de casos de uso.

3.1.5 Modelado de Datos

La estructura de datos se diseñó para optimizar el almacenamiento y consulta en Elasticsearch. Cada alerta se almacena como documento JSON con campos estandarizados que incluyen "*timestamp*" es decir la fecha y hora exacta, nivel de severidad, dispositivo origen y detalles técnicos del evento. Este modelo permite búsquedas complejas y agregaciones rápidas para la generación de reportes.

3.2 Análisis de la infraestructura actual de ECOILPET S.A.

La red de ECOILPET está compuesta por una LAN privada con subred 192.168.x.x que conecta estaciones de trabajo Windows y servidores Linux. La conectividad cableada se basa en switches Cisco administrables, mientras que la red inalámbrica es gestionada completamente en la nube mediante Cisco Meraki, sin necesidad de un controlador físico local.

La infraestructura de seguridad incluye un firewall Palo Alto como dispositivo de perímetro. Tanto los switches Cisco como los dispositivos Meraki y el firewall Palo Alto permiten visibilidad avanzada del tráfico y exportación de "*logs"* vía "*Syslog*", lo que posibilita su integración con el sistema SIEM.

En este entorno híbrido, los agentes Wazuh se instalaron únicamente en un subconjunto de "*endpoints*" Linux y Windows como entorno de prueba funcional, mientras que los dispositivos de red se integraron en modalidad "*agentless*" utilizando "*Syslog"*.

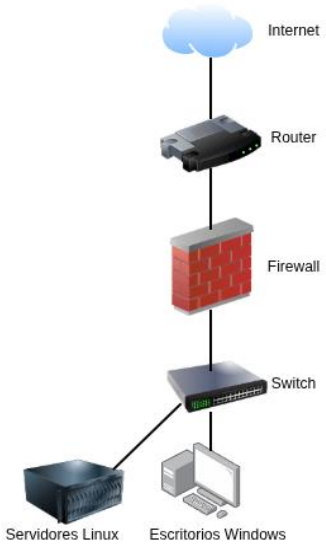


Figura 3. Infraestructura tecnológica de la empresa ECOILPET S.A.

3.3 Requerimientos técnicos

3.4 Hardware y conectividad

El diseño ideal de un sistema SIEM requiere una infraestructura robusta. Según la documentación oficial de Wazuh, el servidor que ejecuta el Wazuh Manager debe contar con al menos 2 CPU, 4 GB de RAM y 10 GB de almacenamiento. Para entornos empresariales como ECOILPET, se recomienda una configuración con 8 CPU, 16 GB de RAM y almacenamiento SSD de al menos 250 GB, para retención de eventos por 90 días. No obstante, esta implementación corresponde a una prueba piloto, realizada sobre una máquina virtual con 1 CPU, 4 GB de RAM y 100 GB de disco. Esta configuración fue suficiente para validar todas las funcionalidades del sistema Wazuh en una escala controlada. La instalación se aplicó a algunos hosts internos, junto con la integración de dispositivos de red. Esta validación constituye la base para una futura ampliación a toda la infraestructura.

3.5 Software

El sistema SIEM está compuesto por tecnologías de código abierto:

- Ubuntu Server 22.04 LTS en el nodo.
- Wazuh Manager: módulo central para recibir eventos y generar alertas.
- Wazuh Indexer (Elasticsearch): almacenamiento indexado.
- Wazuh Dashboard (Kibana personalizado): visualización y análisis.
- Filebeat: reenvío de logs del Manager al Indexer.

Se instalaron agentes Wazuh en sistemas Linux y Windows, mientras que los dispositivos de red (Meraki, "*switch"* Cisco, "*firewall"* Palo Alto) fueron integrados mediante "*Syslog"*, sin necesidad de agentes.

El despliegue del "*stack*" completo se realizó mediante el script oficial de instalación automatizada de Wazuh, que simplifica y garantiza una instalación rápida y segura.

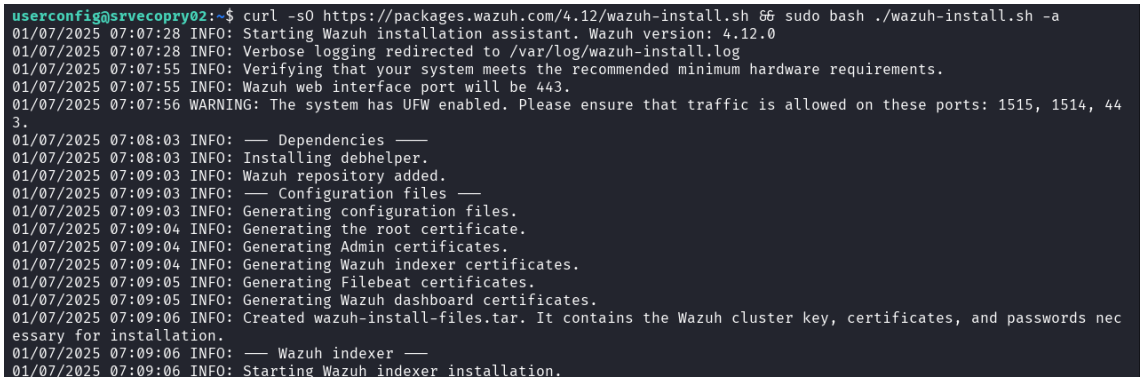


Figura 4. Instalación de Wazuh mediante script oficial.

3.6 Arquitectura propuesta por el SIEM

La arquitectura de seguridad implementada en ECOILPET se basa en la centralización y análisis de la información de seguridad a través del sistema SIEM Wazuh. En este esquema, el servidor Wazuh actúa como el núcleo del SIEM, donde reside toda la lógica de correlación de eventos, análisis de comportamiento y generación de alertas. Para que este servidor funcione de manera óptima, es fundamental que reciba los datos de seguridad de toda la infraestructura. Esta alimentación de información se logra a través de dos mecanismos principales: los agentes de Wazuh y la integración con "*Syslog"*. Los agentes, instalados en los hosts (servidores y estaciones de trabajo), monitorean la actividad local en tiempo real y envían sus logs al servidor. Paralelamente, los dispositivos de red ("*firewall", "router", "switch*") que no pueden tener agentes instalados, envían sus propios "*logs"* al servidor Wazuh utilizando el protocolo estándar de red, "*Syslog*". De esta manera, el servidor centralizado de Wazuh recibe un flujo constante de logs de toda la infraestructura, permitiendo una visión integral para la detección y respuesta a incidentes de seguridad.

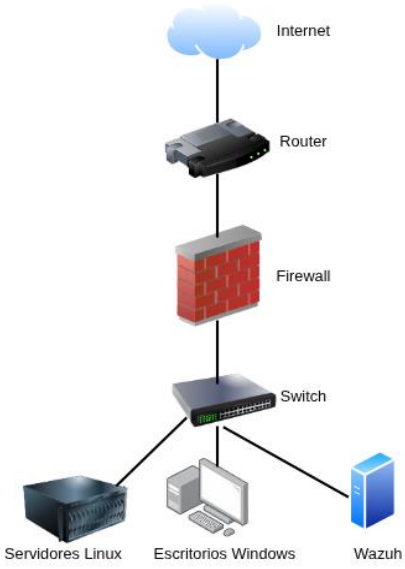


Figura 5. Arquitectura del sistema SIEM implementado en ECOILPET S.A.

3.7 Instalación y configuración de Wazuh

3.7.1 Instalación del manager

La instalación del Wazuh Manager se realiza sobre Ubuntu Server. Se descargan los "*scripts*" oficiales y se ejecutan comandos de instalación automatizada, que configuran también "*Filebeat*" y el servicio de "*API REST*". El servicio se inicia con "*systemd"* y se valida su estado mediante comandos como "systemctl status wazuh-manager".

3.7.2 Configuración de agentes y agentless

Los agentes de Wazuh se despliegan en cada "*endpoint*" para recolectar eventos y enviarlos al servidor central. En sistemas Linux, el proceso comienza instalando el agente como se observa en la figura 6. Seguido de esto, se procede a reiniciar el servicio como se ve en la figura 7.

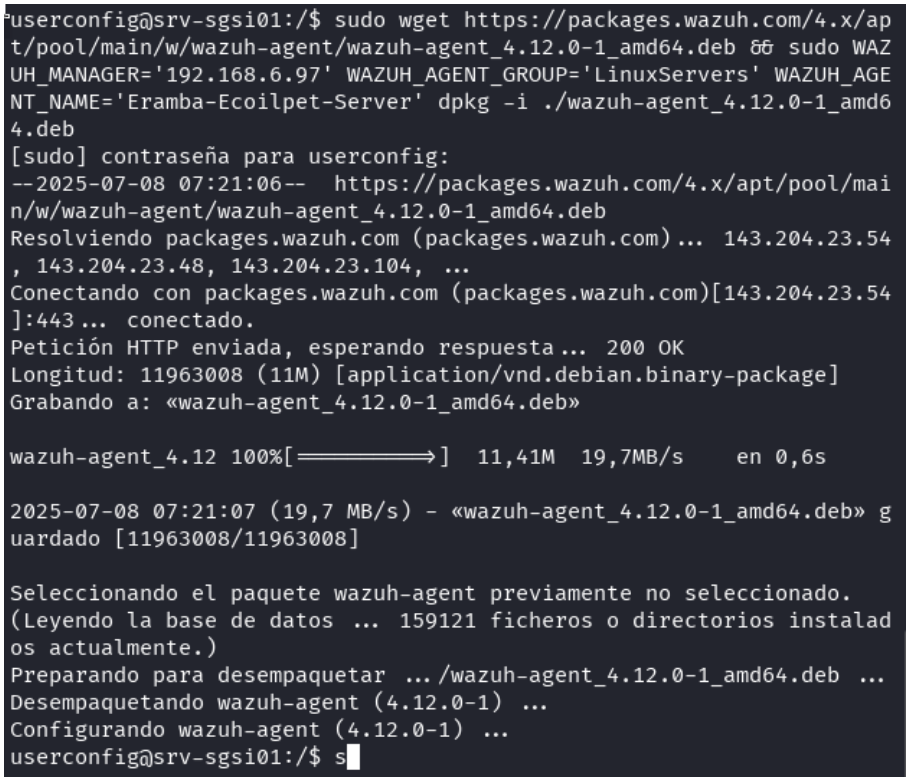


Figura 6. Instalación agente servidor Linux.

```
userconfig@srv-sgsi01:/$ sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
Created symlink /etc/systemd/system/multi-user.target.wants/wazuh-ag
ent.service → /lib/systemd/system/wazuh-agent.service.
```

Figura 7. Inicio del agente Linux.

En entornos Windows, se utiliza un instalador MSI provisto por Wazuh, como wazuh-agent-4.12.0-1.msi. Este puede ejecutarse en modo silencioso con parámetros como:

```
msiexec /i wazuh-agent-4.12.0-1.msi /quiet WAZUH_MANAGER="IP"
```

Esto permite registrar automáticamente el agente con el servidor. Una vez desplegado, cada agente debe ser autorizado desde el servidor, ya sea de forma automática o mediante una clave de registro, garantizando así la autenticación de los dispositivos conectados.

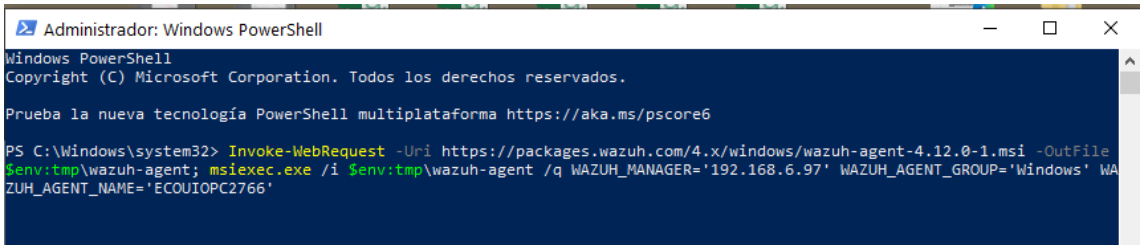


Figura 8. Instalación de agentes en endpoints de Windows.1

Para Windows, se debe instalar e integrar "*Sysmon*" ("System Monitor") como parte del monitoreo avanzado del sistema operativo. "*Sysmon*" es una herramienta de Microsoft que registra eventos detallados como creación de procesos, conexiones de red, carga de controladores, entre otros. Para su correcta integración con Wazuh, primero se instala "*Sysmon"*

en los "*endpoints"* Windows utilizando una configuración personalizada definida en un archivo llamado sysconfig.xml (Ver Anexo E).

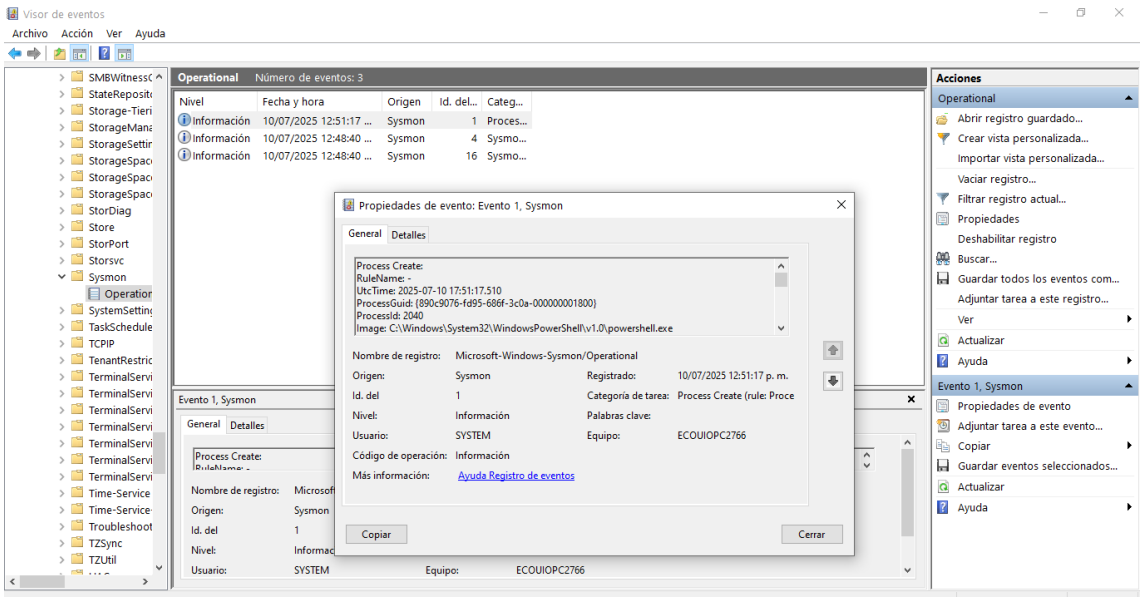


Figura 9. Sysmon en Windows detectando Powershell.

El archivo sysconfig.xml específica qué eventos deben ser monitoreados y con qué nivel de detalle, permitiendo filtrar información irrelevante y enfocarse en actividades potencialmente maliciosas. Además, en este mismo archivo se pueden crear exclusiones directas, por ejemplo, para evitar registrar procesos confiables o rutas específicas que no representen riesgo, reduciendo así el ruido en los logs. Una vez instalado y configurado, los eventos generados por Sysmon son capturados por el agente de Wazuh y analizados mediante reglas específicas, facilitando la detección de amenazas, ejecución de comandos sospechosos o movimientos laterales dentro de la red. Esta integración fortalece significativamente las capacidades de seguridad en entornos Windows.

```
<RuleGroup name="" groupRelation="or">
▼<ProcessCreate onmatch="exclude">
   <!--  SECTION EXclude custom (win,sophos) -->
   <Image condition="is">C:\Windows\system32\cleanmgr.exe</Image>
   <Image condition="is">C:\Windows\System32\sdiagnhost.exe</Image>
   <Image condition="is">C:\Windows\System32\dmclient.exe</Image>
   <Image condition="is">C:\Windows\servicing\TrustedInstaller.exe</Image>
   <Image condition="is">C:\Program Files (x86)\Microsoft\EdgeUpdate\MicrosoftEdgeUpdate.exe</Image>
   <Image condition="is">C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe</Image>
   <Image condition="is">C:\Program Files\WinRAR\WinRAR.exe</Image>
   <Image condition="is">C:\Program Files\Just Great Software\EditPad Lite 8\EditPadLite8.exe</Image>
   <Image condition="is">C:\Windows\system32\cleanmgr.exe</Image>
   <Image condition="is">C:\Windows\System32\sdiagnhost.exe</Image>
   <Image condition="is">C:\Windows\System32\dmclient.exe</Image>
   <Image condition="is">C:\Windows\servicing\TrustedInstaller.exe</Image>
   <Image condition="is">C:\Windows\WinSxS\amd64_microsoft-windows-servicingstack_31bf3856ad364e35_10.0.19041.6088_none_7e3c
   <Image condition="is">C:\Program Files (x86)\Microsoft\EdgeUpdate\MicrosoftEdgeUpdate.exe</Image>
   <Image condition="is">C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe</Image>
   <Image condition="is">C:\Users\PASANTE_SI_2\AppData\Local\Microsoft\OneDrive\25.122.0624.0004\FileCoAuth.exe</Image>
   <Image condition="is">C:\Program Files\WinRAR\WinRAR.exe</Image>
   <Image condition="is">C:\Program Files\Just Great Software\EditPad Lite 8\EditPadLite8.exe</Image>
   <Image condition="is">C:\Program Files\Sophos\Sophos File Scanner\SophosFileScanner.exe</Image>
   <Image condition="is">C:\Program Files\Sophos\Endpoint Defense\SophosCleanup.exe</Image>
   <Image condition="is">C:\Program Files\Sophos\Endpoint Defense\SophosSafestore.exe</Image>
   <Image condition="is">C:\Program Files\Sophos\AutoUpdate\SophosLaunchUpdate.exe</Image>
   <Image condition="is">C:\ProgramData\Sophos\AutoUpdate\Cache\sophos_autoupdate1.dir\su-repair.exe</Image>
   <Image condition="is">C:\ProgramData\Sophos\AutoUpdate\Cache\sophos_autoupdate1.dir\SophosUpdate.exe</Image>
   <Image condition="is">C:\Windows\System32\consent.exe</Image>
   <Image condition="is">C:\Program Files\RUXIM\PLUGScheduler.exe</Image>
   <!-- SECTION: Microsoft Windows -->
   <CommandLine condition="begin with"> "C:\Windows\system32\wermgr.exe" "-queuereporting_svc" </CommandLine>
```

Figura 10. Archivo de configuración sysconfig.xml.

Finalmente, el archivo ossec.conf del agente windows contiene toda la configuración como se muestra a continuación.

```
<!--
  Wazuh - Agent - Default configuration for Windows
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>

  <client>
    <server>
      <address>192.168.6.97</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>windows, windows10</config-profile>
    <crypto_method>aes</crypto_method>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <enrollment>
      <enabled>yes</enabled>
      <agent_name>Wazuh-Agent-ECOUIOPC2766</agent_name>
      <groups>Ecoilpet</groups>
    </enrollment>
  </client>


  <!-- Agent buffer options -->
  <client_buffer>
    <disabled>no</disabled>
    <queue_size>5000</queue_size>
    <events_per_second>500</events_per_second>
  </client_buffer>
```

Figura 11. Archivo de configuración ossec.conf del endpoint Windows.

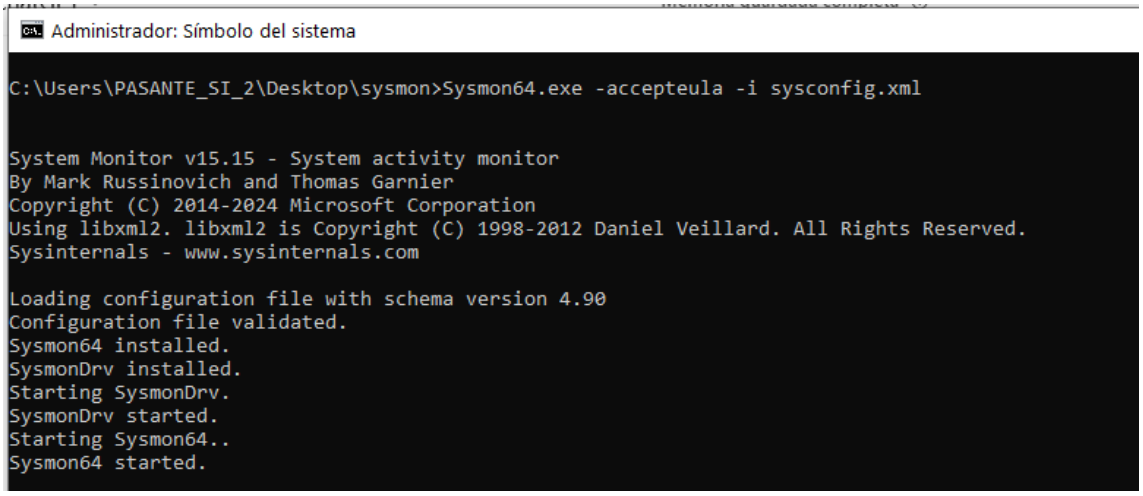


Figura 12. Configuración de Sysmon en Windows.

Para los dispositivos que no permiten la instalación de software, como el switch Cisco, la infraestructura Meraki y el "*firewall"* Palo Alto, se implementó integración exclusivamente mediante "*Syslog"*.

Este método permite que los dispositivos envíen "*logs"* al Wazuh Manager sin necesidad de software adicional. Para ello:

- Se activó la sección <remote> en el archivo ossec.conf del servidor.
- Se habilitó el puerto TCP/514 para la recepción de mensajes.

En cada dispositivo (Meraki, "*switch"* Cisco, "firewall" Palo Alto), se configuró la IP del servidor Wazuh como destino de "*logs" "Syslog*".

Todos los logs recibidos fueron analizados por el motor de reglas de Wazuh y visualizados en tiempo real en los "*dashboards*" personalizados, extendiendo así el monitoreo a los dispositivos de red clave de la organización.

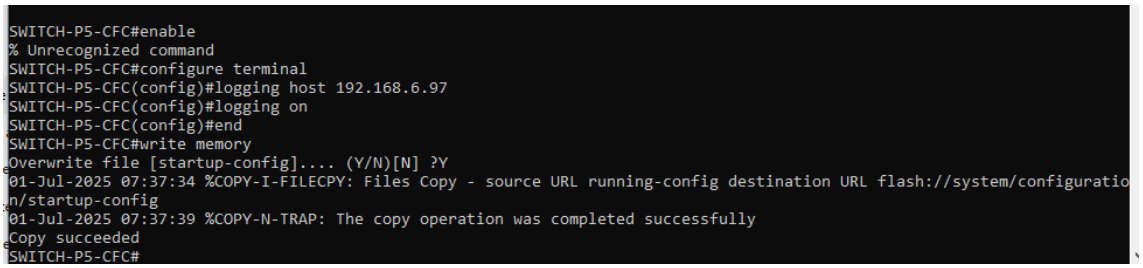


Figura 13. Configuración Syslog en switch cisco.

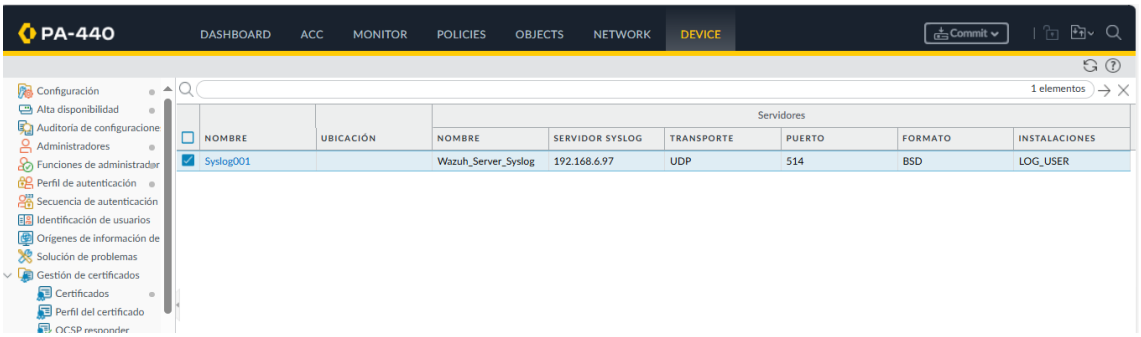


Figura 14. Configuración syslog en firewall paloalto.

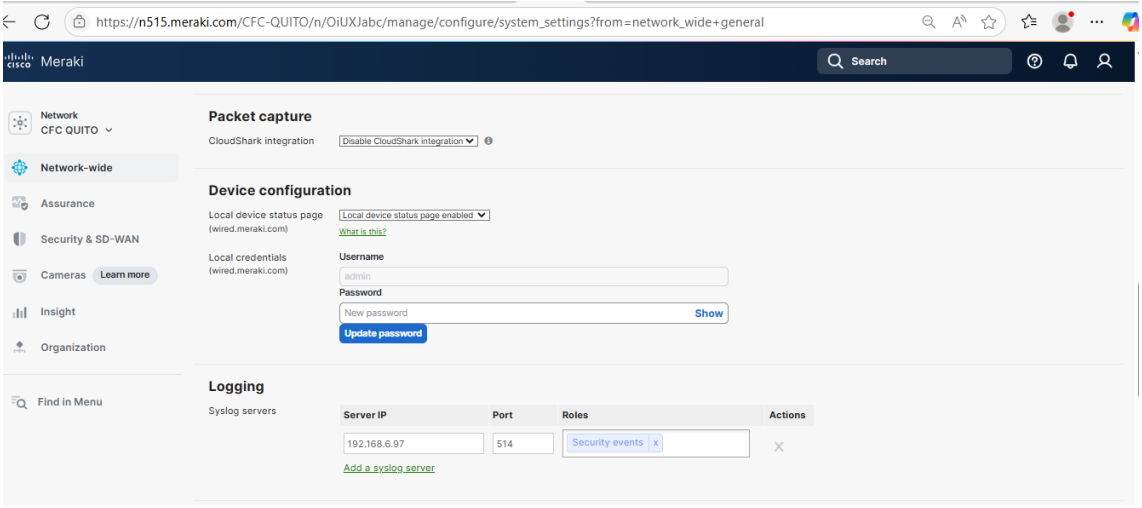


Figura 15. Configuración Syslog en cisco Meraki.

El archivo ossec.conf del Wazuh Manager se configura para centralizar la gestión de agentes, la recolección de logs, y la ejecución de respuestas automáticas ante eventos críticos. En esta configuración, se define el bloque <remote> donde se habilita la comunicación con los agentes utilizando el protocolo UDP o TCP, especificando los "*endpoints"* del servidor mediante <connection> y <port>, generalmente en el puerto 1514 o 1515. Dentro de <active-response>, se configuran respuestas automáticas como el bloqueo de IPs maliciosas mediante configuraciones como "*firewalldrop"*, asociando acciones a reglas específicas y limitando su ejecución por origen o agente. Adicionalmente, se incluyen bloques como <syscheck> para la monitorización de integridad de archivos, <logcollector> para la recolección de logs del sistema y aplicaciones, y <ruleset> donde se define la ubicación de las reglas personalizadas, incluyendo las locales (local_rules.xml) y las exclusiones (ignore.xml). Esta estructura garantiza un comportamiento automatizado y adaptable del Wazuh Manager ante comportamientos sospechosos o maliciosos dentro de la infraestructura monitorizada.

```
  <remote>
    <connection>secure</connection>
    <port>1514</port>
    <protocol>tcp</protocol>
    <queue_size>131072</queue_size>
  </remote>

<!-- Meraki QUITO CFC syslogs -->

<remote>
  <connection>syslog</connection>
  <port>514</port>
  <protocol>udp</protocol>
  <allowed-ips>10.10.10.2</allowed-ips>
</remote>

<!--SW PISO 5 CFC -->
<remote>
  <connection>syslog</connection>
  <port>514</port>
  <protocol>udp</protocol>
  <allowed-ips>192.168.1.245</allowed-ips>
</remote>

<!-- SW CORE -->
<remote>
  <connection>syslog</connection>
  <port>514</port>
  <protocol>udp</protocol>
  <allowed-ips>192.168.1.243</allowed-ips>
</remote>
```

Figura 16. Archivo ossec.conf de Wazuh manager.

3.7.3 Respuesta activa

Se implementaron respuestas automáticas ante eventos:

- Linux: bloqueo de IPs por múltiples intentos fallidos de SSH mediante firewalld (Figura 17).
- Windows: integración con la "*API*" de VirusTotal para escaneo de archivos y eliminación automática mediante scripts (Figura 18).
- *"Sysmon"*: se utilizó para detectar el uso malicioso de PowerShell y otras actividades sospechosas.

Figura 17. Alerta generada de acceso denegado a switch mediante SSH.

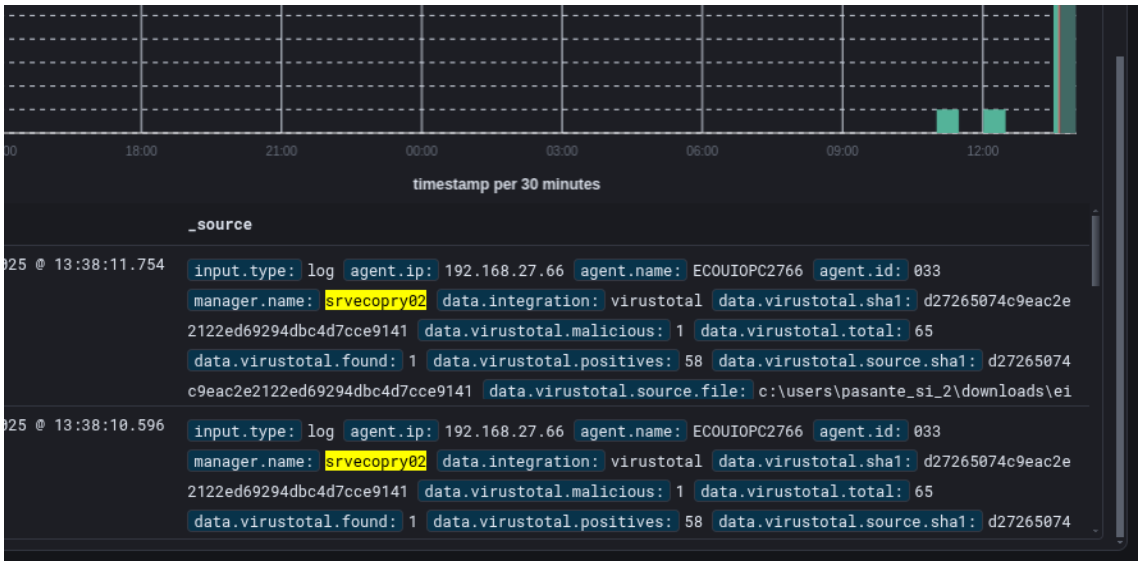


Figura 18. Alerta generada de malware escaneado con virus total.

Figura 19. Respuesta activa bloqueo de ip.

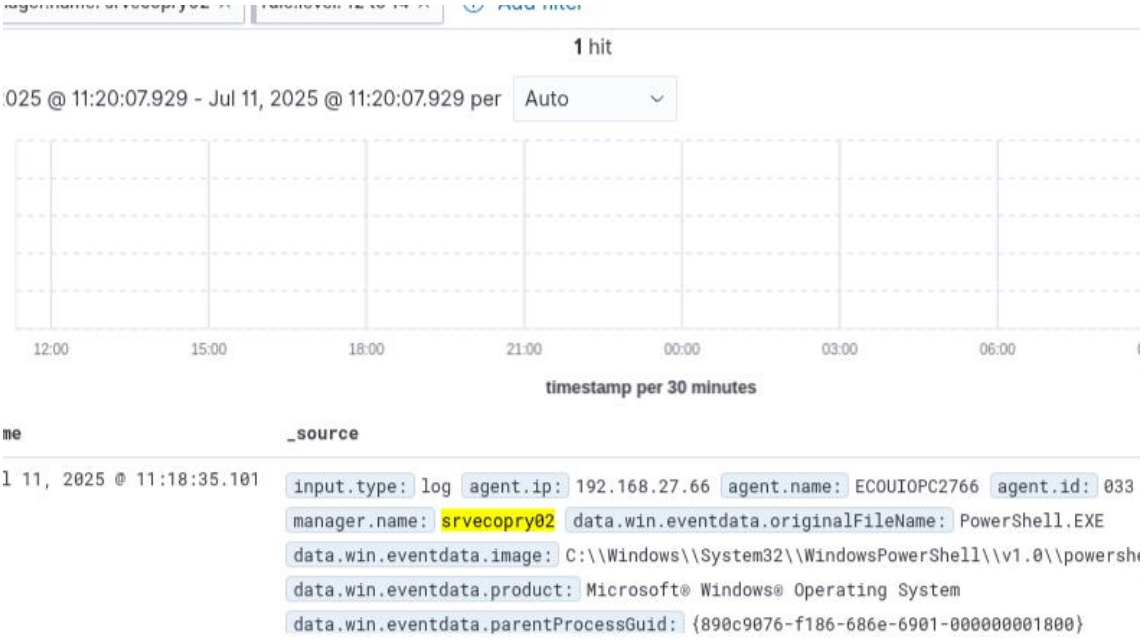


Figura 20. Alerta generada para ejecuccion de powershell.

Para fortalecer las capacidades de respuesta activa en Windows, se implementó un script en Python que interactúa con la "*API*" de Wazuh y VirusTotal para la detección y eliminación automatizada de amenazas. El "*script"*, diseñado para ejecutarse como una respuesta activa de Wazuh, analiza las alertas generadas por el módulo FIM *("File Integrity Monitoring*") al detectar cambios en directorios críticos como C:\Users\<USER>\Downloads. Cuando se identifica un archivo sospechoso, el sistema consulta la API de VirusTotal para verificar su reputación. Si el archivo es marcado como malicioso, el script utiliza funciones seguras de Python (*os.remove()*) para eliminarlo de manera controlada, evitando la manipulación de enlaces simbólicos o rutas malintencionadas. Para garantizar su ejecución en entornos Windows sin dependencias externas, el script se compiló en un ejecutable portable (remove-threat.exe) mediante PyInstaller. Esta integración no solo permite una respuesta inmediata ante amenazas conocidas, sino que también registra cada acción en los "*logs"* de Wazuh para su posterior

auditoría, demostrando una solución eficiente y escalable para la mitigación proactiva de malware.

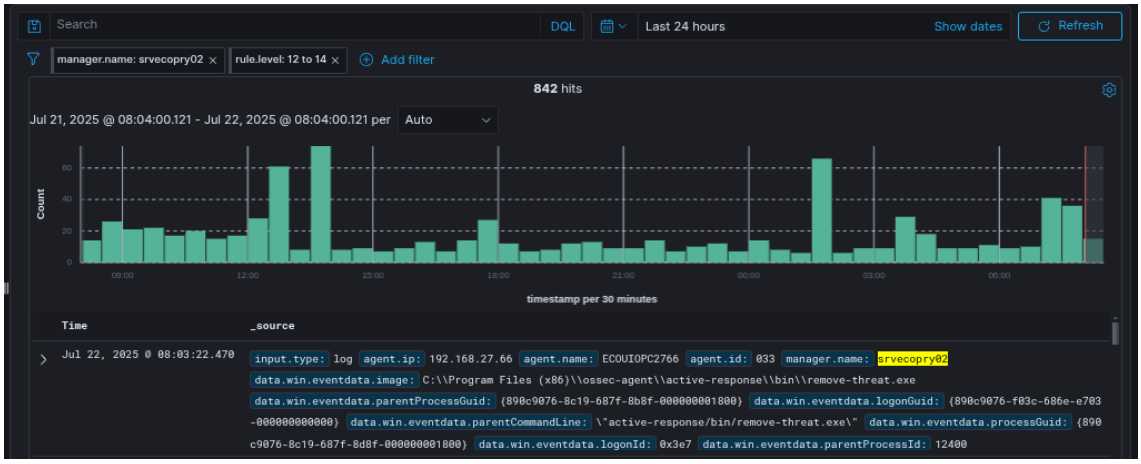


Figura 21. Alerta generada de borrado de malware con el script “remove_threat.py”.

Para llevar a cabo el borrado del archivo malicioso se utilizó el siguiente código en python el cual fue convertido a .exe para su ejecución y se lo situa en la ruta “*C:\Program Files (x86)\ossec-agent\active-response\bin*” del dispositivo windows.

```python
# Copyright (C) 2015-2025, Wazuh Inc.
# All rights reserved.

import os
import sys
import json
import datetime
import stat
import tempfile
import pathlib

if os.name == 'nt':
    LOG_FILE = "C:\\Program Files (x86)\\ossec-agent\\active-response\\active-responses.log"
else:
    LOG_FILE = "/var/ossec/logs/active-responses.log"

ADD_COMMAND = 0
DELETE_COMMAND = 1
CONTINUE_COMMAND = 2
ABORT_COMMAND = 3

OS_SUCCESS = 0
OS_INVALID = -1

class message:
    def __init__(self):
        self.alert = ""
        self.command = 0

def write_debug_file(ar_name, msg):
    with open(LOG_FILE, mode="a") as log_file:
        log_file.write(str(datetime.datetime.now().strftime('%Y/%m/%d %H:%M:%S')) + " " + ar_name + ": " + msg +"\n")

```

Figura 22. Script “remove_threat.py”.

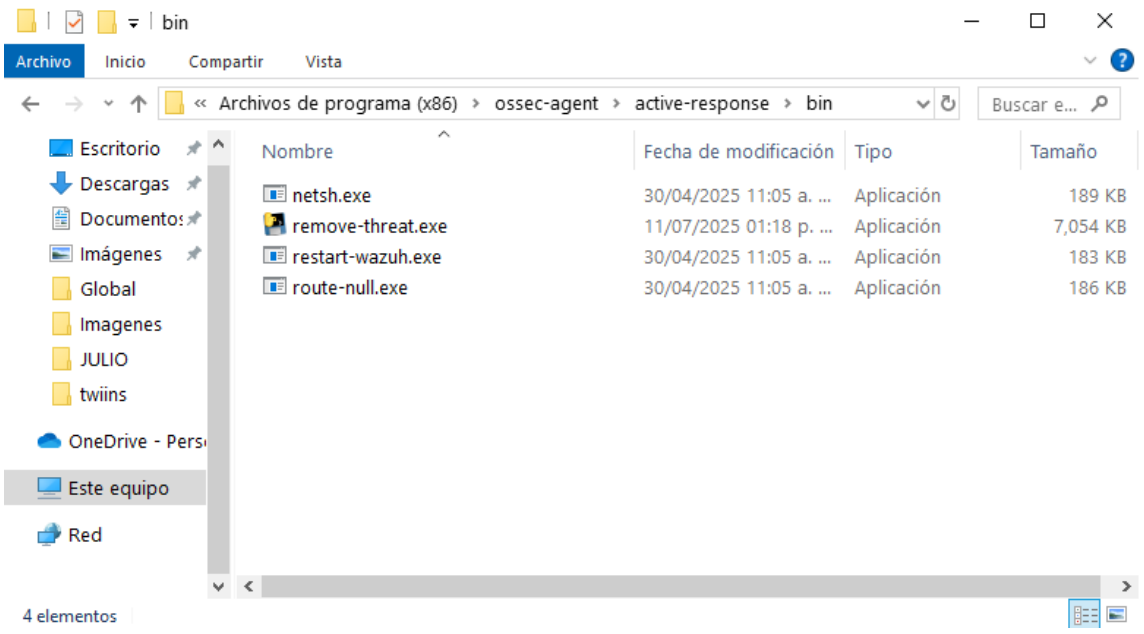


Figura 23. Active response: remove_threat.exe.

3.7.4 Uso de reglas y decodificadores personalizados

Para mejorar la precisión del monitoreo y adaptar las alertas al contexto específico de ECOILPET, se configuraron reglas y decodificadores personalizados dentro del Wazuh Manager. Esto permitió extender las capacidades del sistema para detectar eventos específicos que no estaban contemplados por defecto en las reglas base. En particular, se personalizaron reglas en el archivo:

/var/ossec/etc/rules/local_rules.xml

Estas reglas fueron ajustadas para generar alertas ante comportamientos anómalos detectados en logs de dispositivos switches Cisco, Firewall PaloAlto.

```
<!-- Local rules -->

<!-- IGNORE WAZUH -->
<group name="local,syscheck">
  <rule id="100600" level="0" overwrite="yes">
    <match>netstat listening ports</match>
    <description>Ignorar alertas que contienen netstat listening ports</description>
  </rule>
</group>


<group name="local,syslog,cisco,sshd,">
  <rule id="100001" level="5">
    <if_sid>5716</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>sshd: authentication failed from IP 1.1.1.1.</description>
    <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
  </rule>

<rule id="100200" level="7">
  <decoded_as>cisco-ssh-reject</decoded_as>
  <match>192.168.1.243</match>
  <description>SW-CORE: SSH rechazado en Cisco</description>
  <group>authentication_failed,switch-core,syslog</group>
</rule>

<rule id="100201" level="7">
  <decoded_as>cisco-ssh-reject</decoded_as>
  <match>192.168.1.245</match>
  <description>SW-P5-CFC: SSH rechazado en Cisco</description>
  <group>authentication_failed,switch-p5-cfc,syslog</group>
</rule>
```

Figura 24. Reglas personalizadas implementadas en el Wazuh Manager.

Wazuh ya incluye por defecto una gran cantidad de reglas y decodificadores que cubren la mayoría de las soluciones de seguridad, redes y sistemas más comunes. Sin embargo, es posible crear reglas personalizadas para adaptar el sistema a necesidades específicas o a dispositivos particulares. Por ejemplo, para categorizar las alertas por dispositivo, se creó el archivo local_rules_meraki.xml, el cual contiene únicamente las reglas relacionadas con equipos Cisco Meraki. Dentro de este archivo, se definió una regla con ID 100500 y nivel de severidad 5, que detecta logs provenientes del dispositivo con IP 200.63.217.118. Esta regla utiliza el campo <match> para identificar la IP en los mensajes, y asigna una descripción personalizada junto con grupos como Meraki y Syslog para facilitar su organización y análisis. Una forma rápida y efectiva de crear reglas en Wazuh consiste en observar los logs reales que se reciben, identificar patrones fijos como direcciones IP, nombres de host o cadenas de texto, y reutilizar reglas existentes como plantilla, ajustando solo los campos necesarios. Además, es recomendable agrupar las reglas en archivos separados por tipo de dispositivo o solución (por ejemplo, Paloalto, Fortinet, etc.), lo que permite un mantenimiento más limpio y ordenado del sistema de detección.

Por otro lado, los decodificadores (/var/ossec/etc/decoders/local_decoder.xml) se usaron para interpretar adecuadamente logs personalizados que no seguían formatos estándar, permitiendo que el motor de reglas los pueda procesar de forma efectiva.

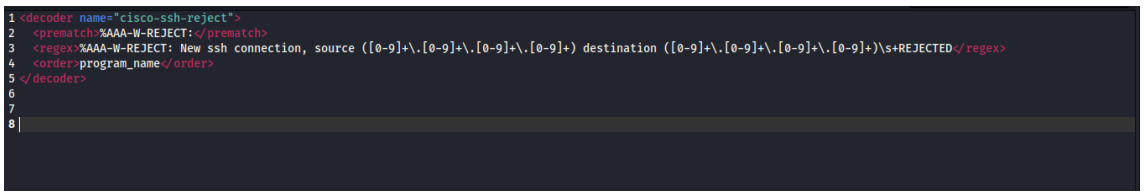


Figura 1. Decodificador personalizado implementadas en Wazuh Manager.

Esta personalización es fundamental en cualquier implementación SIEM madura, ya que permite reducir falsos positivos, aumentar la detección temprana de amenazas y alinear la inteligencia del sistema con los activos más críticos de la organización.

Adicional se configura las reglas a ignorar, eventos específicos del sistema de generación de alertas. Esta funcionalidad es esencial para reducir falsos positivos o eliminar notificaciones relacionadas con comportamientos legítimos que no representan una amenaza. Por ejemplo, en entornos Linux se pueden ignorar alertas del tipo "*netstat listening ports*" o sesiones PAM originadas desde la propia IP del servidor Wazuh. En otros casos, como los eventos generados por el gestor de paquetes dpkg, se pueden desactivar notificaciones relacionadas con paquetes en estado half-configured o installed mediante la herencia de reglas específicas (if_sid).

También se contemplan exclusiones en agentes como Zabbix, donde se ignoran flujos POST legítimos detectados erróneamente como ataques, o en entornos Windows con Sysmon, donde se descartan eventos generados por procesos legítimos como auditpol.exe ejecutado por el wazuh-agent, o actualizaciones del sistema como MicrosoftEdgeUpdate.exe. Estas reglas son agrupadas en bloques <group> con nivel de alerta 0 y pueden incluir condiciones por IP, nombre

de proceso, agente o cadenas de texto específicas. Una vez definidas, las exclusiones entran en efecto tras reiniciar el servicio del Wazuh Manager, optimizando la precisión del sistema de detección.

Figura 25. Exclusiones en reglas con el archivo" ignore.xml".

Las reglas locales personalizadas en Wazuh fueron diseñadas para optimizar la detección de amenazas y reducir falsos positivos en el entorno monitoreado. Estas incluyen la detección proactiva de actividades sospechosas, como intentos de autenticación fallidos desde IPs específicas como el bloqueo de ejecuciones críticas (reglas 255000 y 255001 para *PowerShell* y cmd.exe). Adicionalmente, se priorizó la protección de dispositivos de red, como switches Cisco (reglas 100200), la gestión de cambios no autorizados (regla 105501 para alteraciones de permisos root), y la monitorización de eventos relacionados con la herramienta de seguridad Palo Alto GlobalProtect (regla 64500). Esta configuración refleja un equilibrio entre seguridad operativa y adaptación a las necesidades específicas de la infraestructura.

3.7.5 Dashboards personalizados

Los "*dashboards"* permiten visualizar alertas críticas, comportamiento de "*endpoints"*, actividad de red y tendencias históricas. Se desarrollan "*dashboards"* específicos para:

- Actividad sospechosa en usuarios privilegiados.
- Conexiones remotas no autorizadas.
- Modificaciones de archivos críticos.
- Dispositivos de red con errores o accesos anómalos.

Los "*dashboards"* se basan en visualizaciones como histogramas, mapas de calor y diagramas circulares, facilitando la toma de decisiones por el personal de seguridad. Como se observa en la figura 26, se creó el "*dashboard*" con los eventos más relevantes, "*ssh"* fallido en servidores Linux y "*switches"*, además las 10 alertas más repetidas, lista de malware escaneado con Virus Total, acceso fallido a "*Global Protect* "de PaloAlto, detección de "*powershell"*, "*cmd"*.

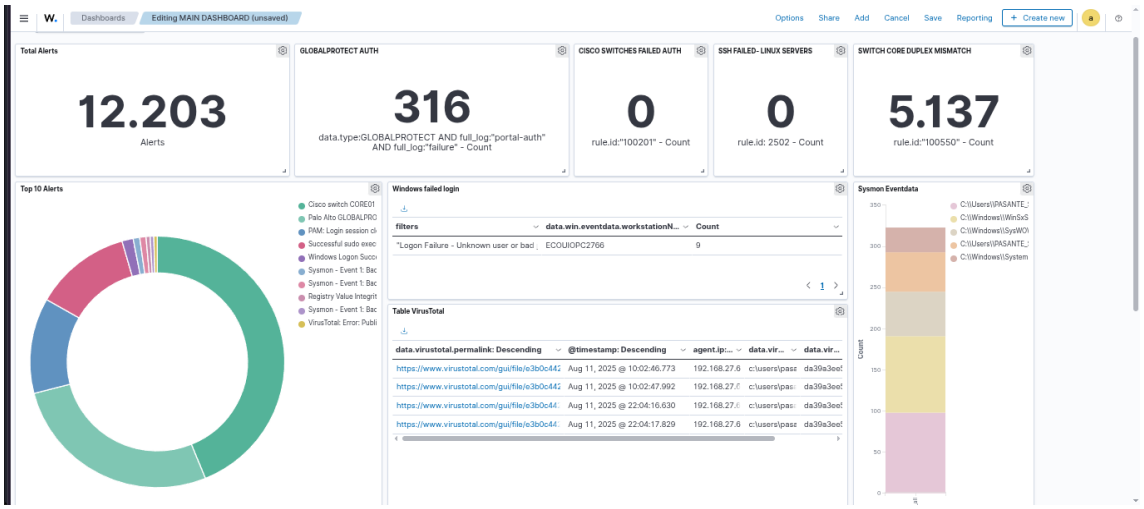


Figura 26. Dashboard implementado, resumiendo todas las actividades.

3.8 Limitaciones y trabajo futuro

Esta implementación se llevó a cabo en un entorno de pruebas con recursos limitados, sobre un subconjunto de equipos. Si bien el sistema SIEM se encuentra completamente funcional, su despliegue a toda la infraestructura de ECOILPET se plantea como un trabajo futuro. Se recomienda:

- Expandir la instalación de agentes Wazuh a todos los endpoints de la organización.
- Optimizar el hardware según las especificaciones recomendadas.
- Desarrollar políticas de alertas más afinadas para distintos perfiles de riesgo.
- Implementar mecanismos de respaldo, alta disponibilidad y rotación de logs a largo plazo.

3.9 Implementación de mitre caldera para pruebas de ataques simulados

Para validar la efectividad del sistema SIEM Wazuh instalado, se desplegó la plataforma de simulación de ataques MITRE Caldera en el mismo servidor donde se ejecuta Wazuh Manager, utilizando Docker para facilitar su instalación y gestión.

La instalación se realizó mediante la descarga de la imagen oficial de CALDERA y la ejecución del contenedor con los puertos necesarios expuestos para acceso web y comunicación con agentes:

docker pull mitre/caldera:latest
docker run -d -p 8888:8888 --name caldera mitre/caldera:latest

Se configuró el acceso a la interfaz web del sistema para gestionar las pruebas y despliegue de agentes.

En paralelo, se instalaron agentes de Caldera en hosts Windows y Linux dentro del entorno de pruebas para ejecutar ataques simulados basados en técnicas MITRE ATT&CK. Estos agentes interactúan con la plataforma CALDERA y permiten generar escenarios de ataque que son detectados por el sistema SIEM Wazuh, generando alertas en tiempo real. Esta metodología permite:

- Validar las reglas y decodificadores personalizados previamente configurados.
- Detectar ataques reales y actividades sospechosas en endpoints.
- Ajustar las políticas de respuesta activa y los dashboards para una mejor visualización.

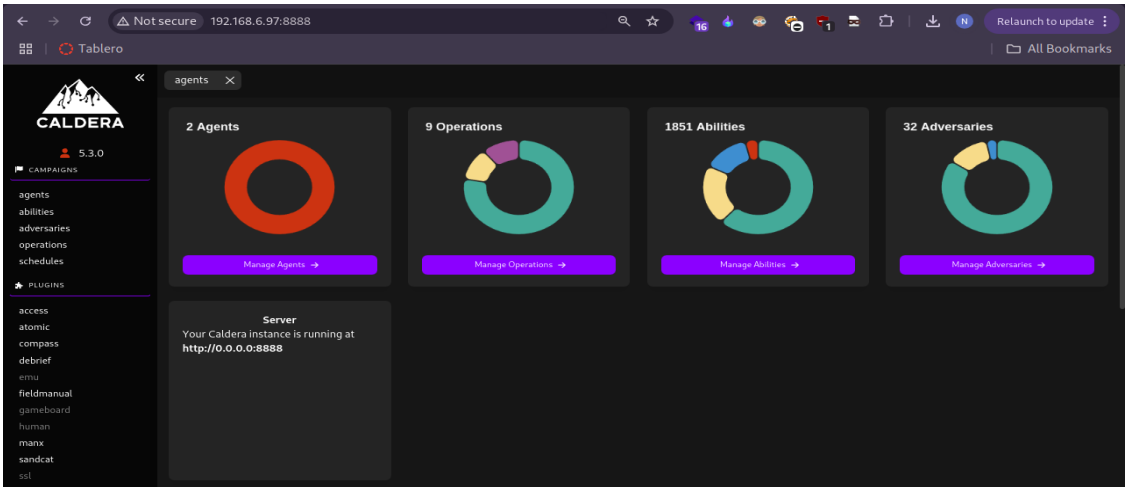


Figura 27. Caldera para simulación de ataques en el entorno de pruebas.

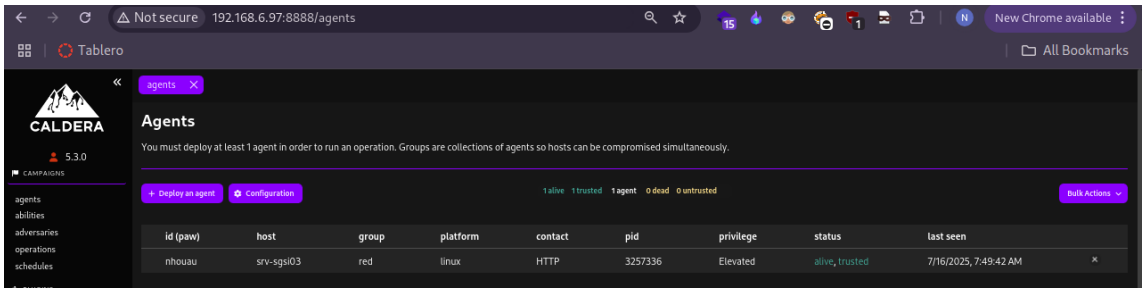


Figura 28. Agente para servidores Linux.

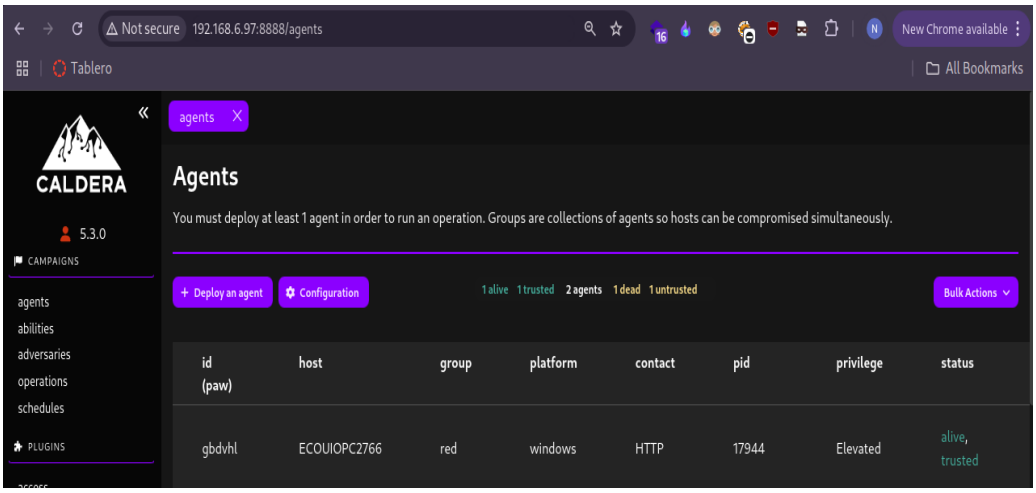


Figura 29. Agente para desktop Windows.

El despliegue y prueba con Caldera representa un paso importante en la madurez del sistema SIEM, pues permite no solo recopilar logs sino también validar la detección proactiva y la capacidad de respuesta frente a ataques avanzados.

4. CAPÍTULO 4

PRUEBAS, RESULTADOS Y VALIDACIÓN

4.1 Diseño de pruebas

4.1.1 Ataques simulados

Para validar el funcionamiento del sistema SIEM Wazuh en la infraestructura de ECOILPET, se realizaron ataques simulados utilizando MITRE Caldera. Esta herramienta permite automatizar campañas de emulación de adversarios basadas en el modelo ATT&CK, replicando técnicas reales utilizadas por atacantes avanzados (MITRE, 2024). En sistemas Linux, se ejecutaron dos ataques principales: la creación de un nuevo usuario con privilegios administrativos, y la sustitución del archivo de tareas programadas "*crontab"* por uno manipulado para lograr persistencia en el sistema. Ambos eventos fueron detectados exitosamente por Wazuh, generando alertas y respuestas automáticas configuradas previamente a través del módulo Active Response.

En equipos con Windows, las pruebas se centraron en ataques comunes que comprometen la seguridad de los "*endpoints"*. Se utilizó PowerShell para crear usuarios de forma no autorizada, ejecutar comandos codificados en Base64, una técnica ampliamente empleada para evadir mecanismos tradicionales de detección, y simular técnicas de "*masquerading"*, como la ejecución de powershell.exe bajo el nombre de proceso taskhostw.exe. Esta última técnica es reconocida por su capacidad de disfrazar procesos maliciosos bajo nombres legítimos del sistema. Todos estos ataques fueron capturados por el agente Wazuh utilizando integraciones con "*Sysmon"* y reglas de detección personalizadas.

Adicionalmente, se configuraron reglas específicas para detectar la ejecución de cmd.exe, herramienta que suele ser utilizada en secuencias de ataque manuales. Esta detección se logró mediante el análisis de "*logs"* generados por "*Sysmon*", recolectados a través de "*Syslog"* hacia el SIEM. Para mejorar la precisión de las detecciones, se realizaron exclusiones en el archivo sysmon.xml, eliminando eventos rutinarios del sistema y priorizando aquellos considerados críticos para la seguridad, como la carga de módulos sospechosos o la ejecución de scripts anómalos.

También se simularon ataques hacia los dispositivos de red de la empresa, incluyendo accesos maliciosos a Cisco Meraki. Los logs generados por estos dispositivos fueron recolectados por Wazuh vía "*Syslog"*, permitiendo correlacionar intentos de acceso o cambios no autorizados. Las direcciones IP de origen asociadas a estos accesos se analizaron en tiempo real utilizando la API de AbuseIPDB, como se ve en la figura de abajo, lo que permitió bloquear aquellas que tenían una reputación negativa.

El script analiza un archivo de logs de Wazuh para identificar direcciones IP que intentaron acceder al Meraki. Extrae las IPs de cada línea relevante y evita revisar aquellas que ya fueron consultadas ese día o que ya están registradas como sospechosas. Luego, consulta la API de AbuseIPDB para obtener la puntuación de abuso de cada IP y, si esta es igual o superior al 50%, la guarda en una lista de IPs maliciosas, que podrá ser utilizada posteriormente para acciones como bloqueo o eliminación. Todo el proceso queda registrado tanto en consola como en un archivo de log para seguimiento.

```python
import re
import requests
import os
import sys
import time
from datetime import datetime

# ═══ CONFIGURACIÓN ═══
API_KEY = "REDACTED"
LOG_FILE = "/var/ossec/logs/alerts/alerts.log"
BASE_DIR = "/home/userconfig/abuseIP"

BAN_LIST_FILE = os.path.join(BASE_DIR, "ban.txt")
CHECKED_FILE = os.path.join(BASE_DIR, "checked_today.txt")
BAD_TODAY_FILE = os.path.join(BASE_DIR, "bad_today.txt")
LOG_OUTPUT_FILE = os.path.join(BASE_DIR, "abuselog.txt")

HEADERS = {"Key": API_KEY, "Accept": "application/json"}

# ═══ LOGGING EN CONSOLA Y ARCHIVO ═══
class TeeLogger:
    def __init__(self, filename):
        self.terminal = sys.stdout
        self.log = open(filename, "a")

    def write(self, message):
        self.terminal.write(message)
        self.log.write(message)

    def flush(self):
```

Figura 30. Script “abuse.py”.

4.1.2 Casos de uso definidos

Durante las pruebas de validación del sistema SIEM, se ejecutaron ataques simulados representativos de amenazas reales, enfocados en diferentes componentes de la infraestructura de ECOILPET. En sistemas Linux, se realizaron intentos de acceso malicioso mediante SSH, incluyendo escenarios de fuerza bruta y uso de credenciales vulneradas, así como modificaciones no autorizadas a archivos críticos como /etc/passwd. Para entornos Windows, las pruebas se centraron en técnicas ofensivas comunes como la ejecución de scripts PowerShell codificados en Base64 (T1059.001 según MITRE ATT&CK) y el uso de cmd.exe para comandos sospechosos como descargas remotas de archivos (T1059.003). En la capa de red, se simularon ataques contra la VPN GlobalProtect de Palo Alto mediante conexiones no autenticadas, junto con accesos malintencionados a switches Cisco y dispositivos Meraki que incluían intentos de conexión por puertos no habituales y cambios no autorizados en configuraciones.

4.2 Registro de resultados

4.2.1 Detección de amenazas

Los resultados de las pruebas mostraron una detección efectiva de amenazas simuladas. Por ejemplo, en Linux, los intentos de fuerza bruta SSH activaron la regla 5763 de Wazuh, ejecutando automáticamente la acción de bloquear la IP atacante mediante Active Response. En Windows, las ejecuciones de PowerShell codificadas en Base64 generaron alertas mediante la regla 100201. Estos eventos fueron registrados visualmente en el "*dashboard"* de Kibana.

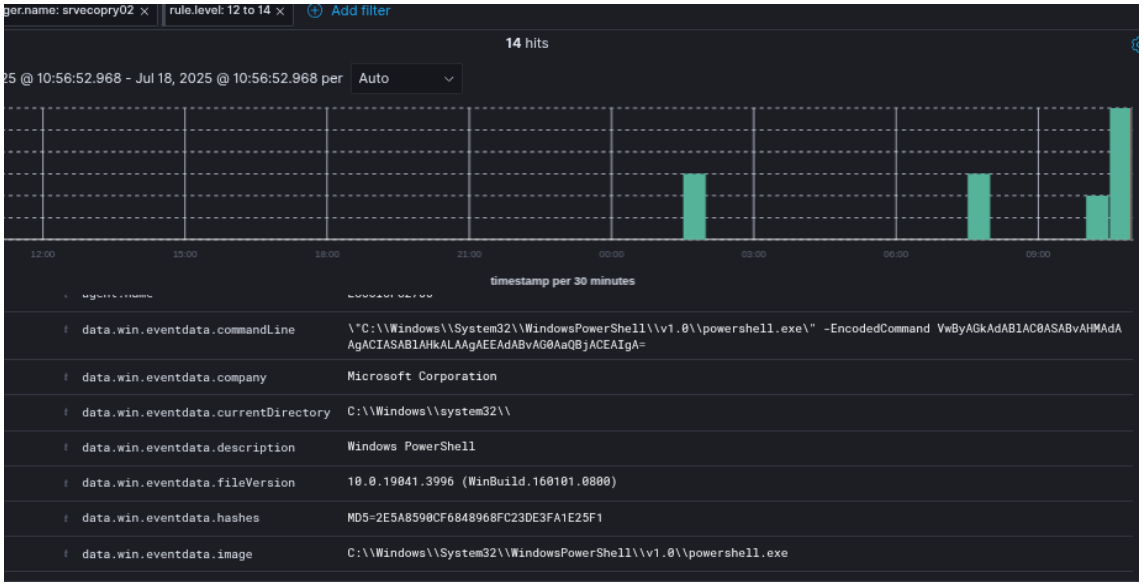


Figura 31. Alerta de PowerShell codificado.

La detección de cmd.exe también se logró a través de reglas personalizadas en "*Sysmon"* y correlacionadas por Wazuh. En infraestructura de red, los logs de Palo Alto y Meraki permitieron detectar eventos anómalos, como intentos de acceso fuera del horario laboral o cambios de configuración. Las IPs involucradas se compararon con listas negras usando AbuseIPDB, fortaleciendo el análisis forense.

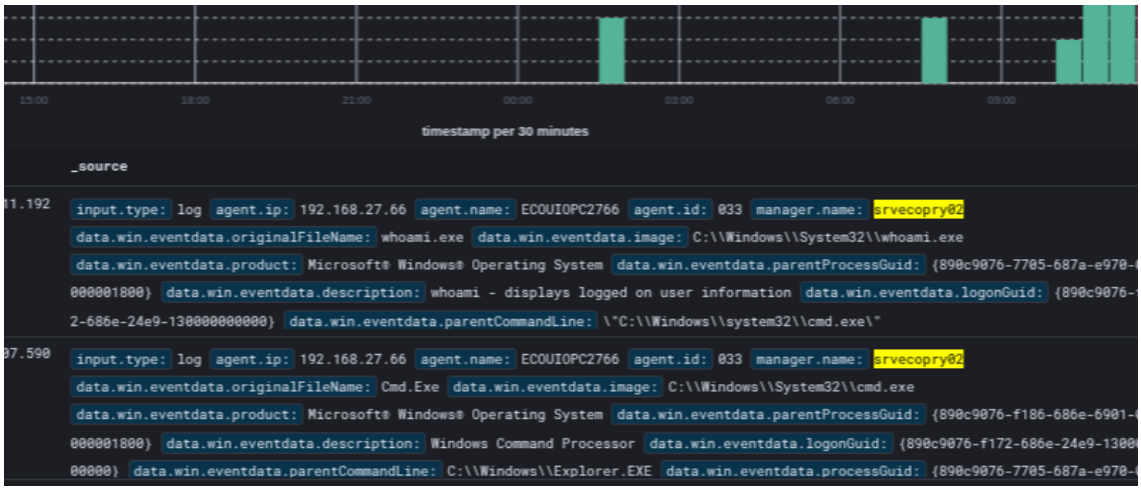


Figura 32. Alerta de cmd ejecutado.

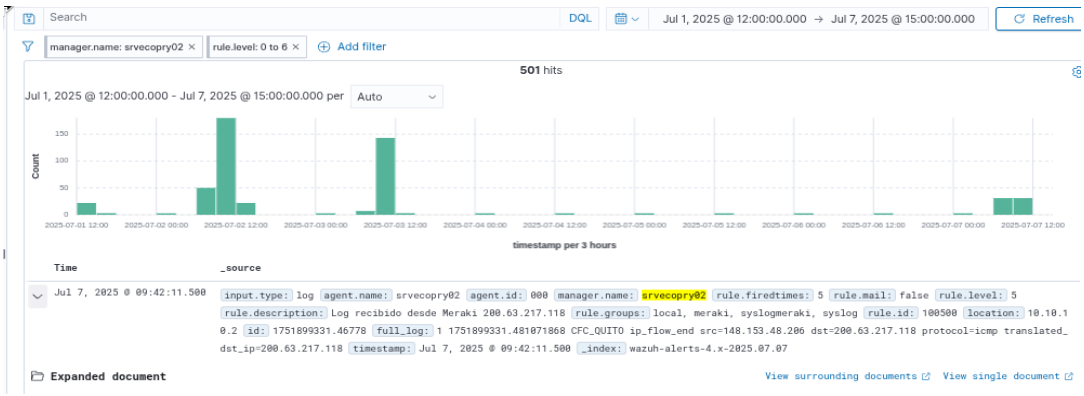


Figura 33. Alerta conexiones a Cisco Meraki.

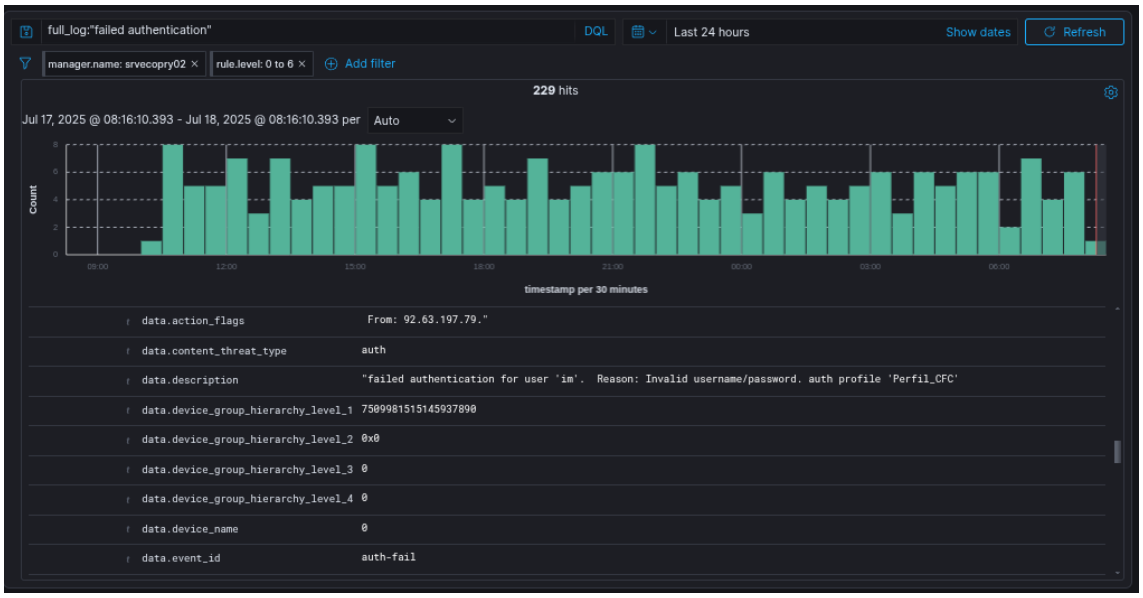


Figura 34. Alerta autenticación fallida en Palo Alto.

4.2.2 Alertas generadas

El sistema Wazuh demostró su efectividad generando alertas clasificadas según su nivel de severidad durante las pruebas realizadas. Las alertas de mayor prioridad incluyeron la detección de ejecución de PowerShell codificado en Base64 (activando la regla 100201) y múltiples intentos fallidos de acceso SSH desde direcciones IP no autorizadas (regla 5763), lo que activó respuestas automáticas como el bloqueo de las IPs implicadas. En el nivel de riesgo medio, el SIEM identificó correctamente eventos como la ejecución de cmd.exe con parámetros sospechosos, detectado mediante reglas personalizadas basadas en "*logs"* de "*Sysmon"* y los intentos de ingreso fallido al firewall Palo Alto realizados fuera del horario laboral establecido. Cada alerta generada proporcionó información detallada incluyendo fecha y hora exacta, dirección IP de origen, sistema afectado, usuario involucrado y acción tomada, facilitando tanto el análisis forense inmediato como la generación de evidencias para cumplimiento normativo.

4.2.3 Tiempo de respuesta

El tiempo de respuesta fue uno de los aspectos más destacables. Al simular múltiples intentos de SSH fallidos, Wazuh reaccionó en menos de 10 segundos, bloqueando la IP ofensora. En Windows, la combinación de FIM y VirusTotal permitió la detección y eliminación de archivos maliciosos en menos de un minuto. Esta capacidad redujo el MTTD/MTTR, alineándose con estándares internacionales de seguridad.

4.3 Análisis del desempeño del sistema

A nivel de desempeño, el sistema operó satisfactoriamente en un entorno con recursos limitados (1 CPU, 4 GB RAM). La carga promedio de CPU se mantuvo por debajo del 40% y la memoria usada por Elasticsearch fue estable. Esto demuestra que Wazuh puede funcionar eficientemente en entornos medianos como el de ECOILPET. Se logró mantener un bajo nivel de falsos positivos gracias a ajustes personalizados en reglas y decodificadores. Esto coincide con la literatura, donde se señala que la personalización mejora significativamente la precisión del monitoreo.

4.4 Cumplimiento normativo

La incorporación de Wazuh como plataforma SIEM dentro del Sistema de Gestión de la Seguridad de la Información (SGSI) refuerza el cumplimiento de la ISO/IEC 27001:2022 en áreas tanto técnicas como organizativas, especialmente en el registro y la monitorización de eventos, la gestión de vulnerabilidades, el control de configuraciones y la gestión de incidentes. Wazuh centraliza y protege los registros, posibilitando su recolección, análisis y correlación en tiempo real (controles A.8.15 y A.8.16), proporciona funcionalidades para identificar, priorizar y dar seguimiento a vulnerabilidades técnicas (A.8.8) y dispone de un módulo de evaluación de configuraciones que facilita la auditoría y la generación de evidencia de configuraciones

seguras (A.8.9). Del mismo modo, la capacidad de visibilidad y trazabilidad que ofrece contribuye a los procesos de planificación, detección, respuesta y aprendizaje frente a incidentes (A.5.24–A.5.27).

No obstante, conviene considerar a Wazuh como un elemento de soporte técnico dentro del SGSI y no como sustituto de las políticas, la gobernanza y la formación necesarias para alcanzar la certificación. En relación con la Ley Orgánica de Protección de Datos Personales (LOPDP), Wazuh apoya los aspectos técnicos del cumplimiento, al implementar medidas como control de accesos, trazabilidad de eventos y respuesta ante incidentes (Asamblea Nacional, 2021). Sin embargo, debe aclararse que un SIEM no garantiza el cumplimiento total de la LOPDP. El cumplimiento legal requiere también de políticas internas, gestión de riesgos, evaluación de impacto, y la designación de un delegado de protección de datos.

5. CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones generales del proyecto

La implementación del sistema SIEM basado en Wazuh en ECOILPET demostró ser una solución eficiente y costo-efectiva para mejorar la postura de seguridad de la organización. A través de la centralización de logs, correlación de eventos y generación de alertas en tiempo real, se logró una visibilidad completa de la infraestructura tecnológica. Los resultados mostraron que Wazuh es capaz de detectar amenazas comunes como intentos de fuerza bruta, ejecución de scripts maliciosos y accesos no autorizados

Un aspecto destacable fue la flexibilidad de la plataforma para integrarse tanto con agentes instalados en "*endpoints"* como con dispositivos de red mediante monitoreo "*agentless"*. Esta característica permitió cubrir los activos críticos sin impactar el desempeño operativo. Además, al ser una solución código abierto, se evitó el alto costo de licenciamiento asociado a alternativas comerciales, logrando un ahorro en el presupuesto de seguridad.

5.2 Aportes a la empresa ECOILPET S.A.

Para ECOILPET, este proyecto significó un avance importante en su madurez de ciberseguridad. La implementación del SIEM permitió al equipo de TI visualizar todos los eventos de seguridad a través de "*dashboards"* interactivos en Kibana, lo que facilitó la toma de decisiones basada en datos Desde el punto de vista normativo, estas capacidades técnicas se traducen en contribuciones concretas a la conformidad con marcos regulatorios: por una parte, Wazuh refuerza los controles de ISO/IEC 27001:2022 relacionados con el registro, la monitorización y la gestión de incidentes antes indicados; por otra, favorece el cumplimiento

de la Ley Orgánica de Protección de Datos Personales del Ecuador (LOPDP) al garantizar la trazabilidad de accesos a información sensible y generar evidencias auditables.

Es importante señalar que estas aportaciones son de naturaleza técnica y deben integrarse con políticas, gobernanza y formación dentro del SGSI para alcanzar una conformidad y certificación completas. La implementación de reglas personalizadas y la integración con herramientas como VirusTotal y AbuseIPDB mejoraron notablemente la precisión del sistema. Estas mejoras técnicas permitieron reducir los falsos positivos y optimizar el uso de recursos computacionales, demostrando que Wazuh puede adaptarse efectivamente a las necesidades específicas de una organización mediana como ECOILPET.

5.3 Recomendaciones para futuras implementaciones

Para maximizar el potencial del SIEM implementado, se recomienda expandir su cobertura a todos los dispositivos críticos de la red que actualmente no están siendo monitoreados. Esto incluye servidores adicionales y estaciones de trabajo que aún no cuentan con agentes instalados. Una cobertura completa permitiría tener una visión completa de la postura de seguridad de la empresa. Otra mejora importante sería la integración con plataformas SOAR ("*Security Orchestration, Automation and Response*") para automatizar respuestas ante incidentes complejos.

Herramientas como TheHive o Shuffle podrían conectarse con Wazuh para ejecutar flujos de trabajo automatizados, como el bloqueo de direcciones IP maliciosas o la notificación inmediata al equipo de seguridad a través de canales como Slack o Microsoft Teams. Además, se sugiere incorporar fuentes externas de inteligencia de amenazas (*Threat Intelligence*) para enriquecer el análisis de seguridad. Plataformas como MISP (*Malware Information Sharing Platform*) o AlienVault OTX podrían proporcionar información actualizada sobre indicadores de compromiso (*IOCs*) que mejorarían la capacidad de detección proactiva del sistema.

5.4 Sostenibilidad y mantenimiento del sistema

Para garantizar el funcionamiento óptimo del SIEM a largo plazo, es fundamental establecer un plan de mantenimiento periódico. Este plan debería incluir la actualización mensual de las reglas de detección, utilizando los repositorios oficiales de Wazuh y las comunidades de ciberseguridad. También se recomienda realizar simulacros de ataque trimestrales utilizando herramientas como MITRE Caldera, para validar continuamente la efectividad del sistema.

La capacitación continua del personal es otro aspecto crítico para la sostenibilidad. El equipo de TI debería recibir formación regular sobre el análisis de alertas, la interpretación de "*dashboards"* y la respuesta a incidentes, preferiblemente con un enfoque en el "*framework"* MITRE ATT&CK para entender las tácticas y técnicas de los atacantes.

Desde el punto de vista técnico, se sugiere implementar mecanismos de alta disponibilidad para los componentes críticos del SIEM, como el Wazuh Indexer y el servidor manager. Configurar réplicas en nodos separados y políticas de respaldo robustas evitaría la posible pérdida de datos en caso de fallos de hardware o desastres.

5.5 Líneas futuras de investigación

El desarrollo e implementación de este sistema SIEM basado en Wazuh abre diversas oportunidades para futuras investigaciones que podrían ampliar significativamente sus capacidades. Una de las áreas más prometedoras es la aplicación de inteligencia artificial para el procesamiento y análisis de logs de seguridad. Mediante técnicas de "*machine learning"* y procesamiento de lenguaje natural, se podrían desarrollar modelos capaces de identificar patrones complejos en grandes volúmenes de datos, clasificando automáticamente incidentes según su gravedad y reduciendo el volumen de trabajo del equipo de seguridad. Estos algoritmos podrían entrenarse para detectar correlaciones entre eventos aparentemente

inconexos, mejorando la identificación de amenazas avanzadas y ataques persistentes que suelen pasar desapercibidos en los sistemas tradicionales.

Otra línea de investigación fundamental sería la expansión del SIEM hacia entornos industriales y de "*IoT*", que presentan desafíos técnicos particulares. La creciente interconexión entre redes corporativas y sistemas operacionales demanda soluciones especializadas que puedan manejar protocolos industriales como Modbus o DNP3, al tiempo que respeten los estrictos requisitos de disponibilidad y rendimiento de estos entornos críticos. Sería necesario desarrollar conectores específicos, reglas de detección adaptadas a amenazas industriales y mecanismos de monitorización que no afecten el funcionamiento de equipos sensibles, todo ello manteniendo los principios de seguridad por diseño.

En cuanto a la integración de las consultas para detectar si una dirección IP es maliciosa mediante el script abuse.py, se propone automatizar el bloqueo de dichas direcciones IP, extendiendo esta capacidad no solo a dispositivos Meraki, sino también a firewalls de próxima generación como Palo Alto, con el objetivo de fortalecer la respuesta automática frente a incidentes de seguridad. En el ámbito normativo, resulta esencial investigar cómo alinear el sistema con los marcos regulatorios emergentes en materia de privacidad y protección de datos. La integración con estándares como ISO/IEC 27701 para gestión de privacidad y el NIST "*Privacy Framework"* permitiría al SIEM no solo detectar brechas de seguridad, sino también violaciones a políticas de privacidad.

Esto implicaría desarrollar funcionalidades avanzadas como el enmascaramiento automático de información sensible en los logs, la detección de accesos no autorizados a datos personales y la generación automatizada de evidencias para auditorías de cumplimiento. La automatización de respuestas mediante plataformas SOAR representa otra área con gran potencial de desarrollo. Investigar la integración con herramientas como Ansible para crear flujos de trabajo complejos

permitiría pasar de la mera detección a la contención automática de incidentes. Sería particularmente valioso desarrollar "*playbooks"* de respuesta alineados con el "*framework"* MITRE ATT&CK, capaces de ejecutar secuencias de acciones defensivas específicas para cada técnica de ataque identificada, mejorando así la capacidad de respuesta frente amenazas sofisticadas.

Por último, para mantener la efectividad del sistema a largo plazo, es crucial establecer mecanismos de actualización constante basados en la colaboración con la comunidad de ciberseguridad. Esto implica investigar metodologías para incorporar rápidamente nuevos indicadores de compromiso *(IOCs*) y tácticas de ataque emergentes, así como participar activamente en redes de intercambio de información sobre amenazas. La creación de procesos estandarizados para absorber el conocimiento colectivo de la comunidad permitiría que el SIEM evolucione al mismo ritmo que las técnicas ofensivas, manteniendo su relevancia como herramienta de defensa en un escenario de amenazas en constante evolución.

6. BIBLIOGRAFIA

Ministerio de Telecomunicaciones y de la Sociedad de la Información del Ecuador. (3 de Agosto de 2022). Obtenido de https://www.telecomunicaciones.gob.ec/por-primera-vez-ecuador-cuenta-con-su-estrategia-nacional-de-ciberseguridad/

Asamblea Nacional del Ecuador. (2021). Obtenido de Ley Orgánica de Protección de Datos Personales. Registro Oficial Suplemento 459.: https://www.registroficial.gob.ec/

Caldera. (2025). Sandcat Plugin Details. Obtenido de https://caldera.readthedocs.io/en/latest/plugins/sandcat/Sandcat-Details.html

Chen, P., Desmet, L., & Huygens, C. (2014). A study on advanced persistent threats. IFIP International Conference on Communications and Multimedia Security, 63–72. doi:https://doi.org/10.1007/978-3-662-44885-4_5

Ghanbari, H., & Koskinen, K. (2024). From SolarWinds to Kaseya: The rise of supply chain attacks in a digital world. Journal of Information Technology Teaching Cases, 10(3), 1-9. doi:https://doi.org/10.1177/20438869241299823

González-Granadillo, G., González-Zarzosa, S., & Díaz, R. (2021). Security Information and Event Management (SIEM): Analysis, trends, and usage in critical infrastructures. Sensors, 21(14). doi:https://doi.org/10.3390/s21144759

Humphreys, E. (2008). Information security management standards: Compliance, governance and risk management. Information Security Technical Report. 247-255. doi:https://doi.org/10.1016/j.istr.2008.10.001

IBM. (23 de Junio de 2023). What is security information and event management (SIEM)? Obtenido de IBM Security Think Blog. : https://www.ibm.com/think/topics/siem

Kaspersky. (30 de agosto de 2023). Empresas latinoamericanas reciben un promedio de dos ataques de ransomware por minuto. Obtenido de Kaspersky: https://latam.kaspersky.com/about/press-releases/empresas-latinoamericanas-reciben-un-promedio-de-dos-ataques-de-ransomware-por-minuto-senala-kaspersky

Kaspersky Lab. (2025). Obtenido de SIEM comercial frente a SIEM de código abierto: ventajas y desventajas.: https://www.kaspersky.es/blog/open-source-siem-hidden-costs/31066/

Kent, K., & Souppaya, M. (2006). Guide to computer security log management. Journal of Information Systems Security, 2(4), 15-29. doi:https://doi.org/10.6028/NIST.SP.800-92

Ministerio de Telecomunicaciones y de la Sociedad de la Información del Ecuador. (18 de Septiembre de 2024). Obtenido de https://www.telecomunicaciones.gob.ec/ecuador-avanza-aceleradamente-en-materia-de-ciberseguridad-segun-estudio-de-la-union-internacional-de-telecomunicaciones/

Ministerio de Telecomunicaciones y de la Sociedad de la Información del Ecuador. (2023). Obtenido de Talleres de ciberseguridad.: https://www.telecomunicaciones.gob.ec/talleres-de-ciberseguridad/

MITRE. (2023). MITRE ATT&CK®. Obtenido de https://attack.mitre.org/

MITRE. (2023). Caldera. Obtenido de https://caldera.mitre.org/

Protección de datos personales. (3 de Junio de 2025). Obtenido de Tecnologia: https://tecnologia.quito.gob.ec/proteccion-de-datos-personales/

Red-Orbita. (3 de Mayo de 2022). Emulación de ataques mediante Atomic Red Team y detección con Azure Sentinel (ATT&CK T1003). . Obtenido de https://red-orbita.com/?p=10408

Shawe, R., & McAndrew, I. (2023). Increasing threats to United States of America infrastructure based on cyber-attacks. Journal of Software Engineering and Applications, 16(10), 530–547. doi:https://doi.org/10.4236/jsea.2023.1610027

Sheeraz, M., et al. (2024). Revolutionizing SIEM security: an innovative correlation engine design for multi-layered attack detection. . doi:https://doi.org/10.3390/s24154901

Superintendencia. (2025). PORTAL DE INFORMACIÓN / CONSULTA DE COMPAÑÍAS. Obtenido de https://appscvsgen.supercias.gob.ec/consultaCompanias/societario/informacionCompanias.jsf

SwiftOnSecurity. (2021). GitHub. Obtenido de https://github.com/SwiftOnSecurity/sysmon-config

Vectra AI. (2024). Obtenido de ¿Qué es Metasploit?. Vectra AI Blog. : https://es.vectra.ai/topics/metasploit

Von Solms, R., & Niekerk, V. J. (2013). From information security to cyber security. Computers & Securit. 38,97-102. doi:https://doi.org/10.1016/j.cose.2013.04.004

Wazuh. (2025). Detecting and removing malware using VirusTotal integration. Obtenido de https://documentation.wazuh.com/current/proof-of-concept-guide/detect-remove-malware-virustotal.html

Wazuh, I. (2024). Elastic Stack integration. Wazuh Documentation. . Obtenido de https://documentation.wazuh.com/current/integrations-guide/elastic-stack/index.html

Wazuh, I. (2025). Getting Started – Components. Wazuh Documentation. Obtenido de https://documentation.wazuh.com/current/getting-started/components.html

7. ANEXOS

7.1 A: Archivo ossec.conf de Wazuh Manager

```
<!--
  Wazuh - Manager - Default configuration for ubuntu 24.04
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>yes</logall>
    <logall_json>yes</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>

<agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>
```

```
<!--changed to level 6 to dont get trash -->
<!--make new rule to change rule ids to 3 or more -->
<!--change to 0 for all -->
  <alerts>
    <log_alert_level>2</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

  <!-- Choose between "plain", "json", or "plain,json" for the format
of internal logs -->
  <logging>
    <log_format>plain</log_format>
  </logging>

  <remote>
    <connection>secure</connection>
    <port>1514</port>
    <protocol>tcp</protocol>
    <queue_size>131072</queue_size>
  </remote>

<!-- Meraki QUITO CFC syslogs -->

<remote>
  <connection>syslog</connection>
  <port>514</port>
  <protocol>udp</protocol>
  <allowed-ips>10.10.10.2</allowed-ips>
```

```
</remote>

<!--SW PISO 5 CFC -->
<remote>
  <connection>syslog</connection>
  <port>514</port>
  <protocol>udp</protocol>
  <allowed-ips>192.168.1.245</allowed-ips>
</remote>

<!-- SW CORE -->
<remote>
  <connection>syslog</connection>
  <port>514</port>
  <protocol>udp</protocol>
  <allowed-ips>192.168.1.243</allowed-ips>
</remote>

<!--PaloAltoNetworks-Main-->
<remote>
  <connection>syslog</connection>
  <port>514</port>
  <protocol>udp</protocol>
  <allowed-ips>any</allowed-ips>
  <local_ip>192.168.6.97</local_ip>
</remote>


  <!-- Policy monitoring -->
```

```
  <rootcheck>
    <disabled>no</disabled>
    <check_files>yes</check_files>
    <check_trojans>yes</check_trojans>
    <check_dev>yes</check_dev>
    <check_sys>yes</check_sys>
    <check_pids>yes</check_pids>
    <check_ports>yes</check_ports>
    <check_if>yes</check_if>

    <!-- Frequency that rootcheck is executed - every 12 hours -->
    <frequency>43200</frequency>

    <rootkit_files>etc/rootcheck/rootkit_files.txt</rootkit_files>

<rootkit_trojans>etc/rootcheck/rootkit_trojans.txt</rootkit_trojans>

    <skip_nfs>yes</skip_nfs>

    <ignore>/var/lib/containerd</ignore>
    <ignore>/var/lib/docker/overlay2</ignore>
  </rootcheck>

  <wodle name="cis-cat">
    <disabled>yes</disabled>
    <timeout>1800</timeout>
    <interval>1d</interval>
    <scan-on-start>yes</scan-on-start>
```

```
  <java_path>wodles/java</java_path>
  <ciscat_path>wodles/ciscat</ciscat_path>
</wodle>

<!-- Osquery integration -->
<wodle name="osquery">
  <disabled>yes</disabled>
  <run_daemon>yes</run_daemon>
  <log_path>/var/log/osquery/osqueryd.results.log</log_path>
  <config_path>/etc/osquery/osquery.conf</config_path>
  <add_labels>yes</add_labels>
</wodle>

<!-- System inventory -->
<wodle name="syscollector">
  <disabled>no</disabled>
  <interval>1h</interval>
  <scan_on_start>yes</scan_on_start>
  <hardware>yes</hardware>
  <os>yes</os>
  <network>yes</network>
  <packages>yes</packages>
  <ports all="no">yes</ports>
  <processes>yes</processes>

  <!-- Database synchronization settings -->
  <synchronization>
    <max_eps>10</max_eps>
  </synchronization>
```

```
</wodle>

<sca>
  <enabled>yes</enabled>
  <scan_on_start>yes</scan_on_start>
  <interval>12h</interval>
  <skip_nfs>yes</skip_nfs>
</sca>

<vulnerability-detection>
  <enabled>yes</enabled>
  <index-status>yes</index-status>
  <feed-update-interval>60m</feed-update-interval>
</vulnerability-detection>

<indexer>
  <enabled>yes</enabled>
  <hosts>
    <host>https://127.0.0.1:9200</host>
  </hosts>
  <ssl>
    <certificate_authorities>
      <ca>/etc/filebeat/certs/root-ca.pem</ca>
    </certificate_authorities>
    <certificate>/etc/filebeat/certs/wazuh-server.pem</certificate>
    <key>/etc/filebeat/certs/wazuh-server-key.pem</key>
  </ssl>
</indexer>
```

```
<!-- File integrity monitoring -->
<syscheck>
  <disabled>no</disabled>

  <!-- Frequency that syscheck is executed default every 12 hours -->
  <frequency>43200</frequency>

  <scan_on_start>yes</scan_on_start>

  <!-- Generate alert when new file detected -->
  <alert_new_files>yes</alert_new_files>

  <!-- Don't ignore files that change more than 'frequency' times -->
  <auto_ignore frequency="10" timeframe="3600">no</auto_ignore>

  <!-- Directories to check  (perform all possible verifications) -->
  <directories>/etc,/usr/bin,/usr/sbin</directories>
  <directories>/bin,/sbin,/boot</directories>

  <!-- Files/directories to ignore -->
  <ignore>/etc/mtab</ignore>
  <ignore>/etc/hosts.deny</ignore>
  <ignore>/etc/mail/statistics</ignore>
  <ignore>/etc/random-seed</ignore>
  <ignore>/etc/random.seed</ignore>
  <ignore>/etc/adjtime</ignore>
  <ignore>/etc/httpd/logs</ignore>
  <ignore>/etc/utmpx</ignore>
  <ignore>/etc/wtmpx</ignore>
```

```
    <ignore>/etc/cups/certs</ignore>
    <ignore>/etc/dumpdates</ignore>
    <ignore>/etc/svc/volatile</ignore>

    <!-- File types to ignore -->
    <ignore type="sregex">.log$|.swp$</ignore>

    <!-- Check the file, but never compute the diff -->
    <nodiff>/etc/ssl/private.key</nodiff>

    <skip_nfs>yes</skip_nfs>
    <skip_dev>yes</skip_dev>
    <skip_proc>yes</skip_proc>
    <skip_sys>yes</skip_sys>

    <!-- Nice value for Syscheck process -->
    <process_priority>10</process_priority>

    <!-- Maximum output throughput -->
    <max_eps>50</max_eps>

    <!-- Database synchronization settings -->
    <synchronization>
      <enabled>yes</enabled>
      <interval>5m</interval>
      <max_eps>10</max_eps>
    </synchronization>
  </syscheck>
```

```
<!-- Active response -->
<global>
  <white_list>127.0.0.1</white_list>
  <white_list>^localhost.localdomain$</white_list>
  <white_list>127.0.0.53</white_list>
</global>


<command>
  <name>disable-account</name>
  <executable>disable-account</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>

<command>
  <name>restart-wazuh</name>
  <executable>restart-wazuh</executable>
</command>

<command>
  <name>firewall-drop</name>
  <executable>firewall-drop</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>

<command>
  <name>host-deny</name>
  <executable>host-deny</executable>
  <timeout_allowed>yes</timeout_allowed>
```

```
</command>

<command>
  <name>route-null</name>
  <executable>route-null</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>

<command>
  <name>win_route-null</name>
  <executable>route-null.exe</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>

<command>
  <name>netsh</name>
  <executable>netsh.exe</executable>
  <timeout_allowed>yes</timeout_allowed>
</command>

<!--
<active-response>
  active-response options here
</active-response>
-->

<!-- Log analysis -->
<localfile>
  <log_format>command</log_format>
```

```
    <command>df -P</command>
    <frequency>360</frequency>
  </localfile>

  <localfile>
    <log_format>full_command</log_format>
    <command>netstat -tulpn | sed 's/\([[:alnum:]]\+\)\
\+[[:digit:]]\+\ \+[[:digit:]]\+\ \+\(.*\):\([[:digit:]]*\)\ \+\([0-
9\.\:\*]\+\).\+\ \([[:digit:]]*\/[[:alnum:]\-]*\).*/\1 \2 == \3 == \4
\5/' | sort -k 4 -g | sed 's/ == \(.*\) ==/:\1/' | sed 1,2d</command>
    <alias>netstat listening ports</alias>
    <frequency>360</frequency>
  </localfile>

  <localfile>
    <log_format>full_command</log_format>
    <command>last -n 20</command>
    <frequency>360</frequency>
  </localfile>

  <ruleset>
    <!-- Default ruleset -->
    <decoder_dir>ruleset/decoders</decoder_dir>
    <rule_dir>ruleset/rules</rule_dir>
    <rule_exclude>0215-policy_rules.xml</rule_exclude>
    <list>etc/lists/audit-keys</list>
    <list>etc/lists/amazon/aws-eventnames</list>
    <list>etc/lists/security-eventchannel</list>
```

```
    <!-- User-defined ruleset -->
    <decoder_dir>etc/decoders</decoder_dir>
    <rule_dir>etc/rules</rule_dir>
  </ruleset>

  <rule_test>
    <enabled>yes</enabled>
    <threads>1</threads>
    <max_sessions>64</max_sessions>
    <session_timeout>15m</session_timeout>
  </rule_test>

  <!-- Configuration for wazuh-authd -->
  <auth>
    <disabled>no</disabled>
    <port>1515</port>
    <use_source_ip>no</use_source_ip>
    <purge>yes</purge>
    <use_password>no</use_password>

<ciphers>HIGH:!ADH:!EXP:!MD5:!RC4:!3DES:!CAMELLIA:@STRENGTH</ciphers>
    <!-- <ssl_agent_ca></ssl_agent_ca> -->
    <ssl_verify_host>no</ssl_verify_host>
    <ssl_manager_cert>etc/sslmanager.cert</ssl_manager_cert>
    <ssl_manager_key>etc/sslmanager.key</ssl_manager_key>
    <ssl_auto_negotiate>no</ssl_auto_negotiate>
  </auth>

  <cluster>
```

```
    <name>wazuh</name>
    <node_name>node01</node_name>
    <node_type>master</node_type>
    <key></key>
    <port>1516</port>
    <bind_addr>0.0.0.0</bind_addr>
    <nodes>
        <node>NODE_IP</node>
    </nodes>
    <hidden>no</hidden>
    <disabled>yes</disabled>
  </cluster>

</ossec_config>

<ossec_config>
  <localfile>
    <log_format>journald</log_format>
    <location>journald</location>
  </localfile>

  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/apache2/error.log</location>
  </localfile>

  <localfile>
    <log_format>apache</log_format>
    <location>/var/log/apache2/access.log</location>
```

```
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/ossec/logs/active-responses.log</location>
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/dpkg.log</location>
  </localfile>

<!--
meraki api test ERA PARA LEER LOS LOGS DESDE LA API DE MERAKI PERO SE
CAMBIA A SYSLOG


<localfile>
  <log_format>json</log_format>
  <location>/var/log/meraki/events_jsonl.log</location>
</localfile>

-->
</ossec_config>

<!--VT INTEGRATION API KEY remove threat -->
<ossec_config>
  <integration>
    <name>virustotal</name>
```

```
<api_key>9a157aa7484e994648162f38401b7174ac41797eb83d431e77821365c7bced
4b</api_key>
    <group>syscheck</group>
    <alert_format>json</alert_format>
  </integration>
</ossec_config>



<ossec_config>
  <command>
    <name>remove-threat</name>
    <executable>remove-threat.exe</executable>
    <timeout_allowed>no</timeout_allowed>
  </command>

  <active-response>
    <disabled>no</disabled>
    <command>remove-threat</command>
    <location>local</location>
    <rules_id>87105</rules_id>
  </active-response>
</ossec_config>
```

7.2 B: Local rules de Wazuh Manager

```
<!-- Local rules -->


<!--IGNORE WAZUH -->
<group name="local,syscheck">
  <rule id="100600" level="0" overwrite="yes">
    <match>netstat listening ports</match>
    <description>Ignorar alertas que contienen netstat listening
ports</description>
  </rule>
</group>



<group name="local,syslog,cisco,sshd,">
  <rule id="100001" level="5">
    <if_sid>5716</if_sid>
    <srcip>1.1.1.1</srcip>
    <description>sshd: authentication failed from IP
1.1.1.1.</description>
    <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5,</group>
  </rule>


<rule id="100200" level="7">
  <decoded_as>cisco-ssh-reject</decoded_as>
  <match>192.168.1.243</match>
  <description>SW-CORE: SSH rechazado en Cisco</description>
  <group>authentication_failed,switch-core,syslog</group>
</rule>
```

```
<rule id="100201" level="7">
  <decoded_as>cisco-ssh-reject</decoded_as>
  <match>192.168.1.245</match>
  <description>SW-P5-CFC: SSH rechazado en Cisco</description>
  <group>authentication_failed,switch-p5-cfc,syslog</group>
</rule>

<!-- IGNORE MERAKI -->
  <rule id="105501" level="0" overwrite="yes">
    <field name="syscheck.uname_after">root</field>
    <description>Ignorar alertas por syscheck.uname_after
root</description>
  </rule>

</group>



<!-- symon windows bad test sysmon.image -->
<group name="sysmon">
  <!-- Rule for PowerShell and scripts -->
  <rule id="255000" level="12">
    <if_group>sysmon_event1</if_group>
    <field
name="win.eventdata.image">\\powershell.exe||\\.ps1||\\.ps2</field>
    <description>Sysmon - Event 1: Bad exe:
$(win.eventdata.image)</description>
    <group>sysmon_event1,powershell_execution</group>
  </rule>
```

```
  <!-- Rule for cmd.exe -->
  <rule id="255001" level="12">
    <if_group>sysmon_event1</if_group>
    <field name="win.eventdata.image">\\cmd.exe</field>
    <description>Sysmon - Event 1: Ejecución de cmd.exe detectada:
$(win.eventdata.image)</description>
    <group>sysmon_event1,cmd_execution</group>
  </rule>
</group>



<!--PaloAlto Global Protect -->
<group name="paloalto,">
<rule id="64500" level="5" overwrite="yes">
  <if_sid>65000</if_sid>
  <match>GLOBALPROTECT</match>
  <description>Palo Alto GLOBALPROTECT event.</description>
  <group>paloalto,globalprotect</group>
</rule>
</group>
```

7.3 C: Local rules de meraki

```
<group name="local,meraki,syslog,">
<rule id="100500" level="5">
  <match>IP</match>
  <description>Meraki event</description>
  <group>meraki,syslog</group>
</rule>
</group>
```

7.4 D: Local decoders.xml de Wazuh Manager

```
<decoder name="cisco-ssh-reject">
  <prematch>%AAA-W-REJECT:</prematch>
  <regex>%AAA-W-REJECT: New ssh connection, source ([0-9]+\.[0-9]+\.[0-
9]+\.[0-9]+) destination ([0-9]+\.[0-9]+\.[0-9]+\.[0-
9]+)\s+REJECTED</regex>
  <order>program_name</order>
</decoder>
```

7.5 E: Ignore.xml de wazuh manager

```
<group name="local, ossec">
  <rule id="533" level="0" overwrite="yes">
    <description>Ignore alerts for netstat listening
ports</description>
  </rule>



<rule id="5501" level="0" overwrite="yes">
  <srcip>192.168.6.97</srcip>
  <description>Ignore PAM login session opened from Wazuh server
IP</description>
</rule>

</group>




<group name="local,dpkg">
  <rule id="100101" level="0">
    <if_sid>2904</if_sid>
    <decoded_as>dpkg-decoder</decoded_as>
    <description>Ignore Dpkg half-configured alerts (rule
2904)</description>
    <ignore>true</ignore>
  </rule>

  <rule id="100102" level="0">
    <if_sid>2902</if_sid>
```

```
    <decoded_as>dpkg-decoder</decoded_as>
    <description>Ignore Dpkg installed package alerts (rule
2902)</description>
    <ignore>true</ignore>
  </rule>
</group>

<!--testing level 10 to 0 in zabbix -->
<group name="local, zabbix, custom">
  <rule id="100200" level="1">
    <field name="agent.name">Zabbix-Server</field>
    <field name="rule.description">High amount of POST requests in a
small period of time (likely bot).</field>
    <description>Ignorar alertas de alto nivel generadas por Zabbix-
Server (POST flood legítimo)</description>
  </rule>
</group>

<group name="zabbix-ignore">
  <rule id="100333" level="0">
    <if_sid>31533</if_sid>
    <match>zabbix/zabbix.php</match>
    <description>Ignore Zabbix POST requests detected as bot
activity</description>
    <hostname>Zabbix-Server</hostname> <!-- nombre del agente -->
  </rule>
</group>

<!--sysmon-->
```

```
<group name="sysmon,">
  <rule id="100108" level="0">
    <if_sid>255000</if_sid> <!-- Hereda condiciones de la regla que
saltó -->
    <field name="win.eventdata.parentImage">C:\\Program Files
(x86)\\ossec-agent\\wazuh-agent.exe</field>
    <field
name="win.eventdata.image">C:\\Windows\\SysWOW64\\auditpol.exe</field>
    <description>Exclusión: auditpol ejecutado por wazuh-agent
(legítimo)</description>
  </rule>

  <rule id="100109" level="0">
    <if_sid>92213</if_sid>
    <field name="win.eventdata.image">C:\\Program Files
(x86)\\Microsoft\\EdgeUpdate\\MicrosoftEdgeUpdate.exe</field>
    <description>Exclusión: EdgeUpdate creando ejecutable temporal
legítimo</description>
  </rule>
</group>
root@srvecopry02:/var/ossec/etc/rules# systemctl restart wazuh-manager
root@srvecopry02:/var/ossec/etc/rules# ls
a.out       local_rules_meraki.xml  palo_alto.xml
ignore.xml  local_rules.xml
root@srvecopry02:/var/ossec/etc/rules# cat ignore.xml
<group name="local, ossec">
  <rule id="533" level="0" overwrite="yes">
    <description>Ignore alerts for netstat listening
ports</description>
```

```
  </rule>


<rule id="5501" level="0" overwrite="yes">
  <srcip>192.168.6.97</srcip>
  <description>Ignore PAM login session opened from Wazuh server
IP</description>
</rule>

</group>



<group name="local,dpkg">
  <rule id="100101" level="0">
    <if_sid>2904</if_sid>
    <decoded_as>dpkg-decoder</decoded_as>
    <description>Ignore Dpkg half-configured alerts (rule
2904)</description>
    <ignore>true</ignore>
  </rule>

  <rule id="100102" level="0">
    <if_sid>2902</if_sid>
    <decoded_as>dpkg-decoder</decoded_as>
    <description>Ignore Dpkg installed package alerts (rule
2902)</description>
    <ignore>true</ignore>
  </rule>
```

```
</group>

<!--testing level 10 to 0 in zabbix -->
<group name="local, zabbix, custom">
  <rule id="100200" level="1">
    <field name="agent.name">Zabbix-Server</field>
    <field name="rule.description">High amount of POST requests in a
small period of time (likely bot).</field>
    <description>Ignorar alertas de alto nivel generadas por Zabbix-
Server (POST flood legítimo)</description>
  </rule>
</group>

<group name="zabbix-ignore">
  <rule id="100333" level="0">
    <if_sid>31533</if_sid>
    <match>zabbix/zabbix.php</match>
    <description>Ignore Zabbix POST requests detected as bot
activity</description>
    <hostname>Zabbix-Server</hostname> <!-- nombre del agente -->
  </rule>
</group>

<!--sysmon-->
<group name="sysmon,">
  <rule id="100108" level="0">
    <if_sid>255000</if_sid> <!-- Hereda condiciones de la regla que
saltó -->
```

```
    <field name="win.eventdata.parentImage">C:\\Program Files
(x86)\\ossec-agent\\wazuh-agent.exe</field>
    <field
name="win.eventdata.image">C:\\Windows\\SysWOW64\\auditpol.exe</field>
    <description>Exclusión: auditpol ejecutado por wazuh-agent
(legítimo)</description>
  </rule>

  <rule id="100109" level="0">
    <if_sid>92213</if_sid>
    <field name="win.eventdata.image">C:\\Program Files
(x86)\\Microsoft\\EdgeUpdate\\MicrosoftEdgeUpdate.exe</field>
    <description>Exclusión: EdgeUpdate creando ejecutable temporal
legítimo</description>
  </rule>
</group>
```

7.6 F: Sysconfig.xml de endpoints windows

```
<!--
  sysmon-config | A Sysmon configuration focused on default high-quality event tracing and easy customization by the community
  Source version:     74 | Date: 2021-07-08
  Source project:     https://github.com/SwiftOnSecurity/sysmon-config
  Source license:     Creative Commons Attribution 4.0 | You may privatize, fork, edit, teach, publish, or deploy for commercial use - with attribution in the text.

  Fork version:       <N/A>
  Fork author: <N/A>
  Fork project:       <N/A>
  Fork license:       <N/A>

  REQUIRED: Sysmon version 13 or higher (due to changes in syntax and bug-fixes)
        https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

  NOTE: To collect Sysmon logs centrally for free, see https://aka.ms/WEF | Command to allow log access to the Network Service:
        wevtutil.exe sl Microsoft-Windows-Sysmon/Operational /ca:O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)(A;;0x1;;;NS)

  NOTE: Do not let the size and complexity of this configuration discourage you from customizing it or building your own.
```

This configuration is based around known, high-signal event tracing, and thus appears complicated, but it is only very detailed. Significant effort over years has been invested in front-loading as much filtering as possible onto the client. This is to make analysis of intrusions possible by hand, and to try to surface anomalous activity as quickly as possible to technicians armed only with Event Viewer. Its purpose is to democratize system monitoring for all organizations.

NOTE: Sysmon is NOT a whitelist solution or HIDS correlation engine, it is a computer change logging tool.

Do NOT ignore everything possible. Sysmon's purpose is providing context during a threat or problem investigation. Legitimate processes are routinely used by threats - do not blindly exclude them. Additionally, be mindful of process-hollowing / imitation.

NOTE: By default this monitors DNS, which is extremely noisy. If you are starting out on your monitoring journey, just remove that section.

You can remove DNS events from Event Viewer screen by applying a 'Filter Current View' for event IDs of: -22

Additionally, if you want to monitor DNS, you should deploy client-side adblocking to reduce lookups. See the DNS section for info.

NOTE: This configuration is designed for PER-MACHINE installs of Chrome and OneDrive. That moves their binaries out of user-controlled folders.

Otherwise, attackers could imitate these common applications, and bypass your logging. Below are silent upgrades you can do, no user impact:

```
        - https://docs.microsoft.com/en-us/onedrive/per-machine-
installation
        - https://cloud.google.com/chrome-enterprise/browser/download/
        - As of 2021-02-16 there is no machine-level version of
Microsoft Teams. The one provided copies itself to the user profile.

  NOTE: Sysmon is not hardened against an attacker with admin rights.
Additionally, this configuration offers an attacker, willing
        to study it, limited ways to evade some of the logging. If you
are in a very high-threat environment, you should consider a broader,
        log-most approach. However, in the vast majority of cases, an
attacker will bumble through multiple behavioral traps which
        this configuration monitors, especially in the first minutes.

  NOTE: If you encounter unexplainable event inclusion/exclusion, you
may have a second Sysmon instance installed under a different exe
filename.
        To clear this, try downloading the latest version and
uninstalling with -u force. If it hangs, kill the processes and run it
again to cleanup.

  TECHNICAL:
  - Run sysmon.exe -? for a briefing on Sysmon configuration.
  - Sysmon XML cannot use the AMPERSAND sign. Replace it with this:
&amp;
  - Sysmon 8+ can track which rule caused an event to be logged through
the "RuleName" field.
  - If you only specify exclude for a filtering subsection, everything
in that subsection is logged by default.
```

```
  - Some Sysmon monitoring abilities are not meant for widely deployed
general-purpose use due to performance impact. Depends on environment.
  - Duplicate or overlapping "Include" rules do not result in duplicate
events being logged.
  - All characters enclosed by XML tags are always interpreted
literally. Sysmon does not support wildcards (*), alternate characters,
or RegEx.
  - In registry events, the value name is appended to the full key path
with a "\" delimiter. Default key values are named "\(Default)"
  - "Image" is a technical term for a compiled binary file like an EXE
or DLL. Also, it can match just the filename, or entire path.
  - "ProcessGuid" and "LoginGuid" are not random, they contain some
embedded information.
https://gist.github.com/mattifestation/0102042160c9a60b2b847378c0ef70b4

  FILTERING: Filter conditions available for use are: is,is
not,contains,contains any,contains all,excludes,excludes any,excludes
all,begin with,end with,less than,more than,image
  - The "image" filter is usable on any field. Same as "is" but can
either match entire string, or only the text after last "\". Credit:
@mattifestation

-->

<Sysmon schemaversion="4.50">
       <!--SYSMON META CONFIG-->
       <HashAlgorithms>md5,sha256,IMPHASH</HashAlgorithms> <!-- Both
MD5 and SHA256 are the industry-standard algorithms. Remove IMPHASH if
you do not use DLL import fingerprinting. -->
```

<CheckRevocation/> <!-- Check loaded drivers, log if their code-signing certificate has been revoked, in case malware stole one to sign a kernel driver -->

<!-- <ImageLoad/> --> <!-- Would manually force-on ImageLoad monitoring, even without configuration below. Included only documentation. -->

<!-- <ProcessAccessConfig/> --> <!-- Would manually force-on ProcessAccess monitoring, even without configuration below. Included only documentation. -->

<!-- <PipeMonitoringConfig/> --> <!-- Would manually force-on PipeCreated / PipeConnected events, even without configuration below. Included only documentation. -->

<!-- <ArchiveDirectory> -->

<EventFiltering>

<!--SYSMON EVENT ID 1 : PROCESS CREATION [ProcessCreate]-->

<!--COMMENT: All processes launched will be logged, except for what matches a rule below. It's best to be as specific as possible,

to avoid user-mode executables imitating other process names to avoid logging, or if malware drops files in an existing directory.

Ultimately, you must weigh CPU time checking many detailed rules, against the risk of malware exploiting the blindness created.

```
                    Beware of Masquerading, where attackers imitate
the names and paths of legitimate tools. Ideally, you'd use both file
path and
                    code signatures to validate, but Sysmon does not
support that. Look into AppLocker/WindowsDeviceGuard for whitelisting
support. -->

              <!--DATA: UtcTime, ProcessGuid, ProcessID, Image,
FileVersion, Description, Product, Company, CommandLine,
CurrentDirectory, User, LogonGuid, LogonId, TerminalSessionId,
IntegrityLevel, Hashes, ParentProcessGuid, ParentProcessId,
ParentImage, ParentCommandLine, RuleName-->
       <RuleGroup name="" groupRelation="or">
              <ProcessCreate onmatch="exclude">
                     <!-- SECTION EXclude custom (win,sophos)-->
                  <Image
condition="is">C:\Windows\system32\cleanmgr.exe</Image>
                  <Image
condition="is">C:\Windows\System32\sdiagnhost.exe</Image>
                  <Image
condition="is">C:\Windows\System32\dmclient.exe</Image>
                  <Image
condition="is">C:\Windows\servicing\TrustedInstaller.exe</Image>
                  <Image condition="is">C:\Program Files
(x86)\Microsoft\EdgeUpdate\MicrosoftEdgeUpdate.exe</Image>
                  <Image condition="is">C:\Program Files
(x86)\Microsoft\Edge\Application\msedge.exe</Image>
                  <Image condition="is">C:\Program
Files\WinRAR\WinRAR.exe</Image>
```

```
                    <Image condition="is">C:\Program Files\Just Great
Software\EditPad Lite 8\EditPadLite8.exe</Image>
                    <Image
condition="is">C:\Windows\system32\cleanmgr.exe</Image>
                    <Image
condition="is">C:\Windows\System32\sdiagnhost.exe</Image>
                    <Image
condition="is">C:\Windows\System32\dmclient.exe</Image>
                    <Image
condition="is">C:\Windows\servicing\TrustedInstaller.exe</Image>
                    <Image
condition="is">C:\Windows\WinSxS\amd64_microsoft-windows-
servicingstack_31bf3856ad364e35_10.0.19041.6088_none_7e3cb1287c69f05e\T
iWorker.exe</Image>
                    <Image condition="is">C:\Program Files
(x86)\Microsoft\EdgeUpdate\MicrosoftEdgeUpdate.exe</Image>
                    <Image condition="is">C:\Program Files
(x86)\Microsoft\Edge\Application\msedge.exe</Image>
                    <Image
condition="is">C:\Users\PASANTE_SI_2\AppData\Local\Microsoft\OneDrive\2
5.122.0624.0004\FileCoAuth.exe</Image>
                    <Image condition="is">C:\Program
Files\WinRAR\WinRAR.exe</Image>
                    <Image condition="is">C:\Program Files\Just Great
Software\EditPad Lite 8\EditPadLite8.exe</Image>
                    <Image condition="is">C:\Program Files\Sophos\Sophos
File Scanner\SophosFileScanner.exe</Image>
                    <Image condition="is">C:\Program
Files\Sophos\Endpoint Defense\SophosCleanup.exe</Image>
```

```
                    <Image condition="is">C:\Program
Files\Sophos\Endpoint Defense\SophosSafestore.exe</Image>
                    <Image condition="is">C:\Program
Files\Sophos\AutoUpdate\SophosLaunchUpdate.exe</Image>
                    <Image
condition="is">C:\ProgramData\Sophos\AutoUpdate\Cache\sophos_autoupdate
1.dir\su-repair.exe</Image>
                    <Image
condition="is">C:\ProgramData\Sophos\AutoUpdate\Cache\sophos_autoupdate
1.dir\SophosUpdate.exe</Image>
                    <Image
condition="is">C:\Windows\System32\consent.exe</Image>
                    <Image condition="is">C:\Program
Files\RUXIM\PLUGScheduler.exe</Image>

                        <!--SECTION: Microsoft Windows-->
                        <CommandLine condition="begin with">
"C:\Windows\system32\wermgr.exe" "-queuereporting_svc" </CommandLine>
<!--Windows:Windows error reporting/telemetry-->
                        <CommandLine condition="begin
with">C:\Windows\system32\DllHost.exe /Processid</CommandLine> <!--
Windows-->
                        <CommandLine condition="begin
with">C:\Windows\system32\wbem\wmiprvse.exe -Embedding</CommandLine>
<!--Windows: WMI provider host-->
                        <CommandLine condition="begin
with">C:\Windows\system32\wbem\wmiprvse.exe -secured -
Embedding</CommandLine> <!--Windows: WMI provider host-->
```

```
                    <CommandLine
condition="is">C:\Windows\system32\wermgr.exe -upload</CommandLine> <!-
-Windows:Windows error reporting/telemetry-->
                    <CommandLine
condition="is">C:\Windows\system32\SearchIndexer.exe
/Embedding</CommandLine> <!--Windows: Search Indexer-->
                    <CommandLine
condition="is">C:\windows\system32\wermgr.exe -
queuereporting</CommandLine> <!--Windows:Windows error
reporting/telemetry-->
                    <CommandLine
condition="is">\??\C:\Windows\system32\autochk.exe *</CommandLine> <!--
Microsoft:Bootup: Auto Check Utility-->
                    <CommandLine
condition="is">\SystemRoot\System32\smss.exe</CommandLine> <!--
Microsoft:Bootup: Windows Session Manager-->
                    <CommandLine
condition="is">C:\Windows\System32\RuntimeBroker.exe -
Embedding</CommandLine> <!--Windows:Apps permissions [
https://fossbytes.com/runtime-broker-process-windows-10/ ] -->
                    <Image condition="is">C:\Program Files
(x86)\Common Files\microsoft shared\ink\TabTip32.exe</Image> <!--
Windows: Touch Keyboard and Handwriting Panel Helper-->
                    <Image
condition="is">C:\Windows\System32\TokenBrokerCookies.exe</Image> <!--
Windows: SSO sign-in assistant for MicrosoftOnline.com-->
                    <Image
condition="is">C:\Windows\System32\plasrv.exe</Image> <!--Windows:
Performance Logs and Alerts DCOM Server-->
```

```
                    <Image
condition="is">C:\Windows\System32\wifitask.exe</Image> <!--Windows:
Wireless Background Task-->
                    <Image
condition="is">C:\Windows\system32\CompatTelRunner.exe</Image> <!--
Windows: Customer Experience Improvement-->
                    <Image
condition="is">C:\Windows\system32\PrintIsolationHost.exe</Image> <!--
Windows: Printing-->
                    <Image
condition="is">C:\Windows\system32\SppExtComObj.Exe</Image> <!--
Windows: KMS activation-->
                    <Image
condition="is">C:\Windows\system32\audiodg.exe</Image> <!--Windows:
Launched constantly-->
                    <Image
condition="is">C:\Windows\system32\conhost.exe</Image> <!--Windows:
Command line interface host process-->
                    <Image
condition="is">C:\Windows\system32\mobsync.exe</Image> <!--Windows:
Network file syncing-->
                    <Image
condition="is">C:\Windows\system32\musNotification.exe</Image> <!--
Windows: Update pop-ups-->
                    <Image
condition="is">C:\Windows\system32\musNotificationUx.exe</Image> <!--
Windows: Update pop-ups-->
```

```
                    <Image
condition="is">C:\Windows\system32\powercfg.exe</Image> <!--
Microsoft:Power configuration management-->
                    <Image
condition="is">C:\Windows\system32\sndVol.exe</Image> <!--Windows:
Volume control-->
                    <Image
condition="is">C:\Windows\system32\sppsvc.exe</Image> <!--Windows:
Software Protection Service-->
                    <Image
condition="is">C:\Windows\system32\wbem\WmiApSrv.exe</Image> <!--
Windows: WMI performance adapter host process-->
                    <IntegrityLevel
condition="is">AppContainer</IntegrityLevel> <!--Windows: Don't care
about sandboxed processes right now. Will need to revisit this
decision.-->
                    <ParentCommandLine condition="begin
with">%%SystemRoot%%\system32\csrss.exe
ObjectDirectory=\Windows</ParentCommandLine> <!--Windows:CommandShell:
Triggered when programs use the command shell, but doesn't provide
attribution for what caused it-->
                    <ParentCommandLine
condition="is">C:\windows\system32\wermgr.exe -
queuereporting</ParentCommandLine> <!--Windows:Windows error
reporting/telemetry-->
                    <CommandLine
condition="is">C:\WINDOWS\system32\devicecensus.exe
UserCxt</CommandLine>
```

```
                    <CommandLine
condition="is">C:\Windows\System32\usocoreworker.exe -
Embedding</CommandLine>
                    <ParentImage
condition="is">C:\Windows\system32\SearchIndexer.exe</ParentImage> <!--
Windows:Search: Launches many uninteresting sub-processes-->
                    <!--SECTION: Windows:svchost-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k appmodel -s
StateRepository</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k appmodel -p -s
camsvc</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
appmodel</CommandLine> <!--Windows 10-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k appmodel -p -s
tiledatamodelsvc</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k camera -s
FrameServer</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k dcomlaunch -s
LSM</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k dcomlaunch -s
PlugPlay</CommandLine>
```

```
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
defragsvc</CommandLine> <!--Windows defragmentation-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k devicesflow -s
DevicesFlowUserSvc</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k imgsvc</CommandLine>
<!--Microsoft:The Windows Image Acquisition Service-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k localService -s
EventSystem</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k localService -s
bthserv</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k LocalService -p -s
BthAvctpSvc</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k localService -s
nsi</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k localService -s
w32Time</CommandLine>
      <CommandLine condition="is">C:\Windows\system32\svchost.exe -k
localServiceAndNoImpersonation</CommandLine> <!--Windows: Network
services-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
```

```
localServiceAndNoImpersonation -p</CommandLine> <!--Windows: Network
services-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localServiceNetworkRestricted -s Dhcp</CommandLine> <!--Windows:
Network services-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localServiceNetworkRestricted -s EventLog</CommandLine>
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localServiceNetworkRestricted -s TimeBrokerSvc</CommandLine>
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localServiceNetworkRestricted -s WFDSConMgrSvc</CommandLine>
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
LocalServiceNetworkRestricted -s BTAGService</CommandLine>
                        <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k
LocalSystemNetworkRestricted -p -s NcbService</CommandLine> <!--
Win10:1903:Network Connection Broker-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localServiceNetworkRestricted</CommandLine> <!--Windows: Network
services-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localServiceAndNoImpersonation -s SensrSvc</CommandLine>
```

```
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localServiceAndNoImpersonation -p -s SSDPSRV</CommandLine> <!--
Windows:SSDP [
https://en.wikipedia.org/wiki/Simple_Service_Discovery_Protocol ] -->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localServiceNoNetwork</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted -p -s WPDBusEnum</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted -p -s fhsvc</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted -s DeviceAssociationService</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted -s NcbService</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted -s SensorService</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted -s TabletInputService</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted -s UmRdpService</CommandLine>
```

```
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted -s WPDBusEnum</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted -p -s NgcSvc</CommandLine> <!--
Microsoft:Passport-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localServiceNetworkRestricted -p -s NgcCtnrSvc</CommandLine> <!--
Microsoft:Passport Container-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localServiceAndNoImpersonation -s SCardSvr</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -p -s
wuauserv</CommandLine>
                    <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k netsvcs -p -s
SessionEnv</CommandLine> <!--Windows:Remote desktop configuration-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted -s WdiSystemHost</CommandLine> <!--
Windows: Diagnostic System Host [ http://www.blackviper.com/windows-
services/diagnostic-system-host/ ] -->
                    <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k
localSystemNetworkRestricted -p -s WdiSystemHost</CommandLine> <!--
```

```
Windows: Diagnostic System Host [ http://www.blackviper.com/windows-
services/diagnostic-system-host/ ] -->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted</CommandLine> <!--Windows-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -p -s
wlidsvc</CommandLine> <!--Windows: Windows Live Sign-In Assistant [
https://www.howtogeek.com/howto/30348/what-are-wlidsvc.exe-and-
wlidsvcm.exe-and-why-are-they-running/ ] -->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -p -s
ncaSvc</CommandLine> <!--Windows: Network Connectivity Assistant [
http://www.blackviper.com/windows-services/network-connectivity-
assistant/ ] -->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -s
BDESVC</CommandLine> <!--Windows:Network: BitLocker Drive Encryption-->
                        <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k netsvcs -p -s
BDESVC</CommandLine> <!--Microsoft:Win10:1903:Network: BitLocker Drive
Encryption-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -p -s
BITS</CommandLine> <!--Windows:Network: Background Intelligent File
Transfer (BITS) -->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -s
```

```
BITS</CommandLine> <!--Windows:Network: Background Intelligent File
Transfer (BITS) -->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -s
CertPropSvc</CommandLine>
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -s
DsmSvc</CommandLine>
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -p -s
Appinfo</CommandLine>
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -s
Gpsvc</CommandLine> <!--Windows:Network: Group Policy -->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -s
ProfSvc</CommandLine> <!--Windows: Network services-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -s
SENS</CommandLine> <!--Windows: Network services-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -s
SessionEnv</CommandLine> <!--Windows: Network services-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -s
Themes</CommandLine> <!--Windows: Network services-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs -s
```

```
Winmgmt</CommandLine> <!--Windows: Windows Management Instrumentation
(WMI) -->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k netsvcs</CommandLine>
<!--Windows: Network services-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k networkService -p -s
DoSvc</CommandLine>
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k networkService -s
Dnscache</CommandLine> <!--Windows:Network: DNS caching, other uses -->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k networkService -s
LanmanWorkstation</CommandLine> <!--Windows:Network: "Workstation"
service, used for SMB file-sharing connections and RDP-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k networkService -s
NlaSvc</CommandLine> <!--Windows:Network: Network Location Awareness-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k networkService -s
TermService</CommandLine> <!--Windows:Network: Terminal Services (RDP)-
->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
networkService</CommandLine> <!--Windows: Network services-->
                        <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k networkService -
p</CommandLine> <!--Windows: Network services-->
```

```
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
networkServiceNetworkRestricted</CommandLine> <!--Windows: Network
services-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k rPCSS</CommandLine>
<!--Windows Services-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k secsvcs</CommandLine>
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k swprv</CommandLine>
<!--Microsoft:Software Shadow Copy Provider-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
unistackSvcGroup</CommandLine> <!--Windows 10-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k utcsvc</CommandLine>
<!--Windows Services-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
wbioSvcGroup</CommandLine> <!--Windows Services-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k
werSvcGroup</CommandLine> <!--Windows: ErrorReporting-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k wusvcs -p -s
WaaSMedicSvc</CommandLine> <!--Windows: Update Medic Service [
https://www.thewindowsclub.com/windows-update-medic-service ] -->
```

```
                    <CommandLine
condition="is">C:\Windows\System32\svchost.exe -k wsappx -p -s
ClipSVC</CommandLine> <!--Windows:Apps: Client License Service-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k wsappx -p -s
AppXSvc</CommandLine> <!--Windows:Apps: AppX Deployment Service-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k wsappx -s
ClipSVC</CommandLine> <!--Windows:Apps: Client License Service-->
                    <CommandLine
condition="is">C:\Windows\system32\svchost.exe -k wsappx</CommandLine>
<!--Windows:Apps [ https://www.howtogeek.com/320261/what-is-wsappx-and-
why-is-it-running-on-my-pc/ ] -->
                    <ParentCommandLine
condition="is">C:\Windows\system32\svchost.exe -k
netsvcs</ParentCommandLine> <!--Windows: Network services: Spawns
Consent.exe-->
                    <ParentCommandLine
condition="is">C:\Windows\system32\svchost.exe -k
localSystemNetworkRestricted</ParentCommandLine> <!--Windows-->
                    <CommandLine
condition="is">C:\Windows\system32\deviceenroller.exe /c
/AutoEnrollMDM</CommandLine> <!--Windows: AzureAD device enrollment
agent-->
                    <!--SECTION: Microsoft:Edge-->
                    <CommandLine condition="begin with">"C:\Program
Files (x86)\Microsoft\Edge Dev\Application\msedge.exe" --
type=</CommandLine>
                    <!--SECTION: Microsoft:dotNet-->
```

<CommandLine condition="begin with">C:\Windows\Microsoft.NET\Framework\v4.0.30319\ngen.exe</CommandLine> <!--Microsoft:DotNet-->

<CommandLine condition="begin with">C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\Ngen.exe</CommandLine> <!--Microsoft:DotNet-->

<CommandLine condition="begin with">C:\Windows\Microsoft.NET\Framework\v4.0.30319\ngentask.exe</CommandLine> <!--Microsoft:DotNet-->

<CommandLine condition="begin with">C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\ngentask.exe</CommandLine> <!--Microsoft:DotNet-->

<Image condition="is">C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscorsvw.exe</Image> <!--Microsoft:DotNet-->

<Image condition="is">C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscorsvw.exe</Image> <!--Microsoft:DotNet-->

<Image condition="is">C:\Windows\Microsoft.Net\Framework64\v3.0\WPF\PresentationFontCache.exe</Image> <!--Windows: Font cache service-->

<ParentCommandLine condition="begin with">C:\Windows\Microsoft.NET\Framework64\v4.0.30319\ngentask.exe</ParentCommandLine>

<ParentImage condition="is">C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscorsvw.exe</ParentImage> <!--Microsoft:DotNet-->

```
                    <ParentImage
condition="is">C:\Windows\Microsoft.NET\Framework64\v4.0.30319\ngentask
.exe</ParentImage> <!--Microsoft:DotNet-->
                    <ParentImage
condition="is">C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscorsvw.e
xe</ParentImage> <!--Microsoft:DotNet-->
                    <ParentImage
condition="is">C:\Windows\Microsoft.NET\Framework\v4.0.30319\ngentask.e
xe</ParentImage> <!--Microsoft:DotNet: Spawns thousands of ngen.exe
processes-->
                    <!--SECTION: Microsoft:Office-->
                    <Image condition="is">C:\Program Files\Microsoft
Office\Office16\MSOSYNC.EXE</Image> <!--Microsoft:Office: Background
process for SharePoint/Office365 connectivity-->
                    <Image condition="is">C:\Program Files
(x86)\Microsoft Office\Office16\MSOSYNC.EXE</Image> <!--
Microsoft:Office: Background process for SharePoint/Office365
connectivity-->
                    <Image condition="is">C:\Program Files\Common
Files\Microsoft
Shared\OfficeSoftwareProtectionPlatform\OSPPSVC.EXE</Image> <!--
Microsoft:Office: Licensing service-->
                    <Image condition="is">C:\Program Files\Microsoft
Office\Office16\msoia.exe</Image> <!--Microsoft:Office: Telemetry
collector-->
                    <Image condition="is">C:\Program Files
(x86)\Microsoft
Office\root\Office16\officebackgroundtaskhandler.exe</Image>
                    <!--SECTION: Microsoft:Office:Click2Run-->
```

```
                    <Image condition="is">C:\Program Files\Common
Files\Microsoft Shared\ClickToRun\OfficeC2RClient.exe</Image> <!--
Microsoft:Office: Background process-->
                    <ParentImage condition="is">C:\Program
Files\Common Files\Microsoft
Shared\ClickToRun\OfficeClickToRun.exe</ParentImage> <!--
Microsoft:Office: Background process-->
                    <ParentImage condition="is">C:\Program
Files\Common Files\Microsoft
Shared\ClickToRun\OfficeC2RClient.exe</ParentImage> <!--
Microsoft:Office: Background process-->
                    <!--SECTION: Windows: Media player-->
                    <Image condition="is">C:\Program Files\Windows
Media Player\wmpnscfg.exe</Image> <!--Windows: Windows Media Player
Network Sharing Service Configuration Application-->
                    <!--SECTION: Google-->
                    <CommandLine condition="begin with">"C:\Program
Files (x86)\Google\Chrome\Application\chrome.exe" --type=</CommandLine>
<!--Google:Chrome: massive command-line arguments-->
                    <CommandLine condition="begin with">"C:\Program
Files\Google\Chrome\Application\chrome.exe" --type=</CommandLine> <!--
Google:Chrome: massive command-line arguments-->
                </ProcessCreate>
        </RuleGroup>

        <!--SYSMON EVENT ID 2 : FILE CREATION TIME RETROACTIVELY CHANGED
IN THE FILESYSTEM [FileCreateTime]-->
                <!--COMMENT:   [
https://attack.mitre.org/wiki/Technique/T1099 ] -->
```

```
		<!--DATA: UtcTime, ProcessGuid, ProcessId, Image,
TargetFilename, CreationUtcTime, PreviousCreationUtcTime-->
	<RuleGroup name="" groupRelation="or">
		<FileCreateTime onmatch="include">
			<Image name="T1099" condition="begin
with">C:\Users</Image> <!--Look for timestomping in user area, usually
nothing should be doing that here-->
			<TargetFilename name="T1099" condition="end
with">.exe</TargetFilename> <!--Look for backdated executables
anywhere-->
			<Image name="T1099" condition="begin
with">\Device\HarddiskVolumeShadowCopy</Image> <!--Nothing should be
written here | Credit: @SBousseaden [
https://twitter.com/SBousseaden/status/1133030955407630336 ] -->
		</FileCreateTime>
	</RuleGroup>

	<RuleGroup name="" groupRelation="or">
		<FileCreateTime onmatch="exclude">
			<Image condition="image">OneDrive.exe</Image> <!--
OneDrive constantly changes file times-->
			<Image
condition="image">C:\Windows\system32\backgroundTaskHost.exe</Image>
			<Image condition="contains">setup</Image> <!--
Ignore setups-->
			<Image condition="contains">install</Image> <!--
Ignore setups-->
```

```
                    <Image condition="contains">Update\</Image> <!--
Ignore setups-->
                    <Image condition="end with">redist.exe</Image> <!
-Ignore setups-->
                    <Image condition="is">msiexec.exe</Image> <!--
Ignore setups-->
                    <Image condition="is">TrustedInstaller.exe</Image>
<!--Ignore setups-->
                    <TargetFilename
condition="contains">\NVIDIA\NvBackend\ApplicationOntology\</TargetFile
name> <!--NVIDIA GeForce Experience Application Ontology, 1000's of
events in user profile-->
            </FileCreateTime>
        </RuleGroup>

        <!--SYSMON EVENT ID 3 : NETWORK CONNECTION INITIATED
[NetworkConnect]-->
            <!--COMMENT:   By default this configuration takes a very
conservative approach to network logging, limited to only extremely
high-signal events.-->
            <!--COMMENT:   [
https://attack.mitre.org/wiki/Command_and_Control ] [
https://attack.mitre.org/wiki/Exfiltration ] [
https://attack.mitre.org/wiki/Lateral_Movement ] -->
            <!--TECHNICAL: For the DestinationHostname, Sysmon uses
the GetNameInfo API, which will often not have any information, and may
just be a CDN. This is NOT reliable for filtering.-->
            <!--TECHNICAL: For the DestinationPortName, Sysmon uses
the GetNameInfo API for the friendly name of ports you see in logs.-->
```

```
		<!--TECHNICAL: These exe do not initiate their
connections, and thus includes do not work in this section: BITSADMIN
NLTEST-->

		<!-- https://www.first.org/resources/papers/conf2017/APT-
Log-Analysis-Tracking-Attack-Tools-by-Audit-Policy-and-Sysmon.pdf -->

		<!--DATA: UtcTime, ProcessGuid, ProcessId, Image, User,
Protocol, Initiated, SourceIsIpv6, SourceIp, SourceHostname,
SourcePort, SourcePortName, DestinationIsIpV6, DestinationIp,
DestinationHostname, DestinationPort, DestinationPortName-->
	<RuleGroup name="" groupRelation="or">
		<NetworkConnect onmatch="include">
			<!--Suspicious sources for network-connecting
binaries-->
			<Image name="Usermode" condition="begin
with">C:\Users</Image> <!--Tools downloaded by users can use other
processes for networking, but this is a very valuable indicator.-->
			<Image name="Caution" condition="begin
with">C:\Recycle</Image> <!--Nothing should operate from the RecycleBin
locations.-->
			<Image condition="begin
with">C:\ProgramData</Image> <!--Normally, network communications
should be sourced from "Program Files" not from ProgramData, something
to look at-->
			<Image condition="begin
with">C:\Windows\Temp</Image> <!--Suspicious anything would communicate
from the system-level temp directory-->
```

```
                    <Image name="Caution" condition="begin
with">\</Image> <!--Devices and VSC shouldn't be executing changes |
Credit: @SBousseaden @ionstorm @neu5ron @PerchedSystems [
https://twitter.com/SwiftOnSecurity/status/1133167323991486464 ] -->
                    <Image name="Caution" condition="begin
with">C:\perflogs</Image> <!-- Credit @blu3_team [ https://blu3-
team.blogspot.com/2019/05/netconn-from-suspicious-directories.html ] --
>
                    <Image name="Caution" condition="begin
with">C:\intel</Image> <!-- Credit @blu3_team [ https://blu3-
team.blogspot.com/2019/05/netconn-from-suspicious-directories.html ] --
>
                    <Image name="Caution" condition="begin
with">C:\Windows\fonts</Image> <!-- Credit @blu3_team [ https://blu3-
team.blogspot.com/2019/05/netconn-from-suspicious-directories.html ] --
>
                    <Image name="Caution" condition="begin
with">C:\Windows\system32\config</Image> <!-- Credit @blu3_team [
https://blu3-team.blogspot.com/2019/05/netconn-from-suspicious-
directories.html ] -->
                    <!--Suspicious Windows tools-->
                    <Image condition="image">at.exe</Image> <!--
Windows: Remote task scheduling, removed in Win10 | Credit @ion-storm -
->
                    <Image condition="image">certutil.exe</Image> <!--
Windows: Certificate tool can contact outbound | Credit @ion-storm @FVT
[ https://twitter.com/FVT/status/834433734602530817 ] -->
                    <Image condition="image">cmd.exe</Image> <!--
Windows: Remote command prompt-->
```

```
                    <Image condition="image">cmstp.exe</Image> <!--
Windows: Connection manager profiles can launch executables from WebDAV
[ https://twitter.com/NickTyrer/status/958450014111633408 ] | Credit
@NickTyrer @Oddvarmoe @KyleHanslovan @subTee -->
                    <Image condition="image">cscript.exe</Image> <!--
WindowsScriptingHost: | Credit @Cyb3rOps [
https://gist.github.com/Neo23x0/a4b4af9481e01e749409 ] -->
                    <Image condition="image">driverquery.exe</Image>
<!--Windows: Remote recognisance of system configuration,
oudated/vulnerable drivers -->
                    <Image condition="image">dsquery.exe</Image> <!--
Microsoft: Query Active Directory -->
                    <Image condition="image">hh.exe</Image> <!--
Windows: HTML Help Executable, opens CHM files -->
                    <Image
condition="image">infDefaultInstall.exe</Image> <!--Microsoft: [
https://github.com/huntresslabs/evading-autoruns ] | Credit
@KyleHanslovan -->
                    <Image condition="image">java.exe</Image> <!--
Java: Monitor usage of vulnerable application and init from JAR files |
Credit @ion-storm -->
                    <Image condition="image">javaw.exe</Image> <!--
Java: Monitor usage of vulnerable application and init from JAR files -
->
                    <Image condition="image">javaws.exe</Image> <!--
Java: Monitor usage of vulnerable application and init from JAR files -
->
                    <Image condition="image">mmc.exe</Image> <!--
Windows: -->
```

```
                    <Image condition="image">msbuild.exe</Image> <!--
Windows: [ https://www.hybrid-
analysis.com/sample/a314f6106633fba4b70f9d6ddbee452e8f8f44a72117749c212
43dc93c7ed3ac?environmentId=100 ] -->
                    <Image condition="image">mshta.exe</Image> <!--
Windows: HTML application executes scripts without IE protections |
Credit @ion-storm [ https://en.wikipedia.org/wiki/HTML_Application ] --
>
                    <Image condition="image">msiexec.exe</Image> <!--
Windows: Can install from http:// paths | Credit @vector-sec -->
                    <Image condition="image">nbtstat.exe</Image> <!--
Windows: NetBIOS statistics, attackers use to enumerate local network -
->
                    <Image condition="image">net.exe</Image> <!--
Windows: Note - May not detect anything, net.exe is a front-end to
lower APIs | Credit @ion-storm -->
                    <Image condition="image">net1.exe</Image> <!--
Windows: Launched by "net.exe", but it may not detect connections
either -->
                    <Image condition="image">notepad.exe</Image> <!--
Windows: [ https://secrary.com/ReversingMalware/CoinMiner/ ] [
https://blog.cobaltstrike.com/2013/08/08/why-is-notepad-exe-connecting-
to-the-internet/ ] -->
                    <Image condition="image">nslookup.exe</Image> <!--
Windows: Retrieve data over DNS -->
                    <Image condition="image">powershell.exe</Image>
<!--Windows: PowerShell interface-->
```

```
                    <Image
condition="image">powershell_ise.exe</Image> <!--Windows: PowerShell
interface-->
                    <Image condition="image">qprocess.exe</Image> <!--
Windows: [ https://www.first.org/resources/papers/conf2017/APT-Log-
Analysis-Tracking-Attack-Tools-by-Audit-Policy-and-Sysmon.pdf ] -->
                    <Image condition="image">qwinsta.exe</Image> <!--
Windows: Query remote sessions | Credit @ion-storm -->
                    <Image condition="image">qwinsta.exe</Image> <!--
Windows: Remotely query login sessions on a server or workstation |
Credit @ion-storm -->
                    <Image condition="image">reg.exe</Image> <!--
Windows: Remote Registry editing ability | Credit @ion-storm -->
                    <Image condition="image">regsvcs.exe</Image> <!--
Windows: [ https://www.hybrid-
analysis.com/sample/3f94d7080e6c5b8f59eeecc3d44f7e817b31562caeba21d02ad
705a0bfc63d67?environmentId=100 ] -->
                    <Image condition="image">regsvr32.exe</Image> <!--
Windows: [ https://subt0x10.blogspot.com/2016/04/bypass-application-
whitelisting-script.html ] -->
                    <Image condition="image">rundll32.exe</Image> <!--
Windows: [ https://blog.cobaltstrike.com/2016/07/22/why-is-rundll32-
exe-connecting-to-the-internet/ ] -->
                    <Image condition="image">rwinsta.exe</Image> <!--
Windows: Disconnect remote sessions | Credit @ion-storm -->
                    <Image condition="image">sc.exe</Image> <!--
Windows: Remotely change Windows service settings | Credit @ion-storm -
->
```

```
                    <Image condition="image">schtasks.exe</Image> <!--
Windows: Command-line interface to local and remote tasks -->
                    <Image condition="image">taskkill.exe</Image> <!--
Windows: Kill processes, has remote ability -->
                    <Image condition="image">tasklist.exe</Image> <!--
Windows: List processes, has remote ability -->
                    <Image condition="image">wmic.exe</Image> <!--
WindowsManagementInstrumentation: Credit @Cyb3rOps [
https://gist.github.com/Neo23x0/a4b4af9481e01e749409 ] -->
                    <Image condition="image">wscript.exe</Image> <!--
WindowsScriptingHost: | Credit @arekfurt -->
                    <!--Live of the Land Binaries and scripts (LOLBAS)
-->
                    <Image condition="image">bitsadmin.exe</Image> <!-
- Windows: Background Intelligent Transfer Service - Can download from
URLs -->
                    <Image condition="image">esentutl.exe</Image> <!--
Windows: Database utilities for the ESE - Can fetch from UNC paths -->
                    <Image condition="image">expand.exe</Image> <!--
Windows: Expands one or more compressed files - Can fetch from UNC
paths -->
                    <Image condition="image">extrac32.exe</Image> <!--
Windows: Uncompress .cab files - Can fetch from UNC paths -->
                    <Image condition="image">findstr.exe</Image> <!--
Windows: Search for strings - Can fetch from UNC paths -->
                    <Image
condition="image">GfxDownloadWrapper.exe</Image> <!-- Intel Graphics
Control Panel: Remote file download -->
```

```
                    <Image condition="image">ieexec.exe</Image> <!--
Windows: Microsoft .NET Framework application - Download and execute
from URLs -->
                    <Image condition="image">makecab.exe</Image> <!--
Windows: Packages existing files into a .cab - Can fetch from UNC paths
-->
                    <Image condition="image">replace.exe</Image> <!--
Windows: Used to replace file with another file - Can fetch from UNC
paths -->
                    <Image condition="image">Excel.exe</Image> <!--
Windows Office: Excel - Can download from URLs -->
                    <Image condition="image">Powerpnt.exe</Image> <!--
Windows Office: PowerPoint - Can download from URLs -->
                    <Image condition="image">Winword.exe</Image> <!--
Windows Office: Word - Can download from URLs -->
                    <Image condition="image">squirrel.exe</Image> <!--
Windows: Update the Nuget/Squirrel packages. Part of Teams. - Can
download from URLs -->
                    <!--Relevant 3rd Party Tools-->
                    <Image condition="image">nc.exe</Image> <!--
Nmap's modern version of netcat [
https://nmap.org/ncat/guide/index.html#ncat-overview ] [
https://securityblog.gr/1517/create-backdoor-in-windows-with-ncat/ ] --
>
                    <Image condition="image">ncat.exe</Image> <!--
Nmap's modern version of netcat [
https://nmap.org/ncat/guide/index.html#ncat-overview ] [
https://securityblog.gr/1517/create-backdoor-in-windows-with-ncat/ ] --
>
```

```
                    <Image condition="image">psexec.exe</Image> <!--
Sysinternals:PsExec client side | Credit @Cyb3rOps -->
                    <Image condition="image">psexesvc.exe</Image> <!--
Sysinternals:PsExec server side | Credit @Cyb3rOps -->
                    <Image condition="image">tor.exe</Image> <!--Tor [
https://www.hybrid-
analysis.com/sample/800bf028a23440134fc834efc5c1e02cc70f05b2e800bbc285d
7c92a4b126b1c?environmentId=100 ] -->
                    <Image condition="image">vnc.exe</Image> <!-- VNC
client | Credit @Cyb3rOps -->
                    <Image condition="image">vncservice.exe</Image>
<!-- VNC server | Credit @Cyb3rOps -->
                    <Image condition="image">vncviewer.exe</Image> <!-
- VNC client | Credit @Cyb3rOps -->
                    <Image condition="image">winexesvc.exe</Image> <!-
- Winexe service executable | Credit @Cyb3rOps -->
                    <Image condition="image">nmap.exe</Image>
                    <Image condition="image">psinfo.exe</Image>
                    <!--Ports: Suspicious-->
                    <DestinationPort name="SSH"
condition="is">22</DestinationPort> <!--SSH protocol, monitor admin
connections-->
                    <DestinationPort name="Telnet"
condition="is">23</DestinationPort> <!--Telnet protocol, monitor admin
connections, insecure-->
                    <DestinationPort name="SMTP"
condition="is">25</DestinationPort> <!--SMTP mail protocol port,
insecure, used by threats-->
```

```
                    <DestinationPort name="IMAP"
condition="is">143</DestinationPort> <!--IMAP mail protocol port,
insecure, used by threats-->
                    <DestinationPort name="RDP"
condition="is">3389</DestinationPort> <!--Windows:RDP: Monitor admin
connections-->
                    <DestinationPort name="VNC"
condition="is">5800</DestinationPort> <!--VNC protocol: Monitor admin
connections, often insecure, using hard-coded admin password-->
                    <DestinationPort name="VNC"
condition="is">5900</DestinationPort> <!--VNC protocol Monitor admin
connections, often insecure, using hard-coded admin password-->
                    <DestinationPort name="Alert,Metasploit"
condition="is">4444</DestinationPort>
                    <!--Ports: Proxy-->
                    <DestinationPort name="Proxy"
condition="is">1080</DestinationPort> <!--Socks proxy port | Credit
@ion-storm-->
                    <DestinationPort name="Proxy"
condition="is">3128</DestinationPort> <!--Socks proxy port | Credit
@ion-storm-->
                    <DestinationPort name="Proxy"
condition="is">8080</DestinationPort> <!--Socks proxy port | Credit
@ion-storm-->
                    <!--Ports: Tor-->
                    <DestinationPort name="Tor"
condition="is">1723</DestinationPort> <!--Tor protocol [
https://attack.mitre.org/wiki/Technique/T1090 ] | Credit @ion-storm-->
```

```
                    <DestinationPort name="Tor"
condition="is">9001</DestinationPort> <!--Tor protocol [
http://www.computerworlduk.com/tutorial/security/tor-enterprise-2016-
blocking-malware-darknet-use-rogue-nodes-3633907/ ] -->
                    <DestinationPort name="Tor"
condition="is">9030</DestinationPort> <!--Tor protocol [
http://www.computerworlduk.com/tutorial/security/tor-enterprise-2016-
blocking-malware-darknet-use-rogue-nodes-3633907/ ] -->
                </NetworkConnect>
        </RuleGroup>

        <RuleGroup name="" groupRelation="or">
                <NetworkConnect onmatch="exclude">
                    <!--SECTION: Microsoft-->
                    <Image condition="begin
with">C:\ProgramData\Microsoft\Windows Defender\Platform\</Image>
                    <Image condition="end
with">AppData\Local\Microsoft\Teams\current\Teams.exe</Image> <!--
Microsoft: Teams-->
                    <DestinationHostname condition="end
with">.microsoft.com</DestinationHostname> <!--Microsoft:Update
delivery-->
                    <DestinationHostname condition="end
with">microsoft.com.akadns.net</DestinationHostname> <!--
Microsoft:Update delivery-->
                    <DestinationHostname condition="end
with">microsoft.com.nsatc.net</DestinationHostname> <!--
Microsoft:Update delivery-->
                    <!--OCSP known addresses-->
```

```
                    <DestinationIp
condition="is">23.4.43.27</DestinationIp> <!--Digicert [
https://otx.alienvault.com/indicator/ip/23.4.43.27 ] -->
                    <DestinationIp
condition="is">72.21.91.29</DestinationIp> <!--Digicert [
https://otx.alienvault.com/indicator/ip/72.21.91.29 ] -->
                    <!--Section: Loopback Addresses-->
                    <DestinationIp
condition="is">127.0.0.1</DestinationIp> <!--Credit @ITProPaul-->
                    <DestinationIp condition="begin
with">fe80:0:0:0</DestinationIp> <!--Credit @ITProPaul-->
             </NetworkConnect>
      </RuleGroup>

      <!--SYSMON EVENT ID 4 : RESERVED FOR SYSMON SERVICE STATUS
MESSAGES-->

             <!--DATA: UtcTime, State, Version, SchemaVersion-->
             <!--Cannot be filtered.-->

      <!--SYSMON EVENT ID 5 : PROCESS ENDED [ProcessTerminate]-->
             <!--COMMENT:  Useful data in building infection
timelines.-->

             <!--DATA: UtcTime, ProcessGuid, ProcessId, Image-->
      <RuleGroup name="" groupRelation="or">
             <ProcessTerminate onmatch="include">
                    <Image condition="begin with">C:\Users</Image> <!-
-Process terminations by user binaries-->
```

```
                        <Image condition="begin with">\</Image> <!--
Devices and VSC shouldn't be executing changes | Credit: @SBousseaden
@ionstorm @neu5ron @PerchedSystems [
https://twitter.com/SwiftOnSecurity/status/1133167323991486464 ] -->
                </ProcessTerminate>
        </RuleGroup>

        <RuleGroup name="" groupRelation="or">
                <ProcessTerminate onmatch="exclude">
                </ProcessTerminate>
        </RuleGroup>

        <!--SYSMON EVENT ID 6 : DRIVER LOADED INTO KERNEL [DriverLoad]--
>
                <!--COMMENT:   Because drivers with bugs can be used to
escalate to kernel permissions, be extremely selective
                        about what you exclude from monitoring. Low event
volume, little incentive to exclude.
                        [ https://attack.mitre.org/wiki/Technique/T1014 ]
-->
                <!--TECHNICAL: Sysmon will check the signing certificate
revocation status of any driver you don't exclude.-->

                <!--DATA: UtcTime, ImageLoaded, Hashes, Signed,
Signature, SignatureStatus-->
        <RuleGroup name="" groupRelation="or">
                <DriverLoad onmatch="exclude">
```

```
                        <Signature
condition="contains">microsoft</Signature> <!--Exclude signed Microsoft
drivers-->
                        <Signature
condition="contains">windows</Signature> <!--Exclude signed Microsoft
drivers-->
                        <Signature condition="begin with">Intel
</Signature> <!--Exclude signed Intel drivers-->
                </DriverLoad>
        </RuleGroup>

        <!--SYSMON EVENT ID 7 : DLL (IMAGE) LOADED BY PROCESS
[ImageLoad]-->
                <!--COMMENT:   Can cause high system load, disabled by
default.-->
                <!--COMMENT:   [
https://attack.mitre.org/wiki/Technique/T1073 ] [
https://attack.mitre.org/wiki/Technique/T1038 ] [
https://attack.mitre.org/wiki/Technique/T1034 ] -->

                <!--DATA: UtcTime, ProcessGuid, ProcessId, Image,
ImageLoaded, Hashes, Signed, Signature, SignatureStatus-->
        <RuleGroup name="" groupRelation="or">
                <ImageLoad onmatch="include">
                        <!--NOTE: Using "include" with no rules means
nothing in this section will be logged-->
                </ImageLoad>
        </RuleGroup>
```

```
        <!--SYSMON EVENT ID 8 : REMOTE THREAD CREATED
[CreateRemoteThread]-->
                <!--COMMENT:  Monitor for processes injecting code into
other processes. Often used by malware to cloak their actions. Also
when Firefox loads Flash.
                [ https://attack.mitre.org/wiki/Technique/T1055 ] -->

                <!--DATA: UtcTime, SourceProcessGuid, SourceProcessId,
SourceImage, TargetProcessId, TargetImage, NewThreadId, StartAddress,
StartModule, StartFunction-->
        <RuleGroup name="" groupRelation="or">
                <CreateRemoteThread onmatch="exclude">
                        <!--COMMENT: Exclude mostly-safe sources and log
anything else.-->
                        <SourceImage
condition="is">C:\Windows\system32\wbem\WmiPrvSE.exe</SourceImage>
                        <SourceImage
condition="is">C:\Windows\system32\svchost.exe</SourceImage>
                        <SourceImage
condition="is">C:\Windows\system32\wininit.exe</SourceImage>
                        <SourceImage
condition="is">C:\Windows\system32\csrss.exe</SourceImage>
                        <SourceImage
condition="is">C:\Windows\system32\services.exe</SourceImage>
                        <SourceImage
condition="is">C:\Windows\system32\winlogon.exe</SourceImage>
                        <SourceImage
condition="is">C:\Windows\system32\audiodg.exe</SourceImage>
```

```
                    <StartModule
condition="is">C:\Windows\system32\kernel32.dll</StartModule>
                    <TargetImage condition="is">C:\Program Files
(x86)\Google\Chrome\Application\chrome.exe</TargetImage>
            </CreateRemoteThread>
        </RuleGroup>

        <!--SYSMON EVENT ID 9 : RAW DISK ACCESS [RawAccessRead]-->
            <!--EVENT 9: "RawAccessRead detected"-->
            <!--COMMENT:  Can cause high system load, disabled by
default.-->
            <!--COMMENT:  Monitor for raw sector-level access to the
disk, often used to bypass access control lists or access locked files.
                    Disabled by default since including even one entry
here activates this component. Reward/performance/rule maintenance
decision.
                    Encourage you to experiment with this feature
yourself. [ https://attack.mitre.org/wiki/Technique/T1067 ] -->
            <!--COMMENT:  You will likely want to set this to a full
capture on domain controllers, where no process should be doing raw
reads.-->

            <!--DATA: UtcTime, ProcessGuid, ProcessId, Image, Device-
->
        <RuleGroup name="" groupRelation="or">
            <RawAccessRead onmatch="include">
                <!--NOTE: Using "include" with no rules means
nothing in this section will be logged-->
            </RawAccessRead>
```

```
	</RuleGroup>

	<!--SYSMON EVENT ID 10 : INTER-PROCESS ACCESS [ProcessAccess]-->
		<!--EVENT 10: "Process accessed"-->
		<!--COMMENT:  Can cause high system load, disabled by default.-->
		<!--COMMENT:  Monitor for processes accessing other process' memory.-->

		<!--DATA: UtcTime, SourceProcessGuid, SourceProcessId, SourceThreadId, SourceImage, TargetProcessGuid, TargetProcessId, TargetImage, GrantedAccess, CallTrace-->
	<RuleGroup name="" groupRelation="or">
		<ProcessAccess onmatch="include">
			<!--NOTE: Using "include" with no rules means nothing in this section will be logged-->
		</ProcessAccess>
	</RuleGroup>

	<!--SYSMON EVENT ID 11 : FILE CREATED [FileCreate]-->
		<!--EVENT 11: "File created"-->
		<!--NOTE:     Other filesystem "minifilters" can make it appear to Sysmon that some files are being written twice. This is not a Sysmon issue, per Mark Russinovich.-->
		<!--NOTE:     You may not see files detected by antivirus. Other filesystem minifilters, like antivirus, can act before Sysmon receives the alert a file was written.-->
```

```
                <!--DATA: UtcTime, ProcessGuid, ProcessId, Image,
TargetFilename, CreationUtcTime-->
        <RuleGroup name="" groupRelation="or">
                <FileCreate onmatch="include">
                        <TargetFilename name="T1023"
condition="contains">\Start Menu</TargetFilename> <!--Windows: Startup
links and shortcut modification [
https://attack.mitre.org/wiki/Technique/T1023 ] -->
                        <TargetFilename name="T1165"
condition="contains">\Startup\</TargetFilename> <!--Microsoft:Changes
to user's auto-launched files and shortcuts-->
                        <TargetFilename name="OutlookAttachment"
condition="contains">\Content.Outlook\</TargetFilename> <!--
Microsoft:Outlook: attachments-->
                        <TargetFilename name="Downloads"
condition="contains">\Downloads\</TargetFilename> <!--Downloaded files.
Does not include "Run" files in IE-->
                        <TargetFilename condition="end
with">.application</TargetFilename> <!--Microsoft:ClickOnce: [
https://blog.netspi.com/all-you-need-is-one-a-clickonce-love-story/ ] -
->
                        <TargetFilename condition="end with">.appref-
ms</TargetFilename> <!--Microsoft:ClickOnce application | Credit @ion-
storm -->
                        <TargetFilename condition="end
with">.bat</TargetFilename> <!--Batch scripting-->
                        <TargetFilename condition="end
with">.chm</TargetFilename>
```

```
                    <TargetFilename condition="end
with">.cmd</TargetFilename> <!--Batch scripting: Batch scripts can also
use the .cmd extension | Credit: @mmazanec -->
                    <TargetFilename condition="end
with">.cmdline</TargetFilename> <!--Microsoft:dotNet: Executed by
cvtres.exe-->
                    <TargetFilename name="T1176" condition="end
with">.crx</TargetFilename> <!--Chrome extension-->
                    <TargetFilename condition="end
with">.dmp</TargetFilename> <!--Process dumps [ (fr)
http://blog.gentilkiwi.com/securite/mimikatz/minidump ] -->
                    <TargetFilename condition="end
with">.docm</TargetFilename> <!--Microsoft:Office:Word: Macro-->
                    <TargetFilename name="DLL" condition="end
with">.dll</TargetFilename> <!--Microsoft:Office:Word: Macro-->
                    <TargetFilename name="EXE" condition="end
with">.exe</TargetFilename> <!--Executable-->
                    <TargetFilename name="ProcessHostingdotNETCode"
condition="end with">.exe.log</TargetFilename> <!-- [
https://github.com/bitsadmin/nopowershell ] | Credit: @SBousseaden [
https://twitter.com/SBousseaden/status/1137493597769687040 ]  -->
                    <TargetFilename condition="end
with">.jar</TargetFilename> <!--Java applets-->
                    <TargetFilename condition="end
with">.jnlp</TargetFilename> <!--Java applets-->
                    <TargetFilename condition="end
with">.jse</TargetFilename> <!--Scripting [ Example:
https://www.sophos.com/en-us/threat-center/threat-analyses/viruses-and-
spyware/Mal~Phires-C/detailed-analysis.aspx ] -->
```

```
                    <TargetFilename condition="end
with">.hta</TargetFilename> <!--Scripting-->
                    <TargetFilename condition="end
with">.job</TargetFilename> <!--Scheduled task-->
                    <TargetFilename condition="end
with">.pptm</TargetFilename> <!--Microsoft:Office:Word: Macro-->
                    <TargetFilename condition="end
with">.ps1</TargetFilename> <!--PowerShell [ More information:
http://www.hexacorn.com/blog/2014/08/27/beyond-good-ol-run-key-part-16/
] -->
                    <TargetFilename condition="end
with">.sct</TargetFilename> <!--Scripting | Credit @bartblaze -->
                    <TargetFilename condition="end
with">.sys</TargetFilename> <!--System driver files-->
                    <TargetFilename condition="end
with">.scr</TargetFilename> <!--System driver files-->
                    <TargetFilename condition="end
with">.vbe</TargetFilename> <!--VisualBasicScripting-->
                    <TargetFilename condition="end
with">.vbs</TargetFilename> <!--VisualBasicScripting-->
                    <TargetFilename condition="end
with">.wsc</TargetFilename> <!--Scripting | Credit @bartblaze -->
                    <TargetFilename condition="end
with">.wsf</TargetFilename> <!--Scripting | Credit @bartblaze -->
                    <TargetFilename condition="end
with">.xlsm</TargetFilename> <!--Microsoft:Office:Word: Macro-->
                    <TargetFilename condition="end
with">.ocx</TargetFilename> <!--Microsoft:ActiveX-->
```

```
                    <TargetFilename condition="end
with">proj</TargetFilename><!--Microsoft:MSBuild:Script: [
https://twitter.com/subTee/status/885919612969394177 ] -->
                    <TargetFilename condition="end
with">.sln</TargetFilename><!--Microsoft:MSBuild:Script: [
https://twitter.com/subTee/status/885919612969394177 ] -->
                    <TargetFilename condition="end
with">.xls</TargetFilename><!--Microsoft [
https://medium.com/@threathuntingteam/msxsl-exe-and-wmic-exe-a-way-to-
proxy-code-execution-8d524f642b75 ] -->
                    <TargetFilename name="DefaultUserModified"
condition="begin with">C:\Users\Default</TargetFilename> <!--Windows:
Changes to default user profile-->
                    <TargetFilename condition="begin
with">C:\Windows\system32\Drivers</TargetFilename> <!--Microsoft:
Drivers dropped here-->
                    <TargetFilename condition="begin
with">C:\Windows\SysWOW64\Drivers</TargetFilename> <!--Microsoft:
Drivers dropped here-->
                    <TargetFilename name="T1037,T1484"
condition="begin
with">C:\Windows\system32\GroupPolicy\Machine\Scripts</TargetFilename>
<!--Group policy [ More information:
http://www.hexacorn.com/blog/2017/01/07/beyond-good-ol-run-key-part-52/
] -->
                    <TargetFilename name="T1037,T1484"
condition="begin
with">C:\Windows\system32\GroupPolicy\User\Scripts</TargetFilename> <!-
-Group policy [ More information:
```

```
http://www.hexacorn.com/blog/2017/01/07/beyond-good-ol-run-key-part-52/
] -->
                    <TargetFilename condition="begin
with">C:\Windows\system32\Wbem</TargetFilename> <!--Microsoft:WMI: [
More information:
http://2014.hackitoergosum.org/slides/day1_WMI_Shell_Andrei_Dumitrescu.
pdf ] -->
                    <TargetFilename condition="begin
with">C:\Windows\SysWOW64\Wbem</TargetFilename> <!--Microsoft:WMI: [
More information:
http://2014.hackitoergosum.org/slides/day1_WMI_Shell_Andrei_Dumitrescu.
pdf ] -->
                    <TargetFilename condition="begin
with">C:\Windows\system32\WindowsPowerShell</TargetFilename> <!--
Microsoft:Powershell: Look for modifications for persistence [
https://www.malwarearchaeology.com/cheat-sheets ] -->
                    <TargetFilename condition="begin
with">C:\Windows\SysWOW64\WindowsPowerShell</TargetFilename> <!--
Microsoft:Powershell: Look for modifications for persistence [
https://www.malwarearchaeology.com/cheat-sheets ] -->
                    <TargetFilename name="T1053" condition="begin
with">C:\Windows\Tasks\</TargetFilename> <!--Microsoft:ScheduledTasks [
https://attack.mitre.org/wiki/Technique/T1053 ] -->
                    <TargetFilename name="T1053" condition="begin
with">C:\Windows\system32\Tasks</TargetFilename> <!--
Microsoft:ScheduledTasks [
https://attack.mitre.org/wiki/Technique/T1053 ] -->
                    <TargetFilename name="T1053" condition="begin
with">C:\Windows\SysWOW64\Tasks</TargetFilename> <!--
```

```
Microsoft:ScheduledTasks [
https://attack.mitre.org/wiki/Technique/T1053 ] -->
                    <Image condition="begin
with">\Device\HarddiskVolumeShadowCopy</Image> <!--Nothing should be
executing from VSC | Credit: @SBousseaden [
https://twitter.com/SBousseaden/status/1133030955407630336 ] -->
                    <!--Windows application compatibility-->
                    <TargetFilename condition="begin
with">C:\Windows\AppPatch\Custom</TargetFilename> <!--Windows:
Application compatibility shims [ https://www.fireeye.com/blog/threat-
research/2017/05/fin7-shim-databases-persistence.html ] -->
                    <TargetFilename
condition="contains">VirtualStore</TargetFilename> <!--Windows: UAC
virtualization [ https://blogs.msdn.microsoft.com/oldnewthing/20150902-
00/?p=91681 ] -->
                    <!--Exploitable file names-->
                    <TargetFilename condition="end
with">.xls</TargetFilename> <!--Legacy Office files are often used for
attacks-->
                    <TargetFilename condition="end
with">.ppt</TargetFilename> <!--Legacy Office files are often used for
attacks-->
                    <TargetFilename condition="end
with">.rtf</TargetFilename> <!--RTF files often 0day malware vectors
when opened by Office-->
                </FileCreate>
        </RuleGroup>

        <RuleGroup name="" groupRelation="or">
```

```
                <FileCreate onmatch="exclude">
                        <!--SECTION: Microsoft-->
                        <Image condition="is">C:\Program Files (x86)\EMET
5.5\EMET_Service.exe</Image> <!--Microsoft:EMET: Writes to
C:\Windows\AppPatch\-->
                        <!--SECTION: Microsoft:Office:Click2Run-->
                        <Image condition="is">C:\Program Files\Common
Files\Microsoft Shared\ClickToRun\OfficeC2RClient.exe</Image> <!--
Microsoft:Office Click2Run-->
                        <!--SECTION: Windows-->
                        <Image
condition="is">C:\Windows\system32\smss.exe</Image> <!-- Windows:
Session Manager SubSystem: Creates
swapfile.sys,pagefile.sys,hiberfile.sys-->
                        <Image
condition="is">C:\Windows\system32\CompatTelRunner.exe</Image> <!--
Windows: Windows 10 app, creates tons of cache files-->
                        <Image
condition="is">\\?\C:\Windows\system32\wbem\WMIADAP.EXE</Image> <!--
Windows: WMI Performance updates-->
                        <Image
condition="is">C:\Windows\system32\mobsync.exe</Image> <!--Windows:
Network file syncing-->
                        <TargetFilename condition="begin
with">C:\Windows\system32\DriverStore\Temp\</TargetFilename> <!--
Windows: Temp files by DrvInst.exe-->
                        <TargetFilename condition="begin
with">C:\Windows\system32\wbem\Performance\</TargetFilename> <!--
Windows: Created in wbem by WMIADAP.exe-->
```

```
                    <TargetFilename condition="begin
with">C:\Windows\Installer\</TargetFilename> <!--Windows:Installer:
Ignore MSI installer files caching-->
                    <!--SECTION: Windows:Updates-->
                    <TargetFilename condition="begin
with">C:\$WINDOWS.~BT\Sources\</TargetFilename> <!-- Windows: Feature
updates containing lots of .exe and .sys-->
                    <Image condition="begin
with">C:\Windows\winsxs\amd64_microsoft-windows</Image> <!-- Windows:
Windows update-->
            </FileCreate>
        </RuleGroup>

        <!--SYSMON EVENT ID 12 & 13 & 14 : REGISTRY MODIFICATION
[RegistryEvent]-->
            <!--EVENT 12: "Registry object added or deleted"-->
            <!--EVENT 13: "Registry value set"-->
            <!--EVENT 14: "Registry objected renamed"-->

            <!--NOTE:     Windows writes hundreds or thousands of
registry keys a minute, so just because you're not changing things,
doesn't mean these rules aren't being run.-->
            <!--NOTE:     You do not have to spend a lot of time
worrying about performance, CPUs are fast, but it's something to
consider. Every rule and condition type has a small cost.-->
            <!--NOTE:     "contains" works by finding the first
letter, then matching the second, etc, so the first letters should be
as low-occurrence as possible.-->
```

```
                <!--NOTE:      [
https://attack.mitre.org/wiki/Technique/T1112 ] -->

                <!--TECHNICAL: You cannot filter on the "Details"
attribute, due to performance issues when very large keys are written,
and variety of data formats-->
                <!--TECHNICAL: Possible prefixes are HKLM, HKCR, and HKU-
->
                <!--CRITICAL:  Schema version 3.30 and higher change
HKLM\="\REGISTRY\MACHINE\" and HKU\="\REGISTRY\USER\" and
HKCR\="\REGISTRY\MACHINE\SOFTWARE\Classes\" and
CurrentControlSet="ControlSet001"-->
                <!--CRITICAL:  Due to a bug, Sysmon versions BEFORE 7.01
may not properly log with the new prefix style for registry keys that
was originally introduced in schema version 3.30-->
                <!--NOTE:      Because Sysmon runs as a service, it has
no filtering ability for, or concept of, HKCU or HKEY_CURRENT_USER. Use
"contains" or "end with" to get around this limitation-->

                <!-- ! CRITICAL NOTE !:      It may appear this section
is MISSING important entries, but SOME RULES MONITOR MANY KEYS, so look
VERY CAREFULLY to see if something is already covered.
                                                                Sysmon's
wildcard monitoring along with highly-tuned generic strings cuts the
rulesets down immensely, compared to doing this in other tools.
                                                                For example,
most COM hijacking in CLSID's across the registry is covered by a
single rule monitoring a InProcServer32 wildcard-->
```

```
		<!--DATA: EventType, UtcTime, ProcessGuid, ProcessId,
Image, TargetObject, Details (can't filter on), NewName (can't filter
on)-->
	<RuleGroup name="" groupRelation="or">
		<RegistryEvent onmatch="include">
			<!--Autorun or Startups-->
				<!--ADDITIONAL REFERENCE: [
http://www.ghacks.net/2016/06/04/windows-automatic-startup-locations/ ]
-->
				<!--ADDITIONAL REFERENCE: [
https://view.officeapps.live.com/op/view.aspx?src=https://arsenalrecon.
com/downloads/resources/Registry_Keys_Related_to_Autorun.ods ] -->
				<!--ADDITIONAL REFERENCE: [
http://www.silentrunners.org/launchpoints.html ] -->
				<!--ADDITIONAL REFERENCE: [
https://www.microsoftpressstore.com/articles/article.aspx?p=2762082&seq
Num=2 ] -->
				<!--ADDITIONAL REFERENCE: [
https://web.archive.org/web/20200116001643/http://scholarworks.rit.edu/
cgi/viewcontent.cgi?article=1533&context=theses | Understanding malware
autostart techniques - Matthew Gottlieb ] -->
			<TargetObject name="T1060,RunKey"
condition="contains">CurrentVersion\Run</TargetObject> <!--Windows:
Wildcard for Run keys, including RunOnce, RunOnceEx, RunServices,
RunServicesOnce [Also covers terminal server] -->
			<TargetObject name="T1060,RunPolicy"
condition="contains">Policies\Explorer\Run</TargetObject> <!--Windows:
Alternate runs keys | Credit @ion-storm-->
```

```
                    <TargetObject name="T1484"
condition="contains">Group Policy\Scripts</TargetObject> <!--Windows:
Group policy scripts-->
                    <TargetObject name="T1484"
condition="contains">Windows\System\Scripts</TargetObject> <!--Windows:
Wildcard for Logon, Loggoff, Shutdown-->
                    <TargetObject name="T1060"
condition="contains">CurrentVersion\Windows\Load</TargetObject> <!--
Windows: [ https://msdn.microsoft.com/en-us/library/jj874148.aspx ] -->
                    <TargetObject name="T1060"
condition="contains">CurrentVersion\Windows\Run</TargetObject> <!--
Windows: [ https://msdn.microsoft.com/en-us/library/jj874148.aspx ] -->
                    <TargetObject name="T1060"
condition="contains">CurrentVersion\Winlogon\Shell</TargetObject> <!--
Windows: [ https://msdn.microsoft.com/en-
us/library/ms838576(v=winembedded.5).aspx ] -->
                    <TargetObject name="T1060"
condition="contains">CurrentVersion\Winlogon\System</TargetObject> <!--
Windows [ https://www.exterminate-it.com/malpedia/regvals/zlob-dns-
changer/118 ] -->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon\Notify</TargetObject> <!--Windows: Autorun
location [ https://attack.mitre.org/wiki/Technique/T1004 ] [
https://www.cylance.com/windows-registry-persistence-part-2-the-run-
keys-and-search-order ] -->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows
```

```
NT\CurrentVersion\Winlogon\Shell</TargetObject> <!--Windows: [
https://technet.microsoft.com/en-us/library/ee851671.aspx ] -->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon\Userinit</TargetObject> <!--Windows: Autorun
location [ https://www.cylance.com/windows-registry-persistence-part-2-
the-run-keys-and-search-order ] -->
                    <TargetObject condition="begin
with">HKLM\Software\WOW6432Node\Microsoft\Windows
NT\CurrentVersion\Drivers32</TargetObject> <!--Windows: Legacy driver
loading | Credit @ion-storm -->
                    <TargetObject condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Control\Session
Manager\BootExecute</TargetObject> <!--Windows: Autorun | Credit @ion-
storm | [ https://www.cylance.com/windows-registry-persistence-part-2-
the-run-keys-and-search-order ] -->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows
NT\CurrentVersion\AeDebug</TargetObject> <!--Windows: Automatic program
crash debug program [
https://www.symantec.com/security_response/writeup.jsp?docid=2007-
050712-5453-99&tabid=2 ] -->
                    <TargetObject
condition="contains">UserInitMprLogonScript</TargetObject> <!--Windows:
Legacy logon script environment variable [
http://www.hexacorn.com/blog/2014/11/14/beyond-good-ol-run-key-part-18/
] -->
                    <TargetObject name="T1112,ChangeStartupFolderPath"
condition="end with">user shell folders\startup</TargetObject> <!--
```

```
Monitor changes to Startup folder location for monitoring evasion |
Credit @SBousseaden-->
                    <!--Services-->
                    <TargetObject name="T1031,T1050" condition="end
with">\ServiceDll</TargetObject> <!--Windows: Points to a service's DLL
[ https://blog.cylance.com/windows-registry-persistence-part-1-
introduction-attack-phases-and-windows-services ] -->
                    <TargetObject name="T1031,T1050" condition="end
with">\ServiceManifest</TargetObject> <!--Windows: Manifest pointing to
service's DLL [
https://www.geoffchappell.com/studies/windows/win32/services/svchost/in
dex.htm ] -->
                    <TargetObject name="T1031,T1050" condition="end
with">\ImagePath</TargetObject> <!--Windows: Points to a service's EXE
[ https://attack.mitre.org/wiki/Technique/T1050 ] -->
                    <TargetObject name="T1031,T1050" condition="end
with">\Start</TargetObject> <!--Windows: Services start mode changes
(Disabled, Automatically, Manual)-->
                    <!--RDP-->
                    <TargetObject name="RDP port change"
condition="end with">Control\Terminal Server\WinStations\RDP-
Tcp\PortNumber</TargetObject> <!--Windows: RDP port change under
Control [ https://blog.menasec.net/2019/02/of-rdp-hijacking-part1-
remote-desktop.html ]-->
                    <TargetObject name="RDP port change"
condition="end with">Control\Terminal
Server\fSingleSessionPerUser</TargetObject> <!--Windows: Allow same
user to have mutliple RDP sessions, to hide from admin being
impersonated-->
```

```
                    <TargetObject name="ModifyRemoteDesktopState"
condition="end with">fDenyTSConnections</TargetObject> <!--Windows:
Attacker turning on RDP-->
                    <TargetObject condition="end
with">LastLoggedOnUser</TargetObject> <!--Windows: Changing last-logged
in user-->
                    <TargetObject name="ModifyRemoteDesktopPort"
condition="end with">RDP-tcp\PortNumber</TargetObject> <!--Windows:
Changing RDP port to evade IDS-->
                    <TargetObject condition="end
with">Services\PortProxy\v4tov4</TargetObject> <!--Windows: Changing
RDP port to evade IDS-->
                    <!--CLSID launch commands and Default File
Association changes-->
                    <TargetObject name="T1042"
condition="contains">\command\</TargetObject> <!--Windows: Sensitive
sub-key under file associations and CLSID that map to launch command-->
                    <TargetObject name="T1122"
condition="contains">\ddeexec\</TargetObject> <!--Windows: Sensitive
sub-key under file associations and CLSID that map to launch command-->
                    <TargetObject name="T1122"
condition="contains">{86C86720-42A0-1069-A2E8-
08002B30309D}</TargetObject> <!--Windows: Tooltip handler-->
                    <TargetObject name="T1042"
condition="contains">exefile</TargetObject> <!--Windows Executable
handler, to log any changes not already monitored-->
                    <!--Windows COM-->
                    <TargetObject name="T1122" condition="end
with">\InprocServer32\(Default)</TargetObject> <!--Windows:COM Object
```

```
Hijacking [ https://blog.gdatasoftware.com/2014/10/23941-com-object-
hijacking-the-discreet-way-of-persistence ] | Credit @ion-storm -->
                    <!--Windows shell visual modifications used by
malware-->
                    <TargetObject name="T1158" condition="end
with">\Hidden</TargetObject> <!--Windows:Explorer: Some types of
malware try to hide their hidden system files from the user, good
signal event -->
                    <TargetObject name="T1158" condition="end
with">\ShowSuperHidden</TargetObject> <!--Windows:Explorer: Some types
of malware try to hide their hidden system files from the user, good
signal event [ Example:
https://www.symantec.com/security_response/writeup.jsp?docid=2007-
061811-4341-99&tabid=2 ] -->
                    <TargetObject name="T1158" condition="end
with">\HideFileExt</TargetObject> <!--Windows:Explorer: Some malware
hides file extensions to make diagnosis/disinfection more daunting to
novice users -->
                    <!--Windows shell hijack and modifications-->
                    <TargetObject
condition="contains">Classes\*\</TargetObject> <!--Windows:Explorer: [
http://www.silentrunners.org/launchpoints.html ] -->
                    <TargetObject
condition="contains">Classes\AllFilesystemObjects\</TargetObject> <!--
Windows:Explorer: [ http://www.silentrunners.org/launchpoints.html ] --
>
                    <TargetObject
condition="contains">Classes\Directory\</TargetObject> <!--
Windows:Explorer: [
```

```
https://stackoverflow.com/questions/1323663/windows-shell-context-menu-
option ] -->
                    <TargetObject
condition="contains">Classes\Drive\</TargetObject> <!--
Windows:Explorer: [
https://stackoverflow.com/questions/1323663/windows-shell-context-menu-
option ] -->
                    <TargetObject
condition="contains">Classes\Folder\</TargetObject> <!--
Windows:Explorer: ContextMenuHandlers, DragDropHandlers,
CopyHookHandlers, [
https://stackoverflow.com/questions/1323663/windows-shell-context-menu-
option ] -->
                    <TargetObject
condition="contains">Classes\PROTOCOLS\</TargetObject> <!--
Windows:Explorer: Protocol handlers-->
                    <TargetObject
condition="contains">ContextMenuHandlers\</TargetObject> <!--Windows: [
http://oalabs.openanalysis.net/2015/06/04/malware-persistence-
hkey_current_user-shell-extension-handlers/ ] -->
                    <TargetObject
condition="contains">CurrentVersion\Shell</TargetObject> <!--Windows:
Shell Folders, ShellExecuteHooks, ShellIconOverloadIdentifers,
ShellServiceObjects, ShellServiceObjectDelayLoad [
http://oalabs.openanalysis.net/2015/06/04/malware-persistence-
hkey_current_user-shell-extension-handlers/ ] -->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows\CurrentVersion\explorer\ShellExec
uteHooks</TargetObject> <!--Windows: ShellExecuteHooks-->
```

```
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows\CurrentVersion\explorer\ShellServ
iceObjectDelayLoad</TargetObject> <!--Windows: ShellExecuteHooks-->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows\CurrentVersion\explorer\ShellIcon
OverlayIdentifiers</TargetObject> <!--Windows: ShellExecuteHooks-->
                    <!--AppPaths hijacking-->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows\CurrentVersion\App
Paths\</TargetObject> <!--Windows: Credit to @Hexacorn [
http://www.hexacorn.com/blog/2013/01/19/beyond-good-ol-run-key-part-3/
] -->
                    <!--Terminal service boobytrap-->
                    <TargetObject condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp\InitialProgram</TargetObject> <!--
Windows:RDP: Note other Terminal Server run keys are handled by another
wildcard already-->
                    <!--Group Policy integrity-->
                    <TargetObject name="T1484" condition="begin
with">HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon\GPExtensions\</TargetObject> <!--Windows:
Group Policy internally uses a plug-in architecture that nothing should
be modifying-->
                    <!--Winsock and Winsock2-->
                    <TargetObject condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Services\WinSock</TargetObject> <!-
-Windows: Wildcard, includes Winsock and Winsock2-->
```

```
                    <TargetObject condition="end
with">\ProxyServer</TargetObject> <!--Windows: System and user proxy
server-->
                    <!--Credential providers-->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows\CurrentVersion\Authentication\Cre
dential Provider</TargetObject> <!--Wildcard, includes Credential
Providers and Credential Provider Filters-->
                    <TargetObject name="T1101" condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Control\Lsa\</TargetObject> <!-- [
https://attack.mitre.org/wiki/Technique/T1131 ] [
https://attack.mitre.org/wiki/Technique/T1101 ] -->
                    <TargetObject condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders</TargetOb
ject> <!--Windows: Changes to WDigest-UseLogonCredential for password
scraping [ https://www.trustedsec.com/april-2015/dumping-wdigest-creds-
with-meterpreter-mimikatzkiwi-in-windows-8-1/ ] -->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Netsh</TargetObject> <!--Windows: Netsh
helper DLL [ https://attack.mitre.org/wiki/Technique/T1128 ] -->
                    <TargetObject
condition="contains">Software\Microsoft\Windows\CurrentVersion\Internet
Settings\ProxyEnable</TargetObject> <!--Windows: Malware often disables
a web proxy for 2nd stage downloads -->
                    <!--Networking-->
                    <TargetObject condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Control\NetworkProvider\Order\</Tar
getObject> <!--Windows: Order of network providers that are checked to
```

```
connect to destination [ https://www.malwarearchaeology.com/cheat-
sheets ] -->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows
NT\CurrentVersion\NetworkList\Profiles</TargetObject> <!--Windows: |
Credit @ion-storm -->
                    <TargetObject name="T1089" condition="end
with">\EnableFirewall</TargetObject> <!--Windows: Monitor for firewall
disablement, all firewall profiles [
https://attack.mitre.org/wiki/Technique/T1089 ] -->
                    <TargetObject name="T1089" condition="end
with">\DoNotAllowExceptions</TargetObject> <!--Windows: Monitor for
firewall disablement, all firewall profiles [
https://attack.mitre.org/wiki/Technique/T1089 ] -->
                    <TargetObject condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\Fi
rewallPolicy\StandardProfile\AuthorizedApplications\List</TargetObject>
<!--Windows Firewall authorized applications for all networks| Credit
@ion-storm -->
                    <TargetObject condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters\Fi
rewallPolicy\DomainProfile\AuthorizedApplications\List</TargetObject>
<!--Windows Firewall authorized applications for domain networks -->
                    <!--DLLs that get injected into every process at
launch-->
                    <TargetObject name="T1103" condition="begin
with">HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Windows\Appinit_Dlls\</TargetObject> <!--Windows:
```

```
Feature disabled by default [
https://attack.mitre.org/wiki/Technique/T1103 ] -->
                    <TargetObject name="T1103" condition="begin
with">HKLM\Software\Wow6432Node\Microsoft\Windows
NT\CurrentVersion\Windows\Appinit_Dlls\</TargetObject> <!--Windows:
Feature disabled by default [
https://attack.mitre.org/wiki/Technique/T1103 ] -->
                    <TargetObject condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Control\Session
Manager\AppCertDlls\</TargetObject> <!--Windows: Credit to @Hexacorn [
http://www.hexacorn.com/blog/2013/01/19/beyond-good-ol-run-key-part-3/
] [ https://blog.comodo.com/malware/trojware-win32-trojanspy-volisk-a/
] -->
                    <!--Office-->
                    <TargetObject name="T1137"
condition="contains">Microsoft\Office\Outlook\Addins\</TargetObject>
<!--Microsoft:Office: Outlook add-ins, access to sensitive data and
often cause issues-->
                    <TargetObject name="T1137"
condition="contains">Office Test\</TargetObject> <!-- Microsoft:Office:
Persistence method [ http://www.hexacorn.com/blog/2014/04/16/beyond-
good-ol-run-key-part-10/ ] | Credit @Hexacorn -->
                    <TargetObject
name="Context,ProtectedModeExitOrMacrosUsed"
condition="contains">Security\Trusted
Documents\TrustRecords</TargetObject> <!--Microsoft:Office: Monitor
when "Enable editing" or "Enable macros" is used | Credit @OutflankNL |
[ https://outflank.nl/blog/2018/01/16/hunting-for-evil-detect-macros-
being-executed/ ] -->
```

```
                    <TargetObject name="Context,ContactedDomain"
condition="end with">\EnableBHO</TargetObject> <!--Microsoft:Office:
Contacted domains stored here
'HKEY_CURRENT_USER\<SID>\SOFTWARE\Microsoft\Office\16.0\Common\Internet
\Server Cache\<domain>\EnableBHO' -->
                    <!--IE-->
                    <TargetObject name="T1176"
condition="contains">Internet Explorer\Toolbar\</TargetObject> <!--
Microsoft:InternetExplorer: Machine and user [ Example:
https://www.exterminate-it.com/malpedia/remove-mywebsearch ] -->
                    <TargetObject name="T1176"
condition="contains">Internet Explorer\Extensions\</TargetObject> <!--
Microsoft:InternetExplorer: Machine and user [ Example:
https://www.exterminate-it.com/malpedia/remove-mywebsearch ] -->
                    <TargetObject name="T1176"
condition="contains">Browser Helper Objects\</TargetObject> <!--
Microsoft:InternetExplorer: Machine and user [
https://msdn.microsoft.com/en-us/library/bb250436(v=vs.85).aspx ] -->
                    <TargetObject condition="end
with">\DisableSecuritySettingsCheck</TargetObject>
                    <TargetObject condition="end
with">\3\1206</TargetObject> <!--Microsoft:InternetExplorer: Malware
sometimes assures scripting is on in Internet Zone [
https://support.microsoft.com/en-us/help/182569/internet-explorer-
security-zones-registry-entries-for-advanced-users ] -->
                    <TargetObject condition="end
with">\3\2500</TargetObject> <!--Microsoft:InternetExplorer: Malware
sometimes disables Protected Mode in Internet Zone [
```

https://blog.avast.com/2013/08/12/your-documents-are-corrupted-from-image-to-an-information-stealing-trojan/] -->
<TargetObject condition="end with">\3\1809</TargetObject> <!--Microsoft:InternetExplorer: Malware sometimes disables Pop-up Blocker in Internet Zone [https://support.microsoft.com/en-us/help/182569/internet-explorer-security-zones-registry-entries-for-advanced-users] -->
<!--Magic registry keys-->
<TargetObject condition="begin with">HKLM\Software\Classes\CLSID\{AB8902B4-09CA-4BB6-B78D-A8F59079A8D5}\</TargetObject> <!--Windows: Thumbnail cache autostart [http://blog.trendmicro.com/trendlabs-security-intelligence/poweliks-levels-up-with-new-autostart-mechanism/] -->
<TargetObject condition="begin with">HKLM\Software\Classes\WOW6432Node\CLSID\{AB8902B4-09CA-4BB6-B78D-A8F59079A8D5}\</TargetObject> <!--Windows: Thumbnail cache autostart [http://blog.trendmicro.com/trendlabs-security-intelligence/poweliks-levels-up-with-new-autostart-mechanism/] -->
<TargetObject condition="begin with">HKLM\Software\Classes\CLSID\{083863F1-70DE-11d0-BD40-00A0C911CE86}\</TargetObject> <!--Windows: DirectX instances-->
<TargetObject condition="begin with">HKLM\Software\Classes\WOW6432Node\CLSID\{083863F1-70DE-11d0-BD40-00A0C911CE86}\</TargetObject> <!--Windows: DirectX instances-->
<!--Install/Run artifacts-->
<TargetObject condition="end with">\UrlUpdateInfo</TargetObject> <!--Microsoft:ClickOnce: Source URL is stored in this value [

```
https://subt0x10.blogspot.com/2016/12/mimikatz-delivery-via-clickonce-
with.html ] -->
                    <TargetObject condition="end
with">\InstallSource</TargetObject> <!--Windows: Source folder for
certain program and component installations-->
                    <TargetObject name="Alert,Sysinternals Tool Used"
condition="end with">\EulaAccepted</TargetObject> <!--Sysinternals tool
launched. Lots of useful abilities for attackers -->
                    <!--Antivirus tampering-->
                    <TargetObject name="T1089,Tamper-Defender"
condition="end with">\DisableAntiSpyware</TargetObject> <!--
Windows:Defender: State modified via registry-->
                    <TargetObject name="T1089,Tamper-Defender"
condition="end with">\DisableAntiVirus</TargetObject> <!--
Windows:Defender: State modified via registry-->
                    <TargetObject name="T1089,Tamper-Defender"
condition="end with">\SpynetReporting</TargetObject> <!--
Windows:Defender: State modified via registry-->
                    <TargetObject name="T1089,Tamper-Defender"
condition="end with">DisableRealtimeMonitoring</TargetObject> <!--
Windows:Defender: State modified via registry-->
                    <TargetObject name="T1089,Tamper-Defender"
condition="end with">\SubmitSamplesConsent</TargetObject> <!--
Windows:Defender: State modified via registry-->
                    <TargetObject name="T1562,Tamper-Defender"
condition="begin with">HKLM\SOFTWARE\Policies\Microsoft\Windows
Defender\Exclusions\</TargetObject> <!--Windows:Defender: Exclusions in
policy key-->
                    <!--Windows UAC tampering-->
```

```
                    <TargetObject name="T1088" condition="end
with">HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\En
ableLUA</TargetObject> <!--Detect: UAC Tampering | Credit @ion-storm --
>
                    <TargetObject name="T1088" condition="end
with">HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\Lo
calAccountTokenFilterPolicy</TargetObject> <!--Detect: UAC Tampering |
Credit @ion-storm -->
                    <!--Microsoft Security Center tampering | Credit
@ion-storm -->
                    <TargetObject name="T1089,Tamper-SecCenter"
condition="end with">HKLM\Software\Microsoft\Security
Center\</TargetObject> <!-- [
https://attack.mitre.org/wiki/Technique/T1089 ] -->
                    <TargetObject name="T1089,Tamper-SecCenter"
condition="end
with">SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\HideS
CAHealth</TargetObject> <!--Windows:Security Center: Malware sometimes
disables [ https://blog.avast.com/2013/08/12/your-documents-are-
corrupted-from-image-to-an-information-stealing-trojan/ ] -->
                    <!--Windows application compatibility-->
                    <TargetObject name="T1138,AppCompatShim"
condition="begin with">HKLM\Software\Microsoft\Windows
NT\CurrentVersion\AppCompatFlags\Custom</TargetObject> <!--Windows:
AppCompat [ https://www.fireeye.com/blog/threat-research/2017/05/fin7-
shim-databases-persistence.html ] -->
                    <TargetObject name="T1138,AppCompatShim"
condition="begin with">HKLM\Software\Microsoft\Windows
NT\CurrentVersion\AppCompatFlags\InstalledSDB</TargetObject> <!--
```

```
Windows: AppCompat [ https://attack.mitre.org/wiki/Technique/T1138 ] --
>
                    <TargetObject
condition="contains">VirtualStore</TargetObject> <!--Windows: Registry
virtualization, something's wrong if it's in use [
https://msdn.microsoft.com/en-
us/library/windows/desktop/aa965884(v=vs.85).aspx ] -->
                    <!--Windows internals integrity monitoring-->
                    <TargetObject name="T1183,IFEO" condition="begin
with">HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File
Execution Options\</TargetObject> <!--Windows: Malware likes changing
IFEO, like adding Debugger to disable antivirus EXE-->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows\CurrentVersion\WINEVT\</TargetObj
ect> <!--Windows: Event log system integrity and ACLs-->
                    <TargetObject name="Tamper-Safemode"
condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Control\Safeboot\</TargetObject>
<!--Windows: Services approved to load in safe mode. Almost nothing
should ever modify this.-->
                    <TargetObject name="Tamper-Winlogon"
condition="begin
with">HKLM\SYSTEM\CurrentControlSet\Control\Winlogon\</TargetObject>
<!--Windows: Providers notified by WinLogon-->
                    <TargetObject
name="Context,DeviceConnectedOrUpdated" condition="end
with">\FriendlyName</TargetObject> <!--Windows: New devices connected
and remembered-->
```

```
                    <TargetObject name="Context,MsiInstallerStarted"
condition="is">HKLM\Software\Microsoft\Windows\CurrentVersion\Installer
\InProgress\(Default)</TargetObject> <!--Windows: See when
WindowsInstaller is engaged, useful for timeline matching with other
events-->
                    <TargetObject name="Tamper-Tracing"
condition="begin
with">HKLM\Software\Microsoft\Tracing\RASAPI32</TargetObject> <!--
Windows: Malware sometimes disables tracing to obfuscate tracks-->
                    <TargetObject
name="Context,ProcessAccessedPrivateResource" condition="begin
with">HKLM\Software\Microsoft\Windows\CurrentVersion\CapabilityAccessMa
nager\ConsentStore\</TargetObject> <!-- Windows: Win10 tracks when and
what process uses webcam/microphone/location etc [
https://medium.com/@7a616368/can-you-track-processes-accessing-the-
camera-and-microphone-7e6885b37072 ] -->
                    <TargetObject condition="contains">\Keyboard
Layout\Preload</TargetObject> <!--Microsoft:Windows: Keyboard layout
loaded into user session [
https://renenyffenegger.ch/notes/Windows/registry/tree/HKEY_CURRENT_USE
R/Keyboard-Layout/Preload/index ] | Credit @cyb3rops -->
                    <TargetObject condition="contains">\Keyboard
Layout\Substitutes</TargetObject> <!--Microsoft:Windows: Keyboard
layout loaded into user session [
https://renenyffenegger.ch/notes/Windows/registry/tree/HKEY_CURRENT_USE
R/Keyboard-Layout/Preload/index ] | Credit @cyb3rops -->
                    <!--Windows inventory events-->
                    <TargetObject name="InvDB-Path" condition="end
with">\LowerCaseLongPath</TargetObject> <!-- [
```

```
https://binaryforay.blogspot.com/2017/10/amcache-still-rules-
everything-around.html ] -->
                    <TargetObject name="InvDB-Pub" condition="end
with">\Publisher</TargetObject> <!-- [
https://binaryforay.blogspot.com/2017/10/amcache-still-rules-
everything-around.html ] -->
                    <TargetObject name="InvDB-Ver" condition="end
with">\BinProductVersion</TargetObject> <!-- [
https://docs.microsoft.com/en-us/windows/privacy/basic-level-windows-
diagnostic-events-and-fields-1709 ] -->
                    <TargetObject name="InvDB-DriverVer"
condition="end with">\DriverVersion</TargetObject> <!-- [ https://df-
stream.com/2015/02/leveraging-devicecontainers-key/ ] -->
                    <TargetObject name="InvDB-DriverVer"
condition="end with">\DriverVerVersion</TargetObject> <!-- [
https://df-stream.com/2015/02/leveraging-devicecontainers-key/ ] -->
                    <TargetObject name="InvDB-CompileTimeClaim"
condition="end with">\LinkDate</TargetObject> <!-- Compile time of EXE,
may not be reliable [ https://en.wikipedia.org/wiki/Link_time ] -->
                    <TargetObject name="InvDB"
condition="contains">Compatibility Assistant\Store\</TargetObject> <!--
Inventory -->
                    <!--Suspicious sources-->
                    <Image name="Suspicious,ImageBeginWithBackslash"
condition="end with">regedit.exe</Image> <!--Users and helpdesk staff
making system modifications -->
                    <Image name="Suspicious,ImageBeginWithBackslash"
condition="begin with">\</Image> <!--Devices and VSC shouldn't be
executing changes | Credit: @SBousseaden @ionstorm @neu5ron
```

```
@PerchedSystems [
https://twitter.com/SwiftOnSecurity/status/1133167323991486464 ] -->
                </RegistryEvent>
        </RuleGroup>

        <RuleGroup name="" groupRelation="or">
                <RegistryEvent onmatch="exclude">
                <!--COMMENT:   Remove low-information noise. Often these
hide a procress recreating an empty key and do not hide the values
created subsequently.-->
                <!--NOTE:      A lot of noise can be removed by excluding
CreateKey events, which are largely innocuous-->
                        <TargetObject condition="contains">\{CAFEEFAC-
</TargetObject>
                        <EventType condition="is">CreateKey</EventType>
                        <TargetObject condition="begin
with">HKLM\COMPONENTS</TargetObject>
                        <!--Inventory noise-->
                        <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows\CurrentVersion\AppModel\StateRepo
sitory\Cache</TargetObject>
                        <!--Misc-->
                        <TargetObject condition="end
with">Toolbar\WebBrowser</TargetObject> <!--Microsoft:IE: Extraneous
activity-->
                        <TargetObject condition="end
with">Browser\ITBar7Height</TargetObject> <!--Microsoft:IE: Extraneous
activity, covers ShellBrowser and WebBrowser-->
```

```
                    <TargetObject condition="end
with">Browser\ITBar7Layout</TargetObject> <!--Microsoft:IE: Extraneous
activity-->
                    <TargetObject condition="end with">Internet
Explorer\Toolbar\Locked</TargetObject> <!--Windows:Explorer: Extraneous
activity-->
                    <TargetObject condition="end
with">Toolbar\WebBrowser\{47833539-D0C5-4125-9FA8-
0819E2EAAC93}</TargetObject> <!--Windows:Explorer: Extraneous activity-
->
                    <TargetObject condition="end
with">}\PreviousPolicyAreas</TargetObject> <!--Windows: Remove noise
from \Winlogon\GPExtensions by svchost.exe-->
                    <TargetObject
condition="contains">\Control\WMI\Autologger\</TargetObject> <!--
Windows: Remove noise from monitoring "\Start"-->
                    <TargetObject condition="end
with">HKLM\SYSTEM\CurrentControlSet\Services\UsoSvc\Start</TargetObject
> <!--Windows: Remove noise from monitoring "\Start"-->
                    <TargetObject condition="end
with">\Lsa\OfflineJoin\CurrentValue</TargetObject> <!--Windows:
Sensitive value during domain join-->
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows\CurrentVersion\Installer\UserData
\S-1-5-18\</TargetObject> <!--Windows: Remove noise monitoring
installations run as system-->
                    <TargetObject
condition="contains">_Classes\AppX</TargetObject> <!--Windows: Remove
noise monitoring "Shell\open\command"--> <!--Win8+-->
```

```
                    <TargetObject condition="begin
with">HKLM\Software\Microsoft\Windows\CurrentVersion\WINEVT\Publishers\
</TargetObject> <!--Windows: SvcHost Noise-->
                    <!--Bootup Control noise-->
                    <TargetObject condition="end
with">HKLM\SYSTEM\CurrentControlSet\Control\Lsa\LsaPid</TargetObject>
<!--Windows:lsass.exe: Boot noise-->
                    <TargetObject condition="end
with">HKLM\SYSTEM\CurrentControlSet\Control\Lsa\SspiCache</TargetObject
> <!--Windows:lsass.exe: Boot noise--> <!--Win8+-->
                    <TargetObject condition="end
with">HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Domains</Targe
tObject> <!--Windows:lsass.exe: Boot noise--> <!--Win8+-->
                    <!--Services startup settings noise, some low-risk
services routinely change it and this can be ignored-->
                    <TargetObject condition="end
with">\Services\BITS\Start</TargetObject> <!--Windows: Remove noise
from monitoring "\Start"-->
                    <TargetObject condition="end
with">\services\clr_optimization_v2.0.50727_32\Start</TargetObject> <!-
-Microsoft:dotNet: Windows 7-->
                    <TargetObject condition="end
with">\services\clr_optimization_v2.0.50727_64\Start</TargetObject> <!-
-Microsoft:dotNet: Windows 7-->
                    <TargetObject condition="end
with">\services\clr_optimization_v4.0.30319_32\Start</TargetObject> <!-
-Microsoft:dotNet: Windows 10-->
```

```
                    <TargetObject condition="end
with">\services\clr_optimization_v4.0.30319_64\Start</TargetObject> <!-
-Microsoft:dotNet: Windows 10-->
                    <TargetObject condition="end
with">\services\deviceAssociationService\Start</TargetObject> <!--
Windows: Remove noise from monitoring "\Start"-->
                    <TargetObject condition="end
with">\services\fhsvc\Start</TargetObject> <!--Windows: File History
Service-->
                    <TargetObject condition="end
with">\services\nal\Start</TargetObject> <!--Intel: Network adapter
diagnostic driver-->
                    <TargetObject condition="end
with">\services\trustedInstaller\Start</TargetObject> <!--Windows:
Remove noise from monitoring "\Start"-->
                    <TargetObject condition="end
with">\services\tunnel\Start</TargetObject> <!--Windows: Remove noise
from monitoring "\Start"-->
                    <TargetObject condition="end
with">\services\usoSvc\Start</TargetObject> <!--Windows: Remove noise
from monitoring "\Start"-->
                    <!--FileExts noise filtering-->
                    <TargetObject condition="end
with">\UserChoice\ProgId</TargetObject> <!--Windows: Remove noise from
monitoring "FileExts"--> <!--Win8+-->
                    <TargetObject condition="end
with">\UserChoice\Hash</TargetObject> <!--Windows: Remove noise from
monitoring "FileExts"--> <!--Win8+-->
```

```
                    <TargetObject condition="end
with">\OpenWithList\MRUList</TargetObject> <!--Windows: Remove noise
from monitoring "FileExts"-->
                    <TargetObject condition="contains">Shell
Extentions\Cached</TargetObject> <!--Windows: Remove noise generated by
explorer.exe on monitored ShellCached binary keys--> <!--Win8+-->
                    <!--Group Policy noise-->
                    <TargetObject condition="end
with">HKLM\System\CurrentControlSet\Control\Lsa\Audit\SpecialGroups</Ta
rgetObject> <!--Windows: Routinely set through Group Policy, not
especially important to log-->
                    <TargetObject condition="end
with">SOFTWARE\Microsoft\Windows\CurrentVersion\Group
Policy\Scripts\Startup\0\PSScriptOrder</TargetObject> <!--Windows:Group
Policy: Noise below the actual key while building-->
                    <TargetObject condition="end
with">SOFTWARE\Microsoft\Windows\CurrentVersion\Group
Policy\Scripts\Startup\0\SOM-ID</TargetObject> <!--Windows:Group
Policy: Noise below the actual key while building-->
                    <TargetObject condition="end
with">SOFTWARE\Microsoft\Windows\CurrentVersion\Group
Policy\Scripts\Startup\0\GPO-ID</TargetObject> <!--Windows:Group
Policy: Noise below the actual key while building-->
                    <TargetObject condition="end
with">SOFTWARE\Microsoft\Windows\CurrentVersion\Group
Policy\Scripts\Startup\0\0\IsPowershell</TargetObject> <!--
Windows:Group Policy: Noise below the actual key while building-->
                    <TargetObject condition="end
with">SOFTWARE\Microsoft\Windows\CurrentVersion\Group
```

Policy\Scripts\Startup\0\0\ExecTime</TargetObject> <!--Windows:Group Policy: Noise below the actual key while building-->
<TargetObject condition="end with">SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Shutdown\0\PSScriptOrder</TargetObject> <!--Windows:Group Policy: Noise below the actual key while building-->
<TargetObject condition="end with">SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Shutdown\0\SOM-ID</TargetObject> <!--Windows:Group Policy: Noise below the actual key while building-->
<TargetObject condition="end with">SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Shutdown\0\GPO-ID</TargetObject> <!--Windows:Group Policy: Noise below the actual key while building-->
<TargetObject condition="end with">SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Shutdown\0\0\IsPowershell</TargetObject> <!--Windows:Group Policy: Noise below the actual key while building-->
<TargetObject condition="end with">SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\Scripts\Shutdown\0\0\ExecTime</TargetObject> <!--Windows:Group Policy: Noise below the actual key while building-->
<TargetObject condition="contains">\safer\codeidentifiers\0\HASHES\{</TargetObject> <!--Windows: Software Restriction Policies. Can be used to disable security tools, but very noisy to monitor if you use it-->
<!--SECTION: Office C2R-->
<TargetObject condition="contains">VirtualStore\MACHINE\SOFTWARE\Microsoft\Office\Cli

```
ckToRun\</TargetObject> <!--Microsoft: SearchProtocolHost writes to
OfficeC2R registry for Outlook, seemingly regarding mail indexing-->
                        <TargetObject condition="begin
with">HKLM\SOFTWARE\Microsoft\Office\ClickToRun\</TargetObject> <!--
Microsoft: Virtual registry for Office-->
                        <!--SECTION: 3rd party-->
                        <Image condition="is">C:\Program
Files\WIDCOMM\Bluetooth Software\btwdins.exe</Image> <!--Constantly
writes to HKLM-->
                        <TargetObject condition="begin
with">HKCR\VLC.</TargetObject> <!--VLC update noise-->
                        <TargetObject condition="begin
with">HKCR\iTunes.</TargetObject> <!--Apple: iTunes update noise-->
                        <!--WINEVT publishers noise-->
                        <TargetObject
condition="is">HKLM\Software\Microsoft\Windows\CurrentVersion\WINEVT\Pu
blishers\{945a8954-c147-4acd-923f-40c45405a658}</TargetObject> <!--
Windows update-->
                </RegistryEvent>
        </RuleGroup>

        <!--SYSMON EVENT ID 15 : ALTERNATE DATA STREAM CREATED
[FileCreateStreamHash]-->
                <!--EVENT 15: "File stream created"-->
                <!--COMMENT:   Any files created with an NTFS Alternate
Data Stream which match these rules will be hashed and logged.
                        [
https://blogs.technet.microsoft.com/askcore/2013/03/24/alternate-data-
streams-in-ntfs/ ]
```

```
                    ADS's are used by browsers and email clients to
mark files as originating from the Internet or other foreign sources.
                    [ https://textslashplain.com/2016/04/04/downloads-
and-the-mark-of-the-web/ ] -->
                <!--NOTE: Other filesystem minifilters can make it appear
to Sysmon that some files are being written twice. This is not a Sysmon
issue, per Mark Russinovich.-->

                <!--DATA: UtcTime, ProcessGuid, ProcessId, Image,
TargetFilename, CreationUtcTime, Hash-->
                <FileCreateStreamHash onmatch="include">
                        <TargetFilename
condition="contains">Downloads</TargetFilename> <!--Downloaded files.
Does not include "Run" files in IE-->
                        <TargetFilename
condition="contains">Temp\7z</TargetFilename> <!--7zip extractions-->
                        <TargetFilename
condition="contains">Startup</TargetFilename> <!--ADS startup |
Example: [ https://www.hybrid-
analysis.com/sample/a314f6106633fba4b70f9d6ddbee452e8f8f44a72117749c212
43dc93c7ed3ac?environmentId=100 ] -->
                        <TargetFilename condition="end
with">.bat</TargetFilename> <!--Batch scripting-->
                        <TargetFilename condition="end
with">.cmd</TargetFilename> <!--Batch scripting | Credit @ion-storm -->
                        <TargetFilename condition="end
with">.doc</TargetFilename> <!--Office doc potentially with macro -->
                        <TargetFilename condition="end
with">.hta</TargetFilename> <!--Scripting-->
```

```
                    <TargetFilename condition="end
with">.jse</TargetFilename> <!--Registry File-->
                    <TargetFilename condition="end
with">.lnk</TargetFilename> <!--Shortcut file | Credit @ion-storm -->
                    <TargetFilename condition="end
with">.ppt</TargetFilename> <!--Office doc potentially with macros-->
                    <TargetFilename condition="end
with">.ps1</TargetFilename> <!--PowerShell-->
                    <TargetFilename condition="end
with">.ps2</TargetFilename> <!--PowerShell-->
                    <TargetFilename condition="end
with">.reg</TargetFilename> <!--Registry File-->
                    <TargetFilename condition="end
with">.sct</TargetFilename> <!--Scripting | Credit @bartblaze -->
                    <TargetFilename condition="end
with">.vb</TargetFilename> <!--VisualBasicScripting files-->
                    <TargetFilename condition="end
with">.vbe</TargetFilename> <!--VisualBasicScripting files-->
                    <TargetFilename condition="end
with">.vbs</TargetFilename> <!--VisualBasicScripting files-->
                    <TargetFilename condition="end
with">.wsc</TargetFilename> <!--Scripting | Credit @bartblaze -->
                    <TargetFilename condition="end
with">.wsf</TargetFilename> <!--Scripting | Credit @bartblaze -->
             </FileCreateStreamHash>

      <RuleGroup name="" groupRelation="or">
             <FileCreateStreamHash onmatch="exclude">
             </FileCreateStreamHash>
```

```
        </RuleGroup>

        <!--SYSMON EVENT ID 16 : SYSMON CONFIGURATION CHANGE-->
                <!--EVENT 16: "Sysmon config state changed"-->
                <!--COMMENT:  This ONLY logs if the hash of the
configuration changes. Running "sysmon.exe -c" with the current
configuration will not be logged with Event 16-->

                <!--DATA: UtcTime, Configuration, ConfigurationFileHash--
>
                <!--Cannot be filtered.-->

        <!--SYSMON EVENT ID 17 & 18 : PIPE CREATED / PIPE CONNECTED
[PipeEvent]-->
                <!--EVENT 17: "Pipe Created"-->
                <!--EVENT 18: "Pipe Connected"-->

                <!--ADDITIONAL REFERENCE: [
https://www.cobaltstrike.com/help-smb-beacon ] -->
                <!--ADDITIONAL REFERENCE: [
https://blog.cobaltstrike.com/2015/10/07/named-pipe-pivoting/ ] -->

                <!--DATA: UtcTime, ProcessGuid, ProcessId, PipeName,
Image-->
                <RuleGroup name="" groupRelation="or">
                        <PipeEvent onmatch="include">
                                <!-- Remote Command Execution Tools -->
                                <PipeName condition="contains
any">paexec;remcom;csexec</PipeName>
```

```
                                <!-- Password or Credential Dumpers -->
                                <PipeName condition="contains
any">\lsadump;\cachedump;\wceservicepipe</PipeName>
                                <!-- Malware -->
                                <PipeName condition="contains
any">\isapi_http;\isapi_dg;\isapi_dg2;\sdlrpc;\ahexec;\winsession;\lsas
sw;\46a676ab7f179e511e30dd2dc41bd388;\9f81f59bc58452127884ce513865ed20;
\e710f28d59aa529d6792ca6ff0ca1b34;\rpchlp_3;\NamePipe_MoreWindows;\pche
ap_reuse;\gruntsvc;\583da945-62af-10e8-4902-
a8f205c72b2e;\bizkaz;\svcctl;\Posh;\jaccdpqnvbrrxlaf;\csexecsvc</PipeNa
me>
                                <PipeName condition="contains
any">\atctl;\userpipe;\iehelper;\sdlrpc;\comnap</PipeName>
                                <!-- Cobalt Strike Pipe Names -->
                                <PipeName condition="contains all">MSSE-;-
server</PipeName>
                                <PipeName condition="begin
with">\postex_</PipeName>
                                <PipeName condition="begin
with">\postex_ssh_</PipeName>
                                <PipeName condition="begin
with">\status_</PipeName>
                                <PipeName condition="begin
with">\msagent_</PipeName>
                        </PipeEvent>
                </RuleGroup>

        <!--SYSMON EVENT ID 19 & 20 & 21 : WMI EVENT MONITORING
[WmiEvent]-->
```

```
		<!--EVENT 19: "WmiEventFilter activity detected"-->
		<!--EVENT 20: "WmiEventConsumer activity detected"-->
		<!--EVENT 21: "WmiEventConsumerToFilter activity
detected"-->

		<!--ADDITIONAL REFERENCE: [
https://www.darkoperator.com/blog/2017/10/15/sysinternals-sysmon-610-
tracking-of-permanent-wmi-events ] -->
		<!--ADDITIONAL REFERENCE: [
https://rawsec.lu/blog/posts/2017/Sep/19/sysmon-v610-vs-wmi-
persistence/ ] -->

		<!--DATA: EventType, UtcTime, Operation, User, Name,
Type, Destination, Consumer, Filter-->
	<RuleGroup name="" groupRelation="or">
		<WmiEvent onmatch="exclude">
			<!--NOTE: Using exclude with no rules means
everything will be logged-->
		</WmiEvent>
	</RuleGroup>

	<!--SYSMON EVENT ID 22 : DNS QUERY [DnsQuery]-->
		<!--EVENT 22: "Dns query"-->

		<!--NOTE:	Due to the volume of events that DNS
queries generate, some orgs may want to remove this section from their
configuration to reduce Sysmon log turnover. -->
```

<!--COMMENT: DNS logging is a very nuanced challenge in monitoring due to event volume. Legitimate domains can be used to host malware/C2, but lookup itself is not very informative.

It's fine to exclude monitoring these bulk low-value lookups, but at same time, you would not have a full log of how malware communicated, potentially missing C2.

This section of Sysmon configuration will require your full judgement and knowledge of your org's priorities. There is no correct answer.-->

<!--OPERATIONS: Chrome and Firefox prefetch DNS lookups, or use alternate DNS lookup methods Sysmon won't capture. You need to turn these off.

Search for Group Policy for these browsers to configure this.-->

<!--OPERATIONS: Most DNS traffic is web advertising. To significantly reduce DNS queries and malware ads, enable client-side advertising filtering via Group Policy. This is easy.

Internet Explorer: https://decentsecurity.com/adblocking-for-internet-explorer-deployment/

Chrome: https://decentsecurity.com/ublock-for-google-chrome-deployment/

Firefox: ToDo

Also note, this configuration is designed for United States computers. Your country's users will may need customization to reduce noise.

-->

<!--CONFIG: DNS poisoning is an issue during threat investigations. Try to only exclude ROUTINE system-level queries you know are strongly validated with HTTPS or code signing.-->

<!--CONFIG: If you exclude microsoft.com, someone could register malware-microsoft.com and it wouldn't be logged. Use "END WITH" with leading . or "IS" operators.-->

<!--CONFIG: Be very specific in exclusions. Threat actors use legitimate services, too. Dont exclude all of AWS or Azure or Google or CDNs!-->

<!--CONFIG: Popularity data: [http://s3-us-west-1.amazonaws.com/umbrella-static/index.html] [https://better.fyi/trackers/alexa-top-500-news/] -->

<!--CRITICAL: Do NOT exclude "wpad" lookups. This is a MitM vector routinely used by attackers. Disable WPAD or enforce client-side DNSSEC for AD domain lookups.-->

<!--CRITICAL: Do NOT exclude IPv6 lookups.-->

<!--DATA: RuleName, UtcTime, ProcessGuid, ProcessId, QueryName, QueryType, QueryStatus, QueryResults (can't filter on)-->

<!--BELOW: These domains should not be excluded at the top level. Be specific if you want to reduce noise under them.-->

<!-- Rejected: .cloudapp.net, customer content [https://blogs.technet.microsoft.com/ptsblog/2012/06/18/security-consideration-when-using-cloudapp-net-domain-as-production-environment-in-windows-azure/] -->

```
                <!-- Rejected: .googleapis.com, customer content [
https://www.zdnet.com/article/this-business-email-scam-spreads-trojans-
through-google-cloud-storage/ ] -->
                <!-- Rejected: .cloudfront.net, customer content -->
                <!-- Rejected: .windows.net, customer content -->
                <!-- Rejected: *github.com, customer content, including
open-source malware components -->

        <RuleGroup name="" groupRelation="or">
                <DnsQuery onmatch="exclude">
                        <!--Network noise-->
                        <QueryName condition="end with">.arpa.</QueryName>
<!--Design decision to not log reverse DNS lookups. You will need to
decide.-->
                        <QueryName condition="end with">.arpa</QueryName>
<!--Design decision to not log reverse DNS lookups. You will need to
decide.-->
                        <QueryName condition="end
with">.msftncsi.com</QueryName> <!--Microsoft proxy detection |
Microsoft default exclusion-->
                        <QueryName
condition="is">..localmachine</QueryName>
                        <QueryName condition="is">localhost</QueryName>
                        <!--Microsoft-->
                        <QueryName condition="end with">-
pushp.svc.ms</QueryName> <!--Microsoft: Doesn't appear to host customer
content or subdomains-->
```

```
                    <QueryName condition="end with">.b-
msedge.net</QueryName> <!--Microsoft: Doesn't appear to host customer
content or subdomains-->
                    <QueryName condition="end
with">.bing.com</QueryName> <!-- Microsoft | Microsoft default
exclusion -->
                    <QueryName condition="end
with">.hotmail.com</QueryName> <!--Microsoft | Microsoft default
exclusion-->
                    <QueryName condition="end
with">.live.com</QueryName> <!--Microsoft | Microsoft default
exclusion-->
                    <QueryName condition="end
with">.live.net</QueryName> <!--Microsoft | Microsoft default
exclusion-->
                    <QueryName condition="end with">.s-
microsoft.com</QueryName> <!--Microsoft-->
                    <QueryName condition="end
with">.microsoft.com</QueryName> <!--Microsoft | Microsoft default
exclusion-->
                    <QueryName condition="end
with">.microsoftonline.com</QueryName> <!--Microsoft | Microsoft
default exclusion-->
                    <QueryName condition="end
with">.microsoftstore.com</QueryName> <!--Microsoft | Microsoft default
exclusion-->
                    <QueryName condition="end with">.ms-
acdc.office.com</QueryName> <!--Microsoft: Doesn't appear to host
customer content or subdomains-->
```

```
                        <QueryName condition="end
with">.msedge.net</QueryName> <!--Microsoft: Doesn't appear to host
customer content or subdomains-->
                        <QueryName condition="end
with">.msn.com</QueryName> <!--Microsoft | Microsoft default exclusion-
->
                        <QueryName condition="end
with">.msocdn.com</QueryName> <!--Microsoft-->
                        <QueryName condition="end
with">.skype.com</QueryName> <!--Microsoft | Microsoft default
exclusion-->
                        <QueryName condition="end
with">.skype.net</QueryName> <!--Microsoft | Microsoft default
exclusion-->
                        <QueryName condition="end
with">.windows.com</QueryName> <!--Microsoft-->
                        <QueryName condition="end
with">.windows.net.nsatc.net</QueryName> <!--Microsoft-->
                        <QueryName condition="end
with">.windowsupdate.com</QueryName> <!--Microsoft-->
                        <QueryName condition="end
with">.xboxlive.com</QueryName> <!--Microsoft-->
                        <QueryName
condition="is">login.windows.net</QueryName> <!--Microsoft-->
                        <Image condition="begin
with">C:\ProgramData\Microsoft\Windows Defender\Platform\</Image> <!--
Microsoft: https://docs.microsoft.com/en-us/windows/security/threat-
protection/microsoft-defender-atp/network-protection -->
                        <!--Microsoft:Office365/AzureAD-->
```

```
                    <QueryName condition="end
with">.activedirectory.windowsazure.com</QueryName> <!--Microsoft:
AzureAD-->
                    <QueryName condition="end
with">.aria.microsoft.com</QueryName> <!--Microsoft:
OneDrive/SharePoint-->
                    <QueryName condition="end
with">.msauth.net</QueryName>
                    <QueryName condition="end
with">.msftauth.net</QueryName>
                    <QueryName condition="end
with">.office.net</QueryName> <!--Microsoft: Office-->
                    <QueryName condition="end
with">.opinsights.azure.com</QueryName> <!--Microsoft: AzureAD/InTune
client event monitoring-->
                    <QueryName condition="end
with">.res.office365.com</QueryName> <!--Microsoft: Office-->
                    <QueryName condition="is">acdc-
direct.office.com</QueryName> <!--Microsoft: Office-->
                    <QueryName condition="is">atm-fp-
direct.office.com</QueryName> <!--Microsoft: Office-->
                    <QueryName
condition="is">loki.delve.office.com</QueryName> <!--Microsoft: Office-
->
                    <QueryName
condition="is">management.azure.com</QueryName> <!--Microsoft:
AzureAD/InTune-->
```

```
                    <QueryName
condition="is">messaging.office.com</QueryName> <!--Microsoft: Office--
>
                    <QueryName
condition="is">outlook.office365.com</QueryName> <!--Microsoft:
Protected by HSTS-->
                    <QueryName
condition="is">portal.azure.com</QueryName> <!--Microsoft:
AzureAD/InTune-->
                    <QueryName
condition="is">protection.outlook.com</QueryName> <!--Microsoft:
Office-->
                    <QueryName
condition="is">substrate.office.com</QueryName> <!--Microsoft: Office--
>
                    <QueryName condition="end
with">.measure.office.com</QueryName> <!--Microsoft: Office-->
                    <!--3rd-party applications-->
                    <QueryName condition="end
with">.adobe.com</QueryName> <!--Adobe-->
                    <QueryName condition="end
with">.adobe.io</QueryName> <!--Adobe-->
                    <QueryName condition="end
with">.mozaws.net</QueryName> <!--Mozilla-->
                    <QueryName condition="end
with">.mozilla.com</QueryName> <!--Mozilla-->
                    <QueryName condition="end
with">.mozilla.net</QueryName> <!--Mozilla-->
```

```
                        <QueryName condition="end
with">.mozilla.org</QueryName> <!--Mozilla-->
                        <QueryName condition="end
with">.spotify.com</QueryName> <!--Spotify-->
                        <QueryName condition="end
with">.spotify.map.fastly.net</QueryName> <!--Spotify-->
                        <QueryName condition="end
with">.wbx2.com</QueryName> <!--Webex-->
                        <QueryName condition="end
with">.webex.com</QueryName> <!--Webex-->
                        <QueryName
condition="is">clients1.google.com</QueryName> <!--Google-->
                        <QueryName
condition="is">clients2.google.com</QueryName> <!--Google-->
                        <QueryName
condition="is">clients3.google.com</QueryName> <!--Google-->
                        <QueryName
condition="is">clients4.google.com</QueryName> <!--Google-->
                        <QueryName
condition="is">clients5.google.com</QueryName> <!--Google-->
                        <QueryName
condition="is">clients6.google.com</QueryName> <!--Google-->
                        <QueryName
condition="is">safebrowsing.googleapis.com</QueryName> <!--Google-->
                        <!--Goodlist CDN-->
                        <QueryName condition="end
with">.akadns.net</QueryName> <!--AkamaiCDN, extensively used by
Microsoft | Microsoft default exclusion-->
```

```
                        <QueryName condition="end
with">.netflix.com</QueryName>
                        <QueryName condition="end
with">aspnetcdn.com</QueryName> <!--Microsoft [
https://docs.microsoft.com/en-us/aspnet/ajax/cdn/overview ]-->
                        <QueryName
condition="is">ajax.googleapis.com</QueryName>
                        <QueryName
condition="is">cdnjs.cloudflare.com</QueryName> <!--Cloudflare: Hosts
popular javascript libraries-->
                        <QueryName
condition="is">fonts.googleapis.com</QueryName> <!--Google fonts-->
                        <QueryName condition="end
with">.typekit.net</QueryName> <!--Adobe fonts-->
                        <QueryName
condition="is">cdnjs.cloudflare.com</QueryName>
                        <QueryName condition="end
with">.stackassets.com</QueryName> <!--Stack Overflow-->
                        <QueryName condition="end
with">.steamcontent.com</QueryName>
                        <QueryName
condition="is">play.google.com</QueryName>
                        <QueryName condition="is">content-
autofill.googleapis.com</QueryName>
                        <!--Web resources-->
                        <QueryName condition="end
with">.disqus.com</QueryName> <!--Microsoft default exclusion-->
                        <QueryName condition="end
with">.fontawesome.com</QueryName>
```

```
                        <QueryName condition="is">disqus.com</QueryName>
<!--Microsoft default exclusion-->
                        <!--Ads-->
                        <QueryName condition="end
with">.1rx.io</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.2mdn.net</QueryName> <!--Ads: Google | Microsoft default
exclusion-->
                        <QueryName condition="end
with">.3lift.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.adadvisor.net</QueryName> <!--Ads: Neustar [
https://better.fyi/trackers/adadvisor.net/ ] -->
                        <QueryName condition="end
with">.adap.tv</QueryName> <!--Ads:AOL | Microsoft default exclusion [
https://www.crunchbase.com/organization/adap-tv ] -->
                        <QueryName condition="end
with">.addthis.com</QueryName> <!--Ads:Oracle | Microsoft default
exclusion [ https://en.wikipedia.org/wiki/AddThis ] -->
                        <QueryName condition="end
with">.adform.net</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.adnxs.com</QueryName> <!--Ads: AppNexus | Microsoft default
exclusion-->
                        <QueryName condition="end
with">.adroll.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.adrta.com</QueryName> <!--Ads-->
```

```
                    <QueryName condition="end
with">.adsafeprotected.com</QueryName> <!--Ads-->
                    <QueryName condition="end
with">.adsrvr.org</QueryName> <!--Ads-->
                    <QueryName condition="end
with">.adsymptotic.com</QueryName> <!--Ads-->
                    <QueryName condition="end
with">.advertising.com</QueryName> <!--Ads | Microsoft default
exclusion-->
                    <QueryName condition="end
with">.agkn.com</QueryName> <!--Ads | [
https://www.home.neustar/privacy ] -->
                    <QueryName condition="end with">.amazon-
adsystem.com</QueryName> <!--Ads-->
                    <QueryName condition="end with">.amazon-
adsystem.com</QueryName> <!--Ads-->
                    <QueryName condition="end
with">.analytics.yahoo.com</QueryName> <!--Ads:Yahoo-->
                    <QueryName condition="end
with">.aol.com</QueryName> <!--Ads | Microsoft default exclusion -->
                    <QueryName condition="end
with">.betrad.com</QueryName> <!--Ads | Microsoft default exclusion-->
                    <QueryName condition="end
with">.bidswitch.net</QueryName> <!--Ads-->
                    <QueryName condition="end
with">.casalemedia.com</QueryName> <!--Ads | Microsoft default
exclusion-->
```

```
                    <QueryName condition="end
with">.chartbeat.net</QueryName> <!--Ads | Microsoft default exclusion
[ https://better.fyi/trackers/chartbeat.com/ ]-->
                    <QueryName condition="end
with">.cnn.com</QueryName> <!-- Microsoft default exclusion-->
                    <QueryName condition="end
with">.convertro.com</QueryName> <!--Ads:Verizon-->
                    <QueryName condition="end
with">.criteo.com</QueryName> <!--Ads [
https://better.fyi/trackers/criteo.com/ ] -->
                    <QueryName condition="end
with">.criteo.net</QueryName> <!--Ads [
https://better.fyi/trackers/criteo.com/ ] -->
                    <QueryName condition="end
with">.crwdcntrl.net</QueryName> <!--Ads: Lotame [
https://better.fyi/trackers/crwdcntrl.net/ ] -->
                    <QueryName condition="end
with">.demdex.net</QueryName> <!--Ads | Microsoft default exclusion-->
                    <QueryName condition="end
with">.domdex.com</QueryName>
                    <QueryName condition="end
with">.dotomi.com</QueryName> <!--Ads | Microsoft default exclusion-->
                    <QueryName condition="end
with">.doubleclick.net</QueryName> <!--Ads:Conversant | Microsoft
default exclusion [ https://www.crunchbase.com/organization/dotomi ] --
>
                    <QueryName condition="end
with">.doubleverify.com</QueryName> <!--Ads: Google-->
```

```
                    <QueryName condition="end
with">.emxdgt.com</QueryName> <!--Ads: EMX-->
                    <QueryName condition="end
with">.everesttech.net</QueryName> <!--Ads | [
https://better.fyi/trackers/everesttech.net/ ] -->
                    <QueryName condition="end
with">.exelator.com</QueryName> <!--Ads:Nielson Marketing Cloud-->
                    <QueryName condition="end with">.google-
analytics.com</QueryName> <!--Ads:Google | Microsoft default exclusion-
->
                    <QueryName condition="end
with">.googleadservices.com</QueryName> <!--Google-->
                    <QueryName condition="end
with">.googlesyndication.com</QueryName> <!--Ads:Google, sometimes
called during malicious ads, but not directly responsible | Microsoft
default exclusion [ https://www.hackread.com/wp-
content/uploads/2018/06/Bitdefender-Whitepaper-Zacinlo.pdf ]-->
                    <QueryName condition="end
with">.googletagmanager.com</QueryName> <!--Google-->
                    <QueryName condition="end
with">.googlevideo.com</QueryName> <!--Google | Microsoft default
exclusion-->
                    <QueryName condition="end
with">.gstatic.com</QueryName> <!--Google | Microsoft default
exclusion-->
                    <QueryName condition="end
with">.gvt1.com</QueryName> <!--Google-->
                    <QueryName condition="end
with">.gvt2.com</QueryName> <!--Google-->
```

```
                        <QueryName condition="end with">.ib-
ibi.com</QueryName> <!--Ads: Offerpath [
https://better.fyi/trackers/ib-ibi.com/ ] -->
                        <QueryName condition="end
with">.jivox.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.krxd.net</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.lijit.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.mathtag.com</QueryName> <!--Microsoft default exclusion-->
                        <QueryName condition="end
with">.moatads.com</QueryName> <!--Ads | Microsoft default exclusion-->
                        <QueryName condition="end
with">.moatpixel.com</QueryName> <!--Ads | Microsoft default exclusion-
->
                        <QueryName condition="end
with">.mookie1.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.myvisualiq.net</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.netmng.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.nexac.com</QueryName> <!--Ads | Microsoft default exclusion-->
                        <QueryName condition="end
with">.openx.net</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.optimizely.com</QueryName> <!--Ads-->
```

```
                        <QueryName condition="end
with">.outbrain.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.pardot.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.phx.gbl</QueryName> <!--Ads | Microsoft default exclusion-->
                        <QueryName condition="end
with">.pinterest.com</QueryName> <!--Pinerest-->
                        <QueryName condition="end
with">.pubmatic.com</QueryName> <!--Ads | Microsoft default exclusion--
>
                        <QueryName condition="end
with">.quantcount.com</QueryName>
                        <QueryName condition="end
with">.quantserve.com</QueryName>
                        <QueryName condition="end
with">.revsci.net</QueryName> <!--Ads:Omniture | Microsoft default
exclusion-->
                        <QueryName condition="end
with">.rfihub.net</QueryName> <!--Ads | Microsoft default exclusion-->
                        <QueryName condition="end
with">.rlcdn.com</QueryName> <!--Ads: Rapleaf [
https://better.fyi/trackers/rlcdn.com/ ] -->
                        <QueryName condition="end
with">.rubiconproject.com</QueryName> <!--Ads: Rubicon Project |
Microsoft default exclusion [
https://better.fyi/trackers/rubiconproject.com/ ] -->
                        <QueryName condition="end
with">.scdn.co</QueryName> <!--Spotify-->
```

```
                        <QueryName condition="end
with">.scorecardresearch.com</QueryName> <!--Ads: Comscore | Microsoft
default exclusion-->
                        <QueryName condition="end with">.serving-
sys.com</QueryName> <!--Ads | Microsoft default exclusion-->
                        <QueryName condition="end
with">.sharethrough.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.simpli.fi</QueryName>
                        <QueryName condition="end
with">.sitescout.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.smartadserver.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.snapads.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.spotxchange.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.taboola.com</QueryName> <!--Ads:Taboola-->
                        <QueryName condition="end
with">.taboola.map.fastly.net</QueryName> <!--Ads:Taboola-->
                        <QueryName condition="end
with">.tapad.com</QueryName>
                        <QueryName condition="end
with">.tidaltv.com</QueryName> <!--Ads: Videology [
https://better.fyi/trackers/tidaltv.com/ ] -->
                        <QueryName condition="end
with">.trafficmanager.net</QueryName> <!--Ads | Microsoft default
exclusion-->
```

```
                        <QueryName condition="end
with">.tremorhub.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.tribalfusion.com</QueryName> <!--Ads: Exponential [
https://better.fyi/trackers/tribalfusion.com/ ] -->
                        <QueryName condition="end
with">.turn.com</QueryName> <!--Ads | Microsoft default exclusion [
https://better.fyi/trackers/turn.com/ ] -->
                        <QueryName condition="end
with">.twimg.com</QueryName> <!--Ads | Microsoft default exclusion-->
                        <QueryName condition="end
with">.tynt.com</QueryName> <!--Ads-->
                        <QueryName condition="end
with">.w55c.net</QueryName> <!--Ads:dataxu-->
                        <QueryName condition="end
with">.ytimg.com</QueryName> <!--Google-->
                        <QueryName condition="end
with">.zorosrv.com</QueryName> <!--Ads:Taboola-->
                        <QueryName condition="is">1rx.io</QueryName> <!--
Ads-->
                        <QueryName
condition="is">adservice.google.com</QueryName> <!--Google-->
                        <QueryName
condition="is">ampcid.google.com</QueryName> <!--Google-->
                        <QueryName
condition="is">clientservices.googleapis.com</QueryName> <!--Google-->
                        <QueryName
condition="is">googleadapis.l.google.com</QueryName> <!--Google-->
```

```
                        <QueryName
condition="is">imasdk.googleapis.com</QueryName> <!--Google [
https://developers.google.com/interactive-media-ads/docs/sdks/html5/ ]
-->
                        <QueryName condition="is">l.google.com</QueryName>
<!--Google-->
                        <QueryName condition="is">ml314.com</QueryName>
<!--Ads-->
                        <QueryName
condition="is">mtalk.google.com</QueryName> <!--Google-->
                        <QueryName
condition="is">update.googleapis.com</QueryName> <!--Google-->
                        <QueryName
condition="is">www.googletagservices.com</QueryName> <!--Google-->
                        <!--SocialNet-->
                        <QueryName condition="end
with">.pscp.tv</QueryName> <!--Twitter:Periscope-->
                        <!--OSCP/CRL Common-->
                        <QueryName condition="end
with">.amazontrust.com</QueryName>
                        <QueryName condition="end
with">.digicert.com</QueryName>
                        <QueryName condition="end
with">.globalsign.com</QueryName>
                        <QueryName condition="end
with">.globalsign.net</QueryName>
                        <QueryName condition="end
with">.intel.com</QueryName>
```

```
                        <QueryName condition="end
with">.symcb.com</QueryName> <!--Digicert-->
                        <QueryName condition="end
with">.symcd.com</QueryName> <!--Digicert-->
                        <QueryName condition="end
with">.thawte.com</QueryName>
                        <QueryName condition="end
with">.usertrust.com</QueryName>
                        <QueryName condition="end
with">.verisign.com</QueryName>
                        <QueryName condition="end
with">ocsp.identrust.com</QueryName>
                        <QueryName condition="end
with">pki.goog</QueryName>
                        <QueryName condition="is">msocsp.com</QueryName>
<!--Microsoft:OCSP-->
                        <QueryName
condition="is">ocsp.comodoca.com</QueryName>
                        <QueryName
condition="is">ocsp.entrust.net</QueryName>
                        <QueryName
condition="is">ocsp.godaddy.com</QueryName>
                        <QueryName condition="is">ocsp.int-
x3.letsencrypt.org</QueryName>
                        <QueryName
condition="is">ocsp.msocsp.com</QueryName> <!--Microsoft:OCSP-->
                        <QueryName condition="end
with">pki.goog</QueryName>
```

```
                        <QueryName
condition="is">ocsp.godaddy.com</QueryName>
                        <QueryName condition="end
with">amazontrust.com</QueryName>
                        <QueryName
condition="is">ocsp.sectigo.com</QueryName>
                        <QueryName condition="is">pki-
goog.l.google.com</QueryName>
                        <QueryName condition="end
with">.usertrust.com</QueryName>
                        <QueryName
condition="is">ocsp.comodoca.com</QueryName>
                        <QueryName
condition="is">ocsp.verisign.com</QueryName>
                        <QueryName
condition="is">ocsp.entrust.net</QueryName>
                        <QueryName condition="end
with">ocsp.identrust.com</QueryName>
                        <QueryName
condition="is">status.rapidssl.com</QueryName>
                        <QueryName
condition="is">status.thawte.com</QueryName>
                        <QueryName condition="is">ocsp.int-
x3.letsencrypt.org</QueryName>
                </DnsQuery>
        </RuleGroup>

        <!--SYSMON EVENT ID 23 : FILE DELETE [FileDelete]-->
                <!--EVENT 22: "File Delete"-->
```

<!--COMMENT: Sandbox usage. When a program signals to Windows a file should be deleted or wiped, Sysmon may be able to capture it.

[https://isc.sans.edu/forums/diary/Sysmon+and+File+Deletion/26084/]

-->

<!--DATA: RuleName, UtcTime, ProcessGuid, ProcessId, User, Image, TargetFilename, Hashes, IsExecutable, Archived -->

<!--
<RuleGroup name="" groupRelation="or">
<ClipboardChange onmatch="include">
</ClipboardChange>
</RuleGroup>
-->

<!--SYSMON EVENT ID 24 : CLIPBOARD EVENT MONITORING [ClipboardChange]-->

<!--EVENT 24: "Clipboard changed"-->

<!--COMMENT: Sandbox usage. Sysmon can capture the contents of clipboard events.

An example of what could be a production usage on restricted desktops is provided below, but it is commented-out. -->

<!--DATA: EventType, UtcTime, ProcessGuid, ProcessId, Image, Session, ClientInfo, Hashes, Archived -->

<!--

```
        <RuleGroup name="" groupRelation="or">
                <ClipboardChange onmatch="include">
                        <Image condition="end with">wscript.exe</Image>
                        <Image condition="end with">cscript.exe</Image>
                        <Image condition="end with">powershell.exe</Image>
                        <Image condition="end with">rdpclip.exe</Image>
                </ClipboardChange>
        </RuleGroup>
        -->

        <!--SYSMON EVENT ID 25 : PROCESS TAMPERING [ProcessTampering]-->
                <!--EVENT 25: "Process Tampering"-->
                <!--COMMENT:  This event is generated when a process
image is changed from an external source, such as a different process.
                        This may or may not provide value in your
environment as it requires tuning and a SIEM to correlate the
ProcessGuids.
                        [ https://medium.com/falconforce/sysmon-13-
process-tampering-detection-820366138a6c ] -->

                <!--DATA: EventType, RuleName, UtcTime, ProcessGuid,
ProcessId, Image, Type -->

        <!--
        <RuleGroup name="" groupRelation="or">
                <ProcessTampering onmatch="exclude">
                        <Image condition="begin with">C:\Program Files
(x86)\Microsoft\Edge\Application\</Image>
                </ProcessTampering>
```

```
        </RuleGroup>
        -->

        <!--SYSMON EVENT ID 255 : ERROR-->
                <!--"This event is generated when an error occurred
within Sysmon. They can happen if the system is under heavy load
                        and certain tasked could not be performed or a bug
exists in the Sysmon service. You can report any bugs on the
                        Sysinternals forum or over Twitter
(@markrussinovich)."-->
                <!--Cannot be filtered.-->

        </EventFiltering>
</Sysmon>
```

7.7 G: Ossec.conf de endpoints windows

```
<!--
  Wazuh - Agent - Default configuration for Windows
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>

  <client>
    <server>
      <address>192.168.6.97</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>windows, windows10</config-profile>
    <crypto_method>aes</crypto_method>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <enrollment>
      <enabled>yes</enabled>
      <agent_name>Wazuh-Agent-ECOUIOPC2766</agent_name>
      <groups>Ecoilpet</groups>
    </enrollment>
  </client>


  <!-- Agent buffer options -->
```

```
  <client_buffer>
    <disabled>no</disabled>
    <queue_size>5000</queue_size>
    <events_per_second>500</events_per_second>
  </client_buffer>

  <!-- Log analysis -->
  <localfile>
    <location>Application</location>
    <log_format>eventchannel</log_format>
  </localfile>

  <localfile>
    <location>Security</location>
    <log_format>eventchannel</log_format>
    <query>Event/System[EventID != 5145 and EventID != 5156 and EventID
!= 5447 and
      EventID != 4656 and EventID != 4658 and EventID != 4663 and
EventID != 4660 and
      EventID != 4670 and EventID != 4690 and EventID != 4703 and
EventID != 4907 and
      EventID != 5152 and EventID != 5157]</query>
  </localfile>

  <localfile>
    <location>System</location>
    <log_format>eventchannel</log_format>
  </localfile>
```

```
  <localfile>
    <location>active-response\active-responses.log</location>
    <log_format>syslog</log_format>
  </localfile>

<localfile>
<location>Microsoft-Windows-Sysmon/Operational</location>
<log_format>eventchannel</log_format>
</localfile>

  <!-- Policy monitoring -->
  <rootcheck>
    <disabled>no</disabled>
    <windows_apps>./shared/win_applications_rcl.txt</windows_apps>
    <windows_malware>./shared/win_malware_rcl.txt</windows_malware>
  </rootcheck>

  <!-- Security Configuration Assessment -->
  <sca>
    <enabled>yes</enabled>
    <scan_on_start>yes</scan_on_start>
    <interval>12h</interval>
    <skip_nfs>yes</skip_nfs>
  </sca>

  <!-- File integrity monitoring -->
  <syscheck>

    <disabled>no</disabled>
```

```
  <!-- Virus Total Integration -->
<directories
realtime="yes">C:\Users\PASANTE_SI_2\Downloads</directories>

    <!-- Frequency that syscheck is executed default every 12 hours -->
    <frequency>43200</frequency>

    <!-- Default files to be monitored. -->
    <directories recursion_level="0"
restrict="regedit.exe$|system.ini$|win.ini$">%WINDIR%</directories>

    <directories recursion_level="0"
restrict="at.exe$|attrib.exe$|cacls.exe$|cmd.exe$|eventcreate.exe$|ftp.
exe$|lsass.exe$|net.exe$|net1.exe$|netsh.exe$|reg.exe$|regedt32.exe|reg
svr32.exe|runas.exe|sc.exe|schtasks.exe|sethc.exe|subst.exe$">%WINDIR%\
SysNative</directories>
    <directories
recursion_level="0">%WINDIR%\SysNative\drivers\etc</directories>
    <directories recursion_level="0"
restrict="WMIC.exe$">%WINDIR%\SysNative\wbem</directories>
    <directories recursion_level="0"
restrict="powershell.exe$">%WINDIR%\SysNative\WindowsPowerShell\v1.0</d
irectories>
    <directories recursion_level="0"
restrict="winrm.vbs$">%WINDIR%\SysNative</directories>

    <!-- 32-bit programs. -->
```

```
    <directories recursion_level="0"
restrict="at.exe$|attrib.exe$|cacls.exe$|cmd.exe$|eventcreate.exe$|ftp.
exe$|lsass.exe$|net.exe$|net1.exe$|netsh.exe$|reg.exe$|regedit.exe$|reg
edt32.exe$|regsvr32.exe$|runas.exe$|sc.exe$|schtasks.exe$|sethc.exe$|su
bst.exe$">%WINDIR%\System32</directories>
    <directories
recursion_level="0">%WINDIR%\System32\drivers\etc</directories>
    <directories recursion_level="0"
restrict="WMIC.exe$">%WINDIR%\System32\wbem</directories>
    <directories recursion_level="0"
restrict="powershell.exe$">%WINDIR%\System32\WindowsPowerShell\v1.0</di
rectories>
    <directories recursion_level="0"
restrict="winrm.vbs$">%WINDIR%\System32</directories>

    <directories realtime="yes">%PROGRAMDATA%\Microsoft\Windows\Start
Menu\Programs\Startup</directories>

    <ignore>%PROGRAMDATA%\Microsoft\Windows\Start
Menu\Programs\Startup\desktop.ini</ignore>

    <ignore
type="sregex">.log$|.htm$|.jpg$|.png$|.chm$|.pnf$|.evtx$</ignore>

    <!-- Windows registry entries to monitor. -->

<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\batfile</windows_
registry>
```

```
<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\cmdfile</windows_
registry>

<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\comfile</windows_
registry>

<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\exefile</windows_
registry>

<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\piffile</windows_
registry>

<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\AllFilesystemObje
cts</windows_registry>

<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\Directory</window
s_registry>

<windows_registry>HKEY_LOCAL_MACHINE\Software\Classes\Folder</windows_r
egistry>
    <windows_registry
arch="both">HKEY_LOCAL_MACHINE\Software\Classes\Protocols</windows_regi
stry>
    <windows_registry
arch="both">HKEY_LOCAL_MACHINE\Software\Policies</windows_registry>
    <windows_registry>HKEY_LOCAL_MACHINE\Security</windows_registry>
```

```
    <windows_registry
arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Internet
Explorer</windows_registry>


<windows_registry>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services<
/windows_registry>

<windows_registry>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\S
ession Manager\KnownDLLs</windows_registry>

<windows_registry>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\S
ecurePipeServers\winreg</windows_registry>

    <windows_registry
arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersio
n\Run</windows_registry>
    <windows_registry
arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersio
n\RunOnce</windows_registry>

<windows_registry>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Current
Version\RunOnceEx</windows_registry>
    <windows_registry
arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersio
n\URL</windows_registry>
    <windows_registry
arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersio
n\Policies</windows_registry>
```

```
    <windows_registry
arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows
NT\CurrentVersion\Windows</windows_registry>
    <windows_registry
arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon</windows_registry>

    <windows_registry
arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Active
Setup\Installed Components</windows_registry>

    <!-- Windows registry entries to ignore. -->

<registry_ignore>HKEY_LOCAL_MACHINE\Security\Policy\Secrets</registry_i
gnore>

<registry_ignore>HKEY_LOCAL_MACHINE\Security\SAM\Domains\Account\Users<
/registry_ignore>
    <registry_ignore type="sregex">\Enum$</registry_ignore>

<registry_ignore>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\M
psSvc\Parameters\AppCs</registry_ignore>

<registry_ignore>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\M
psSvc\Parameters\PortKeywords\DHCP</registry_ignore>

<registry_ignore>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\M
psSvc\Parameters\PortKeywords\IPTLSIn</registry_ignore>
```

```
<registry_ignore>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\M
psSvc\Parameters\PortKeywords\IPTLSOut</registry_ignore>

<registry_ignore>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\M
psSvc\Parameters\PortKeywords\RPC-EPMap</registry_ignore>

<registry_ignore>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\M
psSvc\Parameters\PortKeywords\Teredo</registry_ignore>

<registry_ignore>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\P
olicyAgent\Parameters\Cache</registry_ignore>

<registry_ignore>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentV
ersion\RunOnceEx</registry_ignore>

<registry_ignore>HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\A
DOVMPPackage\Final</registry_ignore>

    <!-- Frequency for ACL checking (seconds) -->
    <windows_audit_interval>60</windows_audit_interval>

    <!-- Nice value for Syscheck module -->
    <process_priority>10</process_priority>

    <!-- Maximum output throughput -->
    <max_eps>50</max_eps>

    <!-- Database synchronization settings -->
```

```
  <synchronization>
    <enabled>yes</enabled>
    <interval>5m</interval>
    <max_eps>10</max_eps>
  </synchronization>
</syscheck>

<!-- System inventory -->
<wodle name="syscollector">
  <disabled>no</disabled>
  <interval>1h</interval>
  <scan_on_start>yes</scan_on_start>
  <hardware>yes</hardware>
  <os>yes</os>
  <network>yes</network>
  <packages>yes</packages>
  <ports all="no">yes</ports>
  <processes>yes</processes>

  <!-- Database synchronization settings -->
  <synchronization>
    <max_eps>10</max_eps>
  </synchronization>
</wodle>

<!-- CIS policies evaluation -->
<wodle name="cis-cat">
  <disabled>yes</disabled>
  <timeout>1800</timeout>
```

```
    <interval>1d</interval>
    <scan-on-start>yes</scan-on-start>

    <java_path>\\server\jre\bin\java.exe</java_path>
    <ciscat_path>C:\cis-cat</ciscat_path>
  </wodle>

  <!-- Osquery integration -->
  <wodle name="osquery">
    <disabled>yes</disabled>
    <run_daemon>yes</run_daemon>
    <bin_path>C:\Program Files\osquery\osqueryd</bin_path>
    <log_path>C:\Program
Files\osquery\log\osqueryd.results.log</log_path>
    <config_path>C:\Program Files\osquery\osquery.conf</config_path>
    <add_labels>yes</add_labels>
  </wodle>

  <!-- Active response -->
  <active-response>
    <disabled>no</disabled>
    <ca_store>wpk_root.pem</ca_store>
    <ca_verification>yes</ca_verification>
  </active-response>

  <!-- Choose between plain or json format (or both) for internal logs
-->
  <logging>
    <log_format>plain</log_format>
```

```
  </logging>

</ossec_config>

<!-- END of Default Configuration. -->
```

7.8 H: Remove threat.py

El archivo completo está disponible en:

```
# Copyright (C) 2015-2025, Wazuh Inc.
# All rights reserved.

import os
import sys
import json
import datetime
import stat
import tempfile
import pathlib

if os.name == 'nt':
    LOG_FILE = "C:\\Program Files (x86)\\ossec-agent\\active-
response\\active-responses.log"
else:
    LOG_FILE = "/var/ossec/logs/active-responses.log"

ADD_COMMAND = 0
DELETE_COMMAND = 1
CONTINUE_COMMAND = 2
ABORT_COMMAND = 3

OS_SUCCESS = 0
OS_INVALID = -1

class message:
```

```
    def __init__(self):
        self.alert = ""
        self.command = 0


def write_debug_file(ar_name, msg):
    with open(LOG_FILE, mode="a") as log_file:
        log_file.write(str(datetime.datetime.now().strftime('%Y/%m/%d
%H:%M:%S')) + " " + ar_name + ": " + msg +"\n")


def setup_and_check_message(argv):
    input_str = ""
    for line in sys.stdin:
        input_str = line
        break


    msg_obj = message()
    try:
        data = json.loads(input_str)
    except ValueError:
        write_debug_file(argv[0], 'Decoding JSON has failed, invalid
input format')
        msg_obj.command = OS_INVALID
        return msg_obj


    msg_obj.alert = data
    command = data.get("command")


    if command == "add":
        msg_obj.command = ADD_COMMAND
```

```
    elif command == "delete":
        msg_obj.command = DELETE_COMMAND
    else:
        msg_obj.command = OS_INVALID
        write_debug_file(argv[0], 'Not valid command: ' + command)

    return msg_obj

def send_keys_and_check_message(argv, keys):
    keys_msg = json.dumps({"version": 1,"origin":{"name":
argv[0],"module":"active-
response"},"command":"check_keys","parameters":{"keys":keys}})
    write_debug_file(argv[0], keys_msg)

    print(keys_msg)
    sys.stdout.flush()

    input_str = ""
    while True:
        line = sys.stdin.readline()
        if line:
            input_str = line
            break

    try:
        data = json.loads(input_str)
    except ValueError:
        write_debug_file(argv[0], 'Decoding JSON has failed, invalid
input format')
```

```
        return OS_INVALID

    action = data.get("command")
    if action == "continue":
        return CONTINUE_COMMAND
    elif action == "abort":
        return ABORT_COMMAND
    else:
        write_debug_file(argv[0], "Invalid value of 'command'")
        return OS_INVALID

def secure_delete_file(filepath_str, ar_name):
    filepath = pathlib.Path(filepath_str)

    # Reject NTFS alternate data streams
    if '::' in filepath_str:
        raise Exception(f"Refusing to delete ADS or NTFS stream:
{filepath_str}")

    # Reject symbolic links and reparse points
    if os.path.islink(filepath):
        raise Exception(f"Refusing to delete symbolic link:
{filepath}")

    attrs = os.lstat(filepath).st_file_attributes
    if attrs & stat.FILE_ATTRIBUTE_REPARSE_POINT:
        raise Exception(f"Refusing to delete reparse point:
{filepath}")
```

```
    resolved_filepath = filepath.resolve()

    # Ensure it's a regular file
    if not resolved_filepath.is_file():
        raise Exception(f"Target is not a regular file:
{resolved_filepath}")

  # Perform deletion
    os.remove(resolved_filepath)

def main(argv):
    write_debug_file(argv[0], "Started")
    msg = setup_and_check_message(argv)

    if msg.command < 0:
        sys.exit(OS_INVALID)

    if msg.command == ADD_COMMAND:
        alert = msg.alert["parameters"]["alert"]
        keys = [alert["rule"]["id"]]
        action = send_keys_and_check_message(argv, keys)

        if action != CONTINUE_COMMAND:
            if action == ABORT_COMMAND:
                write_debug_file(argv[0], "Aborted")
                sys.exit(OS_SUCCESS)
            else:
                write_debug_file(argv[0], "Invalid command")
                sys.exit(OS_INVALID)
```

```
        try:
            file_path = alert["data"]["virustotal"]["source"]["file"]
            if os.path.exists(file_path):
                secure_delete_file(file_path, argv[0])
                write_debug_file(argv[0], json.dumps(msg.alert) + "
Successfully removed threat")
            else:
                write_debug_file(argv[0], f"File does not exist:
{file_path}")
        except OSError as error:
            write_debug_file(argv[0], json.dumps(msg.alert) + "Error
removing threat")
          except Exception as e:
            write_debug_file(argv[0], f"{json.dumps(msg.alert)}: Error
removing threat: {str(e)}")
    else:
        write_debug_file(argv[0], "Invalid command")

    write_debug_file(argv[0], "Ended")
    sys.exit(OS_SUCCESS)

if __name__ == "__main__":
    main(sys.argv)
```

7.9 I: Abuse.py

```
import re
import requests
import os
import sys
import time
from datetime import datetime

# === CONFIGURACIÓN ===
API_KEY = "REDACTED"
LOG_FILE = "/var/ossec/logs/alerts/alerts.log"
BASE_DIR = "/home/userconfig/abuseIP"

BAN_LIST_FILE = os.path.join(BASE_DIR, "ban.txt")
CHECKED_FILE = os.path.join(BASE_DIR, "checked_today.txt")
BAD_TODAY_FILE = os.path.join(BASE_DIR, "bad_today.txt")
LOG_OUTPUT_FILE = os.path.join(BASE_DIR, "abuselog.txt")

HEADERS = {"Key": API_KEY, "Accept": "application/json"}

# === LOGGING EN CONSOLA Y ARCHIVO ===
class TeeLogger:
    def __init__(self, filename):
        self.terminal = sys.stdout
        self.log = open(filename, "a")

    def write(self, message):
        self.terminal.write(message)
        self.log.write(message)
```

```
    def flush(self):
        self.terminal.flush()
        self.log.flush()

sys.stdout = TeeLogger(LOG_OUTPUT_FILE)

# === FUNCIONES ===
def load_file_as_set(path):
    if not os.path.exists(path):
        return set()
    with open(path, "r") as f:
        return set(line.strip() for line in f if line.strip())

def append_line(path, line):
    with open(path, "a") as f:
        f.write(line + "\n")

def write_lines(path, lines):
    with open(path, "w") as f:
        for line in lines:
            f.write(line + "\n")

def check_ip(ip):
    url = "https://api.abuseipdb.com/api/v2/check"
    params = {"ipAddress": ip, "maxAgeInDays": 90}
    try:
        response = requests.get(url, headers=HEADERS, params=params)
        data = response.json()
```

```
        if data.get("data", {}).get("abuseConfidenceScore", 0) > 0:
            return data["data"]
    except Exception as e:
        print(f"[!] ERROR al consultar IP {ip}: {e}")
    return None

# === MAIN ===
def main():
    print(f"\n========== EJECUCIÓN INICIADA: {datetime.now()}
==========")

    checked_ips = load_file_as_set(CHECKED_FILE)
    banned_ips = load_file_as_set(BAN_LIST_FILE)

    total_ips_detectadas = 0
    total_ips_revisadas = 0
    total_ips_maliciosas = 0

    bad_today = set()

    if not os.path.exists(LOG_FILE):
        print(f"[!] ERROR: El archivo de logs no existe: {LOG_FILE}")
        return

    print(f"[i] Cargando IPs ya consultadas hoy: {len(checked_ips)}")
    print(f"[i] Cargando IPs ya baneadas históricamente:
{len(banned_ips)}")
    print(f"[i] Procesando archivo de log completo: {LOG_FILE}")
```

```
    with open(LOG_FILE, "r") as log_file:
        for line in log_file:
            if "CFC_QUITO" in line and "src=" in line:
                match = re.search(r"src=([\d.]+)", line)
                if match:
                    ip = match.group(1)
                    total_ips_detectadas += 1

                    if ip in checked_ips:
                        print(f"[=] IP {ip} ya fue revisada hoy.
Saltando.")
                        continue

                    if ip in banned_ips:
                        print(f"[=] IP {ip} ya está en ban.txt.
Saltando.")
                        continue

                    print(f"[+] Revisando nueva IP: {ip}")
                    append_line(CHECKED_FILE, ip)
                    total_ips_revisadas += 1

                    result = check_ip(ip)
                    if result:
                        score = result.get("abuseConfidenceScore", 0)
                        print(f"[!] IP maliciosa detectada: {ip} -
Score: {score}")
                        if score >= 50:
                            append_line(BAN_LIST_FILE, ip)
```

```
                            bad_today.add(ip)
                            total_ips_maliciosas += 1
                            print(f"[+] IP {ip} guardada en ban.txt por
score >= 50")
                        else:
                            print(f"[i] IP {ip} no guardada en ban.txt
porque score < 50")
                    else:
                        print(f"[i] IP limpia o sin respuesta: {ip}")

    if bad_today:
        write_lines(BAD_TODAY_FILE, sorted(bad_today))
        print(f"[i] Guardadas {len(bad_today)} IPs malas nuevas en
{BAD_TODAY_FILE}")
    else:
        print("[i] No se detectaron IPs malas nuevas hoy.")

    print("\n========== EJECUCIÓN FINALIZADA ==========")
    print(f"Total IPs detectadas en logs: {total_ips_detectadas}")
    print(f"Total IPs revisadas vía API: {total_ips_revisadas}")
    print(f"Total IPs maliciosas agregadas a ban.txt:
{total_ips_maliciosas}")
    print("=========================================\n")

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