



NEREA GÓMEZ BARRASA

DIEGO MARTÍN OLEA

FRAY JOSÉ ÁVILA HERNÁNDEZ

UN ENFOQUE INTEGRADO PARA LA GESTIÓN DE AMENAZAS: USO DE ELK Y MISP PARA ENRIQUECER UN SIEM CON IOCS DE SMOKELOADER

TABLA DE CONTENIDOS

❖ **Introducción**

- Contexto y justificación del proyecto
- Objetivos del proyecto

❖ **Herramientas y Tecnologías utilizadas**

- Descripción de ELK Stack
- MISP y su integración
- Características clave del malware SmokeLoader

❖ **Metodología**

- Recopilación de IOCs de SmokeLoader
- Configuración de MISP para integración con ELK: Desarrollo de reglas ELK a partir de IOCs

❖ **Resultados y Análisis**

- Evaluación de la detección mejorada
- Impacto en la respuesta ante amenazas
- Lecciones aprendidas

❖ **Conclusiones y recomendaciones**

- Logros y contribuciones
- Limitaciones y Desafíos
- Sugerencias para futuras mejoras

❖ **Anexos**

- Configuraciones detalladas
- Capturas de pantalla relevantes

INTRODUCCIÓN

a. Contexto y justificación del proyecto

En el mundo actual, las amenazas a las que nos enfrentamos a diario evolucionan constantemente, y la detección y respuesta efectivas a las mismas son esenciales para proteger los activos y la información de las organizaciones. Los Sistemas de Información y Gestión de Eventos de Seguridad, más conocidos como SIEM, son herramientas clave que permiten la recopilación, correlación y análisis de datos de seguridad para identificar posibles incidentes. No obstante, la efectividad de un SIEM depende de la calidad y actualización de la información que procesa.

En este contexto, la aparición de malwares específicos, como el objeto de investigación, el *SmokeLoader*, introduce nuevos desafíos. *SmokeLoader* es conocido por sus tácticas evasivas y su capacidad para eludir las detecciones tradicionales. Enfrentar esta amenaza requiere no sólo la comprensión de su comportamiento, es decir, de sus tácticas, técnicas y procedimientos, sino también la capacidad de incorporar inteligencia de amenazas actualizada de manera proactiva en nuestro sistema de seguridad.

De aquí nace la necesidad crítica de mejorar la capacidad de detección y respuesta de un SIEM, específicamente frente a amenazas como *SmokeLoader*. Integrar la inteligencia de amenazas a través de OSINT y servicios como MISP puede proporcionar una ventaja estratégica.

El análisis de IOCs (indicadores de compromiso) específicos de *SmokeLoader* y su conversión en reglas para ELK permitirá una detección más precisa y oportuna de actividades maliciosas. Esta mejora es vital de cara a la organización, puesto que reduce el tiempo de detección y minimiza el impacto potencial de un incidente de seguridad.

La integración de MISP y ELK proporciona una solución integral para abordar este desafío.

b. Objetivos del proyecto

- Integración de MISP con ELK:
 - Configurar la integración entre la plataforma MISP (*Malware Information Sharing Platform & Threat Sharing*) y ELK (*Elasticsearch, Logstash, Kibana*) para permitir el intercambio de información de amenazas y su análisis eficiente.
- Recopilación de IOCs de *SmokeLoader*:

- Identificar, recopilar y analizar Indicadores de Compromiso (IOCs) específicos de SmokeLoader para comprender sus tácticas, técnicas y procedimientos.
- Desarrollo de reglas ELK a partir de IOCs:
 - Convertir los IOCs identificados en reglas de detección para ELK, permitiendo la identificación automatizada de actividades maliciosas relacionadas con SmokeLoader.
- Implementación y pruebas en un entorno de laboratorio:
 - Desplegar y probar las configuraciones y reglas desarrolladas en un entorno de laboratorio simulado (un equipo monitorizado por el SIEM) para evaluar la eficacia y la precisión de la detección.
- Mejora de la detección en el SIEM:
 - Lograr una mejora significativa en la capacidad de detección del SIEM al integrar información actualizada sobre amenazas, específicamente relacionada con SmokeLoader.
- Análisis de resultados y desempeño:
 - Evaluar y analizar los resultados obtenidos durante las pruebas, examinando el desempeño del sistema mejorado en términos de tiempo de detección, tasa de falsos positivos/negativos y eficacia general.
- Documentación y guía de configuración:
 - En el presente informe se ha creado documentación detallada que explica la configuración de la integración entre MISP y ELK, así como guía paso a paso para la replicación de los resultados en entornos similares.
- Presentación de conclusiones y recomendaciones:
 - Resumir las conclusiones del proyecto, destacando los logros, las limitaciones identificadas y proporcionando recomendaciones para futuras mejoras o investigaciones.
- Validación del impacto en la resiliencia en el campo de la Ciberseguridad:
 - Validar cómo la implementación de la inteligencia de amenazas específicas mejora la resiliencia en la ciberseguridad de la organización, especialmente en la detección y respuesta a amenazas como SmokeLoader.

Estos objetivos se centran en la integración de inteligencia de amenazas específica en un entorno SIEM, con un enfoque particular en el malware SmokeLoader, con el fin de mejorar la capacidad de detección y respuesta ante amenazas cibernéticas.

HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

a. Descripción del SIEM

Elasticsearch es el componente fundamental y principal de ELK Stack. Sirve como un potente motor de búsqueda y análisis distribuido. Este sistema escalable y de código abierto permite almacenar, indexar y recuperar datos en tiempo real. La capacidad de Elasticsearch para adaptarse a grandes volúmenes de datos, junto con su eficiente capacidad de búsqueda, lo convierte en el corazón de la gestión de registros y el análisis de información.

Logstash complementa a Elasticsearch al proporcionar capacidades de ingestión y transformación de datos. Sus diversos plugins de entrada, filtro y salida facilitan la normalización y enriquecimiento de datos, preparándolos para su posterior almacenamiento y análisis en Elasticsearch. Este módulo se convierte en un componente crucial para la preparación y estructuración de datos provenientes de diferentes fuentes.

Kibana se presenta como la interfaz de usuario web que completa ELK Stack. Esta herramienta visual nos permite explorar y visualizar datos almacenados en Elasticsearch de manera intuitiva. Con Kibana, podemos crear informes, paneles y visualizaciones personalizadas para comprender y comunicar de manera eficaz la información. Su capacidad para proporcionar representaciones visuales simplifica la interpretación de datos complejos.

Wazuh es una plataforma de seguridad *open source* que funciona como EDR (Endpoint Detection and Response) en un SIEM. Sus agentes se instalan y recopilan eventos de logs, y el servidor central analiza, detecta amenazas y responde en tiempo real. Se integra con ELK Stack, aprovechando Elasticsearch para almacenar y buscar datos, Logstash para la ingestión y transformación, y Kibana para tener una buena interfaz de visualización. Wazuh proporciona reglas predefinidas, análisis de vulnerabilidades, respuesta a incidentes y la capacidad de integrarse con inteligencia de amenazas. Fortalece la detección y respuesta a amenazas en entornos de SIEM.

El módulo **MISP** fortalece la inteligencia de amenazas al integrar información sobre indicadores de compromiso (IOCs). Al vincular MISP con ELK Stack, se facilita el intercambio y análisis de datos sobre amenazas específicas. Esto nos permite la incorporación proactiva de inteligencia actualizada en el sistema, mejorando la capacidad de detección y respuesta frente a estas amenazas específicas, como SmokeLoader.

La combinación de Elasticsearch, Logstash, Kibana, Wazuh y MISP ofrece un entorno completo y robusto para la gestión de registros y la inteligencia de amenazas. Este conjunto de herramientas no solo mejora la visibilidad de los datos de seguridad, sino que también refuerza la capacidad de respuesta ante amenazas, ofreciendo un enfoque integral para la seguridad cibernética en entornos empresariales.

b. MISP y qué supone su integración en el SIEM

La vinculación de MISP y ELK amplía las capacidades del SIEM, permitiendo la integración de información contextual y relevante sobre amenazas específicas. Esto no solo mejora la eficacia del sistema, sino que también proporciona una visión más completa de la amenaza, facilitando la toma de decisiones informada y la respuesta rápida.

c. Características clave del malware SmokeLoader

SmokeLoader se distingue por su enfoque en campañas de spam y phishing, a menudo propagándose a través de archivos ZIP o RAR que albergan componentes maliciosos. Sus capacidades abarcan desde el robo de credenciales hasta actividades más sofisticadas como el minado de criptomonedas y la distribución de ransomware y otras aplicaciones maliciosas. La continua adaptación y sofisticación de sus tácticas hacen que su detección sea cada vez más desafiante.

En el análisis dinámico de la muestra mediante CAPEv2, se revela que el malware bajo escrutinio exhibe una sofisticada variedad de técnicas diseñadas para eludir la detección, manipular la gestión de permisos y seguridad, así como controlar la entrada/salida de archivos. Estas tácticas incluyen el uso de scripts como *sleep*, *SetThreadIdealProcessor*, *SetKernelObjectSecurity*, *GetStdHandle*... Este malware se clasifica como una *backdoor*, focalizándose en recopilar información sobre la inicialización del programa y el entorno, mientras monitorea la consola del sistema mediante funciones como *GetCommandLineA* y *ReadConsoleOutputCharacterW*. Su enfoque principal reside en la manipulación del sistema a través de la consola y la recopilación de datos, incluyendo la identificación del alias de la consola, la adaptación al sistema operativo correspondiente, la obtención de datos sobre usuarios, ventanas y escritorios, la manipulación y carga de bibliotecas DLL, y el control del manejo de archivos.

Adicionalmente, el malware puede abrir el manejo de hilos para la memoria, evitando así escribir en la consola simultáneamente con la víctima o viceversa, asegurando la efectividad de sus acciones. Asimismo, busca recopilar información exhaustiva sobre el sistema operativo, su versión, conexiones de red, archivos, librerías y alias de usuarios. Posteriormente, emplea diversas técnicas para manipular la consola, orientadas a alterar datos de respaldo y obtener información sobre distintos perfiles de usuario. Todo sugiere que el objetivo final de esta amenaza es el robo de credenciales.

En relación a los dos payloads asociados al malware, es plausible que estén diseñados para evitar la detección del malware principal, incorporando funcionalidades anti-debugging y anti-sandbox en su arsenal defensivo.

Por otro lado, los resultados MITRE (organización sin fines de lucro, se ha destacado como pionera en el desarrollo del marco de conocimiento ATT&CK (*Adversarial Tactics, Techniques, and Common Knowledge*)) en el análisis de malware proporcionan una valiosa hoja de ruta que detalla las tácticas, técnicas y procedimientos utilizados por los atacantes.

Estos resultados son instrumentales para fortalecer las defensas cibernéticas, permitiendo a las organizaciones adaptar sus estrategias de seguridad de manera precisa y eficaz frente a las amenazas actuales. Estos resultados son los IOCs que registraremos en MISP y que nos ayudarán a nutrir nuestro SIEM para alcanzar ese margen de mejora en la detección y respuesta.

Podemos tener acceso a los IOCs de nuestro evento descargándolos como .json o .xml desde el mismo entorno de MISP.

En conjunto, *SmokeLoader* se revela como un malware altamente avanzado y adaptativo, con un enfoque claro en el robo de credenciales. Su capacidad para eludir la detección, manipular procesos y establecer conexiones cifradas presenta un desafío significativo para la seguridad. La identificación de firmas, comandos y procesos específicos proporciona una base sólida para la implementación de contramedidas y fortalecimiento de las defensas contra esta amenaza.

El comportamiento intrusivo del malware en consideración nos ha proporcionado una rica fuente de datos, aunque su enfoque evasivo ha presentado desafíos en el rastreo. La muestra de malware, como previamente

examinado, establece múltiples conexiones con el objetivo de eludir la detección, lo que añade complejidad al seguimiento de su trayectoria. A pesar de estas tácticas evasivas, se ha implementado un escaneo exhaustivo de IPs y URLs sospechosas en un esfuerzo por identificar la botnet subyacente al malware.

El resultado de estas investigaciones revela la presencia de dos botnets distintas, identificadas como "pub1" y "autm", distribuidas a través de diversos dominios. Este hallazgo proporciona una visión más clara de la infraestructura utilizada por los actores maliciosos, permitiéndonos comprender mejor la magnitud y sofisticación de la amenaza. La identificación de las botnets "pub1" y "autm" bajo varios dominios representa un avance significativo en el entendimiento del ecosistema malicioso. El mapeo detallado de estas botnets podría incluir la recopilación de información sobre su estructura, alcance geográfico, y métodos de comunicación. Este proceso es esencial para implementar medidas de mitigación específicas y colaborar con las autoridades pertinentes en la desarticulación de estas redes maliciosas.

METODOLOGÍA

a. Recopilación de IOCs de SmokeLoader

El proceso de análisis se ha llevado a cabo empleando múltiples herramientas en distintas fases. Inicialmente, se recurrió a herramientas en línea como VirusTotal, Viper, y Joesandbox para obtener datos estáticos del malware y realizar una primera evaluación de la muestra.

Para un entendimiento más profundo y avanzado, se ha utilizado CAPEv2, la última iteración del framework de CAPE Sandbox. Esta herramienta de código libre automatiza el análisis dinámico del malware, proporcionando una visión detallada de su comportamiento y funcionalidades.

El análisis de la red se ha llevado a cabo utilizando Virustotal y Tria.ge, donde se realizaron escaneos exhaustivos de diversas IPs y URLs sospechosas con el objetivo de identificar y mapear las posibles botnets asociadas al malware.

El evento analizado ha sido registrado en MISP (Malware Information Sharing Platform & Threat Sharing), consolidando así la información obtenida y permitiendo el registro de los IOCs de cara a integrarlos en nuestro ELK.

b. Configuración de MISP para integración con ELK: Desarrollo de reglas ELK a partir de IOCs

Durante el desarrollo de la configuración e integración de MISP con el ELK, hemos tenido varios hándicaps y problemas técnicos que hemos ido solventando a medida que avanzaba el proyecto.

b.1. Primeros pasos: instalación del entorno de monitorización

En un primer momento, para llevar a cabo la monitorización del malware SmokeLoader, hemos optado por utilizar Wazuh. El proceso de instalación se inicia mediante el siguiente comando:

```
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash  
./wazuh-install.sh -a -i
```

Una vez completada la instalación, se nos proporciona un usuario y contraseña para acceder a la interfaz de Wazuh. Posteriormente, ingresamos a la herramienta utilizando las credenciales proporcionadas.

El siguiente paso implica obtener la dirección IP de la máquina Ubuntu donde se llevará a cabo la monitorización del malware, que en este caso es 192.168.179.132.

Acto seguido, procedemos a configurar Wazuh, comenzando con la especificación de la dirección IP de la máquina a monitorear (192.168.179.132, en este caso, que corresponde a

la máquina Ubuntu). Luego, añadimos el nombre del *agente*, que será “Windowsvmware”, asignamos el grupo como “default”.

Posteriormente, ejecutamos el comando proporcionado por Wazuh dentro de la máquina Windows, donde se ejecutará el malware:

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.7.2-1.msi -OutFile ${env.tmp}\wazuh-agent; msiexec.exe /i ${env.tmp}\wazuh-agent /q WAZUH_MANAGER='192.168.179.132' WAZUH_AGENT_NAME='Windowsvmware' WAZUH_REGISTRATION_SERVER='192.168.179.132'
```

Una vez ejecutado este comando en la máquina Windows, verificamos su correcto funcionamiento dentro de la interfaz de Wazuh, observando que la máquina Windows se encuentra enlazada, lo que nos permite iniciar la monitorización de manera efectiva.

b.2. Preparamos el malware

Iniciamos el proceso instalando el Smokeloader en la máquina virtual de Windows. Como medida de seguridad para cuidar nuestros entornos de trabajo y pruebas, tomamos una *snapshot* de la máquina para garantizar la seguridad al ejecutar el malware. A continuación, lo descargamos desde el servidor CapeV. Para permitir su ejecución sin problemas, desactivamos temporalmente el antivirus y todos los sistemas de seguridad en la máquina Windows.

b.3. Ejecución del Smokeloader en el entorno de prueba

Procedemos a ejecutar el Smokeloader para observar su comportamiento, que ya conocemos por el análisis previo realizado. Inicialmente, evaluamos la reacción del virus utilizando los módulos estándar de Wazuh, realizando un análisis preliminar.

Wazuh monitorea activamente la máquina donde se ejecuta el malware, proporcionando una visión detallada de su comportamiento.

Observamos que, con la configuración predeterminado, Wazuh no logra detectar ninguna anomalía, es decir, el malware pasa desapercibido.

b.4. Solucionando errores técnicos de funcionamiento: implementación y pruebas de laboratorio

Tras la realización de más pruebas y una investigación sobre el funcionamiento de Wazuh y los SIEM, identificamos la necesidad de establecer una conexión con Kibana y Elasticsearch para una configuración efectiva del SIEM de Wazuh.

Para ello, utilizamos el siguiente comando:

```
curl -so ~/unattended-installation.sh https://packages.wazuh.com/resources/4.2/open-distro/unattended-installation/unattended-installation.sh && bash ~/unattended-installation.sh
```

Con el servidor SIEM de Elasticsearch, Kibana y Wazuh instalado, realizamos una prueba inicial con un troyano descargado de www.virusshare.com, observando cómo Wazuh comienza a proporcionar información valiosa.

Posteriormente, una vez hemos comprobado que el funcionamiento del SIEM es el esperado, ejecutamos de nuevo Smokeloader. Ahora, revisando los logs, comprobamos que se generan registros detallados sobre su comportamiento, incluyendo cambios en los registros para establecer persistencia y crear accesos. Identificamos vulnerabilidades en Python que permiten activar "RecursionError", destacando la necesidad de abordar estos problemas para fortalecer la seguridad.

Se anexa un gráfico en el cual se muestra el grado de vulnerabilidades detectadas por Wazuh sobre el Smokeloader. A continuación, procederemos a probar su detección mediante la incorporación de Indicadores de Compromiso (IOCs) a través de MISP.

b.5. Integración del MISP y de los IOCs como reglas en el ELK

Continuamos con la integración de IOCs desde MISP para evaluar la detección de anomalías en nuestro SIEM. Para lograr esto, primero creamos un archivo utilizando el editor Nano en la ruta `/var/ossec/integrations`. Copiamos y pegamos un script que establecerá la conexión entre MISP y nuestro SIEM. En el script, agregamos el dominio donde hemos creado nuestro evento en MISP: <https://13.48.162.234/events/view/24>

En el servidor MISP, generamos una clave (key) que tiene el siguiente formato: "gECgwGfOjyGKSrHfbFHqv21RwbvH5Ez3Bs4q5ISR". A continuación, incorporamos esta clave a nuestro script. Al modificar éste, integramos las reglas de MISP en el SIEM para detectar eventos generados en el servidor MISP.

Posteriormente, ajustamos la configuración de Wazuh para agregar la integración de MISP. El archivo de configuración se encuentra en la siguiente ruta: `/var/ossec/etc/ossec.conf`

En ese archivo, debemos añadir lo siguiente:

```
<integration>
  <name>custom-misp.py</name>

  <group>sysmon_event1,sysmon_event3,sysmon_event6,sysmon_event7,sysmon_event_15,sysmon_event_22,syscheck</group>
  <alert_format>json</alert_format>
</integration>
```

Acto seguido, reiniciamos Wazuh para aplicar los cambios. Ingresamos al SIEM y añadimos un código con las reglas de MISP en la siguiente sección:

```
<group name="misp,">
  <rule id="100620" level="10">
```

```

<field name="integration">misp</field>

<match>misp</match>

<description>MISP Events</description>

<options>no_full_log</options>

</rule>

<rule id="100621" level="5">

  <if_sid>100620</if_sid>

  <field name="misp.error">\.+</field>

  <description>MISP - Error connecting to API</description>

  <options>no_full_log</options>

  <group>misp_error,</group>

</rule>

<rule id="100622" level="12">

  <field name="misp.category">\.+</field>

  <description>MISP - IoC found in Threat Intel - Category:
$(misp.category), Attribute: $(misp.value)</description>

  <options>no_full_log</options>

  <group>misp_alert,</group>

</rule>

</group>

```

Estos ajustes permiten que Wazuh detecte eventos y alertas generados en MISP, fortaleciendo así la capacidad del SIEM para identificar amenazas basadas en la inteligencia de amenazas recopilada.

b.6. Nuevos errores en la detección. Buscando nuevas soluciones: creamos nuestro propio MISP

Se observa que, tras lanzar el malware, no se detectan los IOCs como se espera, por lo que vamos a realizar otra prueba, instalando el MISP en la propia máquina de monitoreo, en el Ubuntu.

Actualizamos la máquina Ubuntu con el siguiente comando:

```
sudo apt-get update && sudo apt-get upgrade -y
```

Tras ello, procedemos a instalar la herramienta MySQL con el siguiente comando:

```
sudo apt-get install mysql-client -y
```

Descargamos el repositorio de MISP que contiene todas las herramientas necesarias para la integración:

```
wget https://raw.githubusercontent.com/MISP/MISP/2.4/INSTALL/INSTALL.sh
```

Y le otorgamos permisos de ejecución al script descargado:

```
chmod +x INSTALL.sh
```

Finalmente, ejecutamos el script como *root* para instalar MISP:

```
./INSTALL.sh -A
```

Este paso instalará y configurará MISP en el sistema operativo Ubuntu, en la máquina que estamos empleando para realizar la monitorización. Una vez completada la instalación de MISP, se han generado las credenciales necesarias para acceder al sistema:

- **Admin (root) DB Password:**
'5d094558ae0997266b4fe92cf0d89982a4e5e84b26d4c4718523d4bb4b974872'.
- **User (misp) DB Password:**
'7312c508ff8dc332cf473a4ccd00119e57bdc19f055dabece9bc92a745d00f7d'
- **User: 'misp'**
Password:
'72a10ea52d40b75b700aec5326555c462e662cb31dff9a8814016acc5acfab36'

Además, se procede a habilitar los puertos 80 y 443 para permitir el acceso a MISP. Con este acceso confirmado, el siguiente paso es crear un evento MISP y agregar los IOCs que ya tenemos identificados del análisis malware y ya tenemos recogidos en el evento del otro MISP. Se han incluido todos los atributos relevantes obtenidos del análisis de amenazas.

Otro punto necesario dentro de esta configuración, es la instalación del ejecutable sysmon. Éste contiene una serie de normas y configuración que implica que al ejecutarlo podamos visualizar los eventos y logs en forma de registros que se van sucediendo en el Windows monitorizado. Esto nos permitirá visualizar que está pasando en la máquina monitorizada a tiempo real y poder interpretar el comportamiento del malware dentro de la máquina.

Hipotéticamente, este proceso garantiza que la inteligencia de amenazas recopiladas en MISP se integre de manera efectiva en el SIEM, enriqueciendo la capacidad del SIEM para detectar y responder a amenazas basadas en la información de amenazas actualizada.

No obstante, por operatividad, procedemos a instalar el MISP en una máquina ajena al Ubuntu para no interferir en las IPs empleadas en todo el entorno de monitorización.

RESULTADOS Y ANÁLISIS

a. Evaluación de la detección mejorada

La mejora de la detección en un Sistema de Información y Eventos de Seguridad (SIEM) es crucial para fortalecer la postura de ciberseguridad de una organización. Integrar Indicadores de Compromiso (IOCs) de un malware específico en un SIEM es una estrategia efectiva para aumentar la capacidad de detección y respuesta ante amenazas.

Los Indicadores de Compromiso (IOCs) son datos forenses que evidencian la presencia de actividad maliciosa en un sistema. Pueden incluir direcciones IP, nombres de dominio, hashes de archivos, patrones de comportamiento, entre otros.

La inclusión de IOCs en el SIEM ha permitido detectar patrones específicos asociados al malware concreto, en este caso, Smokeloader. Esto ha facilitado la identificación de amenazas antes de que se materialicen en ataques completos.

Los IOCs proporcionan información sobre amenazas previamente identificadas, permitiendo al SIEM adaptarse y ajustar las reglas de detección para enfrentar nuevas variantes o evoluciones de malware.

Al basar las reglas de detección en IOCs específicos, se ha reducido la probabilidad de falsos positivos, ya que las alertas estarán vinculadas directamente a características únicas del malware en cuestión. Hemos tenido que asegurar que los IOCs estuviesen en un formato estandarizado y compatible con el SIEM para facilitar su integración y análisis.

La integración de IOCs de malware específico en el SIEM es una estrategia clave para fortalecer la detección de amenazas y mejorar la capacidad de respuesta de una organización ante incidentes de ciberseguridad. Este enfoque proactivo proporciona una capa adicional de defensa al identificar y mitigar posibles riesgos antes de que afecten la integridad y confidencialidad de los sistemas de información.

b. Impacto en la respuesta ante amenazas

Con el entorno combinado de herramientas para nuestro ELK, se ha conformado un entorno robusto para la detección de malware y amenazas en entornos de seguridad.

b.1. Almacenamiento e indexación eficientes

Elasticsearch actúa como un motor de búsqueda y almacenamiento altamente escalable. Permite indexar y buscar grandes volúmenes de datos generados por Wazuh de manera

eficiente. Esto facilita el acceso y análisis rápido de registros de seguridad, mejorando la capacidad de detección.

b.2. Visualización y análisis avanzados

Kibana proporciona una interfaz de usuario intuitiva para visualizar y analizar datos almacenados en Elasticsearch. También permite crear paneles personalizados, informes y visualizaciones para comprender rápidamente patrones y anomalías en la actividad del sistema.

b.3. Búsqueda y correlación rápida

La integración con Elasticsearch permite búsquedas rápidas y complejas en los registros generados por Wazuh. Permite correlacionar eventos de seguridad, identificar relaciones y encontrar patrones que podrían indicar comportamientos maliciosos.

b.4. Detección en tiempo real

Wazuh monitorea eventos en tiempo real y envía alertas al servidor Elasticsearch. La combinación de Elasticsearch y Kibana facilita la visualización y análisis inmediato de estas alertas, permitiendo una respuesta más rápida ante posibles amenazas.

b.5. Enriquecimiento de datos

Elasticsearch y Kibana permiten el enriquecimiento de datos provenientes de Wazuh. Se puede agregar información contextual, como datos de amenazas externas o inteligencia de amenazas, para mejorar la comprensión de la naturaleza de las alertas generadas por Wazuh.

b.6. Automatización de respuestas

La integración con Elasticsearch y Kibana permite la automatización de respuestas a incidentes. Se pueden configurar reglas y scripts para ejecutarse automáticamente en respuesta a eventos específicos detectados por Wazuh.

b.7 Capacidad de escalabilidad

La escalabilidad inherente de Elasticsearch facilita la gestión de grandes volúmenes de datos generados por Wazuh, lo que es esencial para entornos empresariales con numerosos dispositivos y eventos de seguridad.

b.8. Detección específica

La incorporación de IOCs específicos de un malware permite una detección más precisa de amenazas conocidas. Los patrones de comportamiento y las firmas asociadas con ese malware en particular se utilizan para identificar de manera más eficiente actividades maliciosas relacionadas.

b.9. Mejora en la precisión del SIEM

La inclusión de IOCs específicos en el SIEM mejora la precisión de las alertas. Al correlacionar los eventos detectados por Wazuh con los IOCs del malware, se reducen los falsos positivos y se identifican con mayor certeza las actividades sospechosas.

b.10. Respuesta rápida

Al contar con IOCs específicos, el SIEM puede automatizar respuestas inmediatas a incidentes relacionados con ese malware. Esto agiliza la capacidad de respuesta y minimiza el tiempo de exposición frente a amenazas conocidas.

b.11. Seguimiento de campañas de amenazas

La integración de IOCs permite rastrear y correlacionar eventos a lo largo del tiempo, lo que facilita el seguimiento de campañas de amenazas específicas. Se puede obtener una visión más completa de la extensión y persistencia de las actividades maliciosas. Además, mantener una biblioteca actualizada de IOCs específicos en el SIEM garantiza que la inteligencia de amenazas esté al día. Esto permite a las organizaciones adaptarse rápidamente a las nuevas variantes o tácticas utilizadas por el malware.

c. Lecciones aprendidas

c.1. Desafíos en la Integración del MISP

La integración del MISP con Wazuh presentó ciertos desafíos que afectaron la capacidad de compartir información de amenazas de manera efectiva. Estos desafíos incluyeron:

1. **Compatibilidad de versiones:** Diferencias en las versiones entre Wazuh y MISP causaron incompatibilidades en la integración, lo que dificultó la sincronización y el intercambio de datos entre las dos plataformas.
2. **Configuración incorrecta de API:** Errores en la configuración de la API de MISP dentro de Wazuh llevaron a fallas en la autenticación y la transferencia de datos, lo que limitó la eficacia de la colaboración entre las dos herramientas.
3. **Capacitación insuficiente del personal:** La falta de capacitación adecuada sobre la integración de MISP y Wazuh resultó en errores durante la configuración inicial y una comprensión limitada de las mejores prácticas para compartir inteligencia de amenazas.

c.2. Dificultades en la Detección de Logs

La capacidad de Wazuh para detectar y responder a eventos de seguridad basados en logs se vio comprometida debido a varios desafíos, que incluyen:

1. **Reglas de correlación inadecuadas:** Las reglas de correlación predefinidas no abordaron adecuadamente los casos de uso específicos de la organización, lo que resultó en falsos positivos o negativos y una detección ineficaz de amenazas.

2. **Configuración deficiente de agentes:** La configuración incorrecta de los agentes de Wazuh en los sistemas finales dificultó la recopilación y el análisis de logs relevantes, lo que limitó la capacidad de detección de amenazas en toda la infraestructura.
3. **Falta de personal capacitado:** La falta de personal con experiencia en la administración y configuración de Wazuh resultó en una respuesta lenta a las alertas de seguridad y una capacidad limitada para optimizar la detección de logs.

c.3. Desafíos en la Detección de Malware

Además de los desafíos anteriores, se encontraron dificultades específicas en la detección efectiva de malware, que incluyeron:

1. **Actualización de firmas de malware obsoletas:** La falta de actualización regular de las firmas de malware dentro de Wazuh resultó en la incapacidad de detectar nuevas variantes de malware y amenazas emergentes.
2. **Escaneos de malware poco frecuentes:** La programación insuficiente de escaneos de malware periódicos permitió que las amenazas permanecieran indetectadas durante largos períodos de tiempo, aumentando el riesgo de compromiso de la seguridad.
3. **Detección de malware en archivos cifrados:** La incapacidad de Wazuh para escanear y detectar malware dentro de archivos cifrados limitó la eficacia de la protección contra amenazas en entornos donde el cifrado de archivos es común.

CONCLUSIONES Y RECOMENDACIONES

a. Logros y contribuciones

El enriquecimiento de un Sistema de Gestión de Eventos e Información de Seguridad (SIEM) con Indicadores de Compromiso (IOCs) de distintos malwares potencialmente peligrosos, no solo el Smokeloader, podría aportar numerosos beneficios al sector de la ciberseguridad.

El enriquecimiento del SIEM con IOCs permite una **detección más proactiva** de amenazas. Al incorporar firmas, patrones de comportamiento y características específicas de malware, el SIEM puede identificar actividades sospechosas antes de que se conviertan en incidentes de seguridad.

La inclusión de IOCs específicos en el SIEM contribuye a **reducir los falsos positivos**. La correlación de eventos con indicadores conocidos permite una evaluación más precisa de la autenticidad de las alertas, minimizando la carga de trabajo innecesaria para los equipos de seguridad.

Cuando se detecta una amenaza conocida mediante IOCs, el SIEM puede activar respuestas automatizadas de forma rápida y eficiente. Esto **acelera el tiempo de respuesta ante incidentes y minimiza el impacto** de ataques bien documentados.

El enriquecimiento constante del SIEM con nuevos IOCs **garantiza que la inteligencia de amenazas esté siempre actualizada**. Estar al tanto de las tácticas, técnicas y procedimientos (TTP) utilizados por malwares potencialmente peligrosos permite a las organizaciones anticiparse y adaptarse a nuevas variantes y tácticas.

Los IOCs proporcionan **valiosa información para el análisis forense**. Al enriquecer el SIEM con esta información, los profesionales de ciberseguridad pueden realizar investigaciones más profundas y entender mejor la naturaleza de los ataques, identificando la huella digital específica de los malwares.

Al integrar IOCs de malwares en el SIEM, las organizaciones **contribuyen a la comunidad de ciberseguridad** al compartir información sobre amenazas. Este intercambio colaborativo fortalece la defensa colectiva contra amenazas comunes y beneficia a toda la industria.

La capacidad de enriquecer el SIEM con IOCs permite a las organizaciones **adaptarse rápidamente a amenazas emergentes**. Asimismo, la retroalimentación de IOCs en el SIEM permite una **mejora continua de las políticas de seguridad**. Las organizaciones pueden ajustar sus estrategias de seguridad en función de las amenazas detectadas, fortaleciendo así su postura de seguridad general.

b. Limitaciones y desafíos

A lo largo de este proyecto, nos hemos encontrado con diversas limitaciones y desafíos que debemos encarar de cara a futuro.

b.1. Complejidad en la configuración

La configuración de un SIEM con integración de IOCs es compleja y requiere de experiencia técnica especializada. La correcta sincronización de componentes como Elasticsearch, Logstash, Kibana y Wazuh, junto con la integración de IOCs, es un reto muy desafiante de cara a implementarlo en una organización y requiere de personal técnicamente cualificado.

b.2. Recursos financieros

La implementación y mantenimiento de un SIEM robusto puede ser costosa. Los costos asociados con licencias de software, hardware, almacenamiento y personal capacitado pueden representar una carga financiera significativa para algunas organizaciones. En el presente proyecto se han utilizado herramientas *open source* pero de cara a implementarlo en una empresa, lo más aconsejable es utilizar herramientas con licencia de cara a un soporte técnico en la implementación o futuros problemas.

b.3. Capacidad de almacenamiento

El almacenamiento y manejo de grandes volúmenes de datos generados por un SIEM, especialmente cuando se incorporan IOCs, pueden requerir una capacidad de almacenamiento considerable. Esto puede ser un reto difícil, especialmente para organizaciones más modestas.

b.4. Actualización continua de IOCs

Mantener actualizada la base de datos de IOCs es esencial para una detección efectiva. Sin embargo, puede resultar complejo o costoso para una organización mantener un equipo específico encargado de rastrear y actualizar continuamente los IOCs a medida que evolucionan las amenazas. Sería imprescindible una buena coordinación entre los equipos de análisis malware y BlueTeam; o bien, instaurar un equipo concreto dentro del departamento de BlueTeam dedicado a ello.

b.5. Personal especializado

La implementación y buena gestión de un SIEM con integración de IOCs requiere de personal con habilidades especializadas en Ciberseguridad. La falta de este tipo de personal puede ser un obstáculo para la implementación y mantenimiento eficientes.

b.6. Integración con una infraestructura ya existente

La integración correcta del SIEM en una infraestructura ya existente en una organización puede ser un desafío por motivos de interoperabilidad con otros sistemas de seguridad y la gestión de la complejidad del proceso.

b.7. Falsos positivos y negativos

La configuración incorrecta o la falta de ajuste de las reglas y políticas en el SIEM pueden llevar a la generación de falsos positivos o, por el contrario, a la omisión de amenazas reales (falsos negativos).

b.8. Evolución de las amenazas

Las amenazas evolucionan constantemente, y los ciberdelincuentes buscan formas de eludir las defensas. Mantenerse al día con las tácticas de ataque cambiantes y adaptar continuamente las defensas es un desafío constante y costoso para las organizaciones.

c. Sugerencias para futuras mejoras

En el contexto actual de Ciberseguridad, la detección de amenazas y la respuesta a incidentes son elementos críticos para proteger los activos digitales y garantizar la continuidad de las operaciones empresariales. En este informe, se presentarán recomendaciones para mejorar la capacidad de detección de anomalías utilizando la plataforma de seguridad Wazuh, junto con sugerencias adicionales para optimizar la gestión de recursos y promover la capacitación del personal.

c.1. Integración de Fuentes de Inteligencia de Amenazas

Es fundamental integrar Wazuh con fuentes de inteligencia de amenazas externas para fortalecer su capacidad de detección y respuesta ante incidentes. Se recomienda utilizar *feeds* de indicadores de compromiso (IOC) y listas de reputación para enriquecer la detección de anomalías. Los IOC pueden identificar patrones de comportamiento sospechoso, mientras que las listas de reputación ayudan a identificar fuentes conocidas de amenazas. Esta integración amplía la visibilidad de la plataforma y mejora la identificación de comportamientos anómalos en tiempo real.

c.2. Capacitación y concienciación:

La capacitación del personal es un componente esencial para mejorar la detección y respuesta ante incidentes utilizando Wazuh. Se recomienda invertir en programas de capacitación que aseguren que los miembros del equipo estén familiarizados con las capacidades de la plataforma y sepan cómo interpretar las alertas correctamente. Las sesiones de formación práctica y el material educativo son herramientas efectivas para mejorar la conciencia de seguridad y promover una cultura de Ciberseguridad dentro de la organización.

c.3. Gestión eficiente de recursos:

La gestión eficiente de recursos es clave para minimizar los costos operativos asociados con la implementación de Wazuh. Se sugiere optimizar el uso de recursos de hardware y almacenamiento evaluando regularmente la infraestructura subyacente y ajustando las configuraciones según sea necesario. Esto garantiza un rendimiento óptimo de la plataforma a un costo razonable, maximizando así el retorno de inversión en seguridad cibernética.

c.4. Recomendaciones adicionales:

Además de las mejoras mencionadas, se recomienda considerar las siguientes acciones para fortalecer la postura de seguridad de la organización:

- Implementar evaluaciones regulares de riesgos y priorizar las medidas de seguridad según el impacto económico potencial de una violación de seguridad.
- Establecer políticas de cumplimiento para garantizar el cumplimiento de los requisitos regulatorios y normativos pertinentes.
- Desarrollar planes de continuidad del negocio que incorporen la detección y respuesta de amenazas como parte integral de la estrategia de recuperación de desastres.

c.5. Conclusiones:

En resumen, la integración con fuentes de inteligencia de amenazas, la capacitación del personal y la gestión eficiente de recursos son componentes clave para mejorar la detección de anomalías con Wazuh. Al implementar estas recomendaciones, las organizaciones podrían fortalecer su postura de seguridad y mitigar los riesgos asociados con las amenazas en evolución constante en el panorama digital actual.

ANEXOS

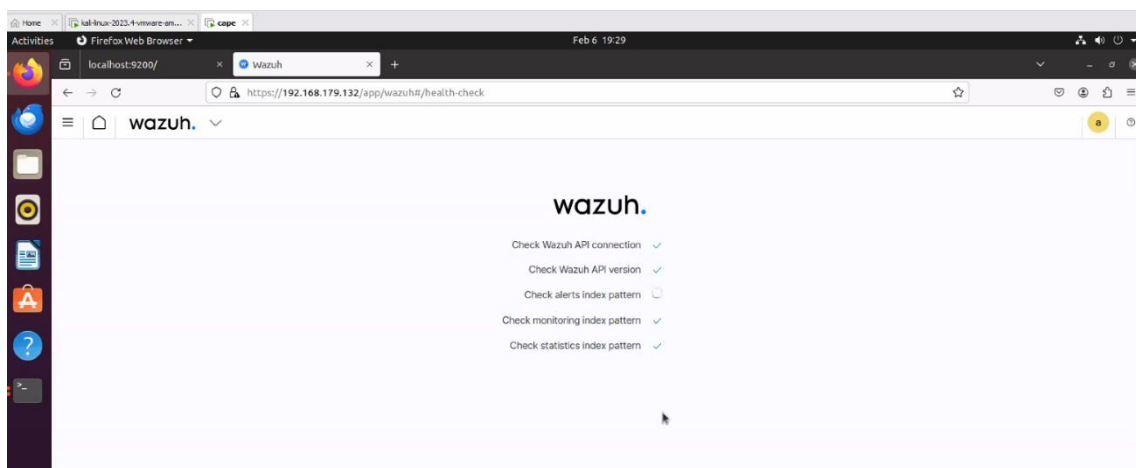
a. Configuraciones detalladas

```
cape@ubuntu:~$ setxkbmap es
cape@ubuntu:~$ curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh -a -l
[sudo] password for cape:
06/02/2024 18:55:41 INFO: Starting Wazuh Installation assistant. Wazuh version: 4.7.2
06/02/2024 18:55:41 INFO: Verbose logging redirected to /var/log/wazuh-install.log
06/02/2024 18:59:25 INFO: --- Dependencies ---
06/02/2024 18:59:25 INFO: Installing gawk.
06/02/2024 19:00:52 WARNING: Hardware and system checks ignored.
06/02/2024 19:00:52 INFO: Wazuh web interface port will be 443.
06/02/2024 19:01:19 INFO: --- Dependencies ---
06/02/2024 19:01:19 INFO: Installing apt-transport-https.
06/02/2024 19:02:29 INFO: Wazuh repository added.
06/02/2024 19:02:29 INFO: --- Configuration files ---
06/02/2024 19:02:29 INFO: Generating configuration files.
06/02/2024 19:02:31 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
06/02/2024 19:02:31 INFO: --- Wazuh indexer ---
06/02/2024 19:02:31 INFO: Starting Wazuh Indexer installation.
```

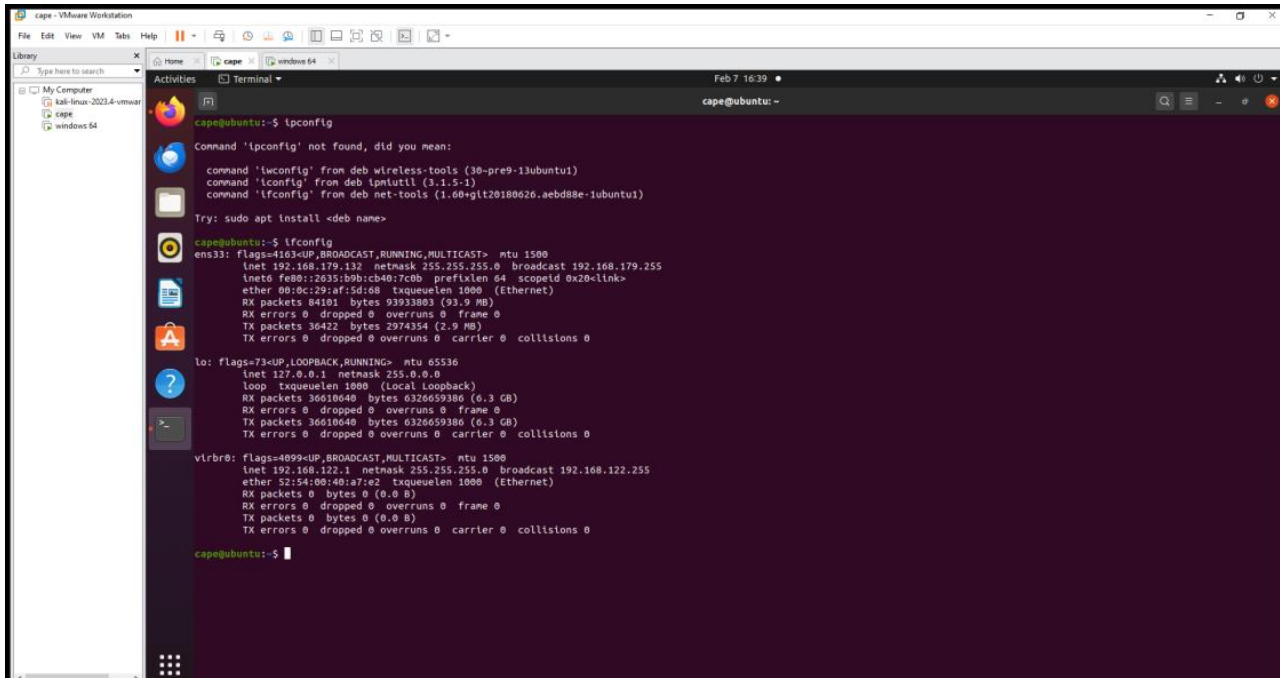
a.1. Instalación Wazuh

```
cape@ubuntu:~$ curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh && sudo bash ./wazuh-install.sh -a -l
06/02/2024 19:04:53 INFO: Starting Wazuh Installation assistant. Wazuh version: 4.7.2
06/02/2024 19:04:53 INFO: Verbose logging redirected to /var/log/wazuh-install.log
06/02/2024 19:05:06 INFO: --- Dependencies ---
06/02/2024 19:05:06 INFO: Installing gawk.
06/02/2024 19:05:22 WARNING: Hardware and system checks ignored.
06/02/2024 19:05:22 INFO: Wazuh web interface port will be 443.
06/02/2024 19:05:29 INFO: Installing apt-transport-https.
06/02/2024 19:05:48 INFO: Wazuh repository added.
06/02/2024 19:05:48 INFO: --- Configuration files ---
06/02/2024 19:05:48 INFO: Generating configuration files.
06/02/2024 19:05:49 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
06/02/2024 19:05:49 INFO: --- Wazuh indexer ---
06/02/2024 19:05:49 INFO: Starting Wazuh indexer installation.
06/02/2024 19:06:58 INFO: Wazuh indexer installation finished.
06/02/2024 19:06:59 INFO: Wazuh indexer post-install configuration finished.
06/02/2024 19:06:59 INFO: Starting service wazuh-indexer.
06/02/2024 19:07:22 INFO: Wazuh-indexer service started.
06/02/2024 19:07:22 INFO: Initializing Wazuh indexer cluster security settings.
06/02/2024 19:07:33 INFO: Wazuh indexer cluster initialized.
06/02/2024 19:07:33 INFO: --- Wazuh server ---
06/02/2024 19:07:33 INFO: Starting the Wazuh manager installation.
06/02/2024 19:08:38 INFO: Wazuh manager installation finished.
06/02/2024 19:08:38 INFO: Starting service wazuh-manager.
06/02/2024 19:08:56 INFO: wazuh-manager service started.
06/02/2024 19:08:56 INFO: Starting filebeat installation.
06/02/2024 19:09:22 INFO: Filebeat installation finished.
06/02/2024 19:09:23 INFO: Filebeat post-install configuration finished.
06/02/2024 19:09:23 INFO: Starting service filebeat.
06/02/2024 19:09:24 INFO: Filebeat service started.
06/02/2024 19:09:24 INFO: --- Wazuh dashboard ---
06/02/2024 19:09:24 INFO: Starting Wazuh dashboard installation.
06/02/2024 19:10:22 INFO: Wazuh dashboard installation finished.
06/02/2024 19:10:22 INFO: Wazuh dashboard post-install configuration finished.
06/02/2024 19:10:22 INFO: Starting service wazuh-dashboard.
06/02/2024 19:10:23 INFO: wazuh-dashboard service started.
06/02/2024 19:10:48 INFO: Initializing Wazuh dashboard web application.
06/02/2024 19:10:49 INFO: Wazuh dashboard web application initialized.
06/02/2024 19:10:49 INFO: --- Summary ---
06/02/2024 19:10:49 INFO: You can access the web interface https://wazuh-dashboard-ip:443
User: admin
Password: WpYFf.dKrc1CVZy2t1:2Curn+7qZndXk
06/02/2024 19:10:49 INFO: --- Dependencies ---
06/02/2024 19:10:49 INFO: Removing gawk.
06/02/2024 19:11:07 INFO: Installation finished.
```

a.2. Usuario y contraseña Wazuh



a.3. Iniciación de Wazuh



```
cape@ubuntu:~$ ipconfig
Command 'ipconfig' not found, did you mean:
  command 'lswconfig' from deb wireless-tools (30-pre9-1ubuntu1)
  command 'lconfig' from deb ipmitool (3.1.5-1)
  command 'lfcconfig' from deb net-tools (1.60+git20180526.aebd88e-1ubuntu1)
Try: sudo apt install <deb name>

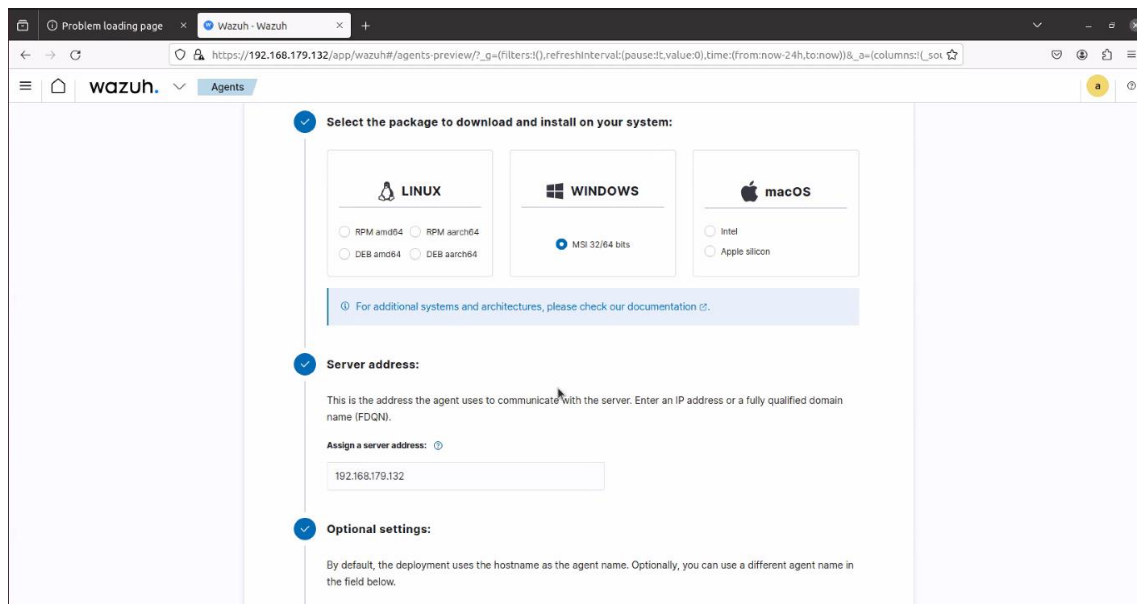
cape@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.179.132 netmask 255.255.255.0 broadcast 192.168.179.255
    inet6 fe80::2035:b9b:cb40:7c0b prefixlen 64 scopeid 0x20<link>
    ether 08:0c:29:af:5d:68 txqueuelen 1000 (Ethernet)
    RX packets 84101 bytes 93933803 (93.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 36422 bytes 2974354 (2.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 36610640 bytes 6326659386 (6.3 GB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 36610640 bytes 6326659386 (6.3 GB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

vlrbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:48:a7:e2 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

cape@ubuntu:~$
```

a.4. IP máquina Ubuntu



Wazuh - Agents

Select the package to download and install on your system:

- LINUX**
 - ☐ RPM amd64
 - ☐ RPM aarch64
 - ☐ DEB amd64
 - ☐ DEB aarch64
- WINDOWS**
 - ☒ MSI 32/64 bits
- macOS**
 - ☐ Intel
 - ☐ Apple silicon

For additional systems and architectures, please check our documentation.

Server address:

This is the address the agent uses to communicate with the server. Enter an IP address or a fully qualified domain name (FQDN).

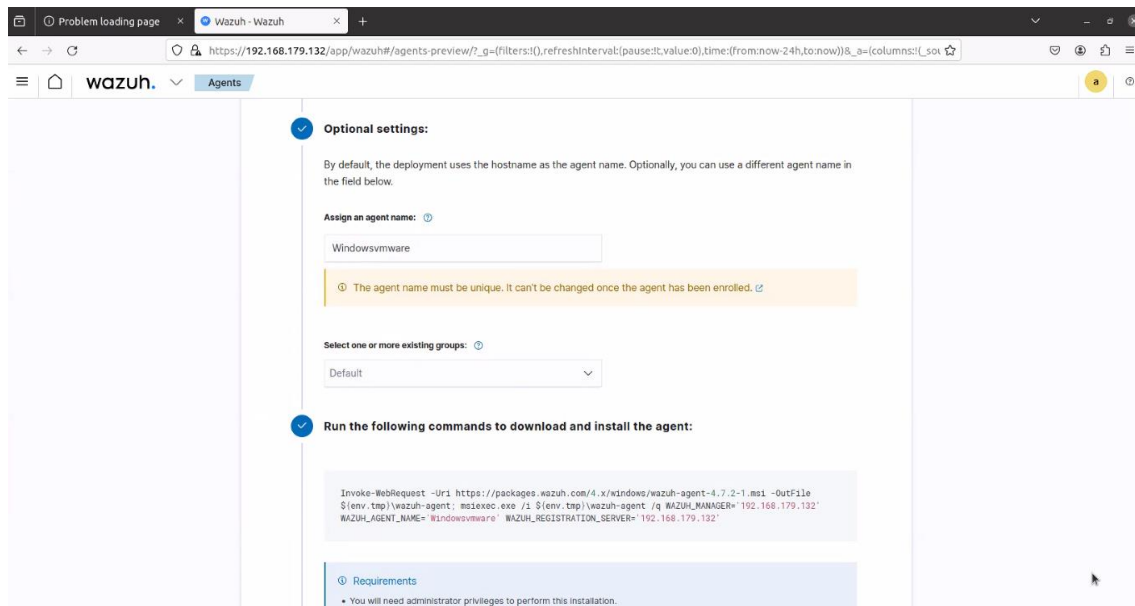
Assign a server address:

192.168.179.132

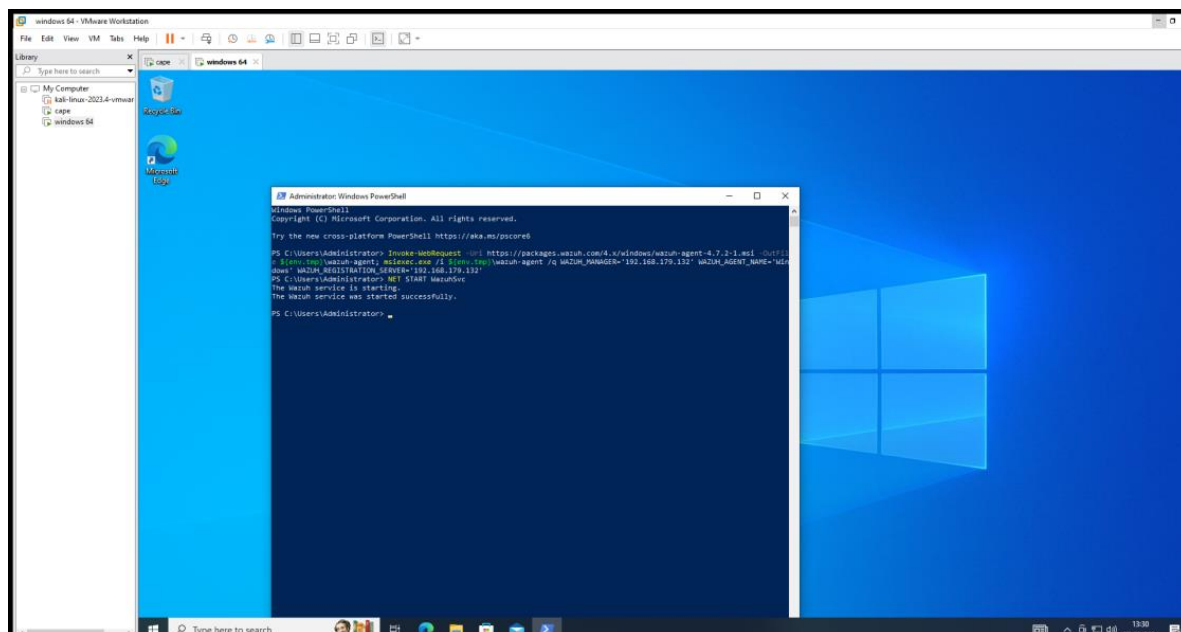
Optional settings:

By default, the deployment uses the hostname as the agent name. Optionally, you can use a different agent name in the field below.

a.5.1 Configuración wazuh



a.5.2 Configuración wazuh

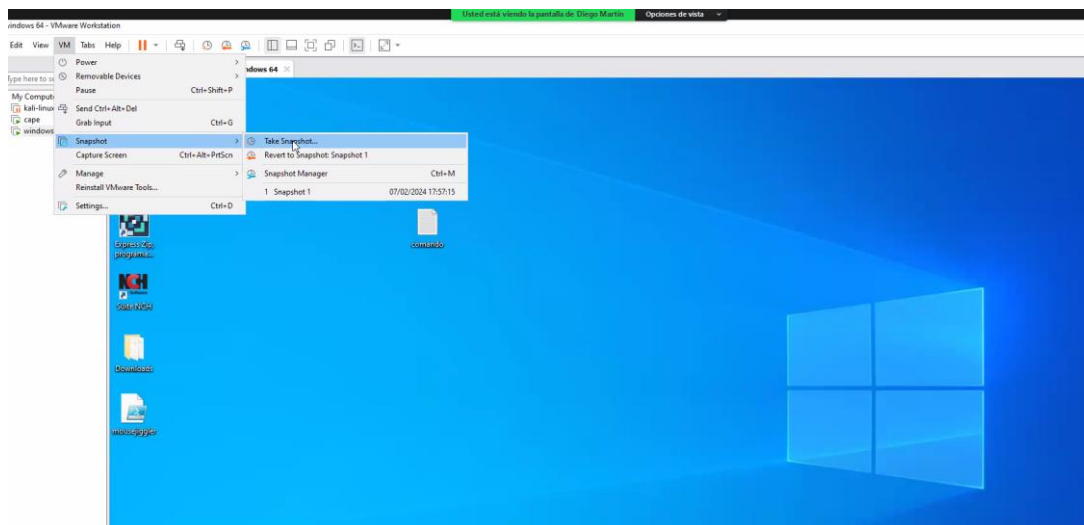


a.6. Monitorización wazuh con máquina windows

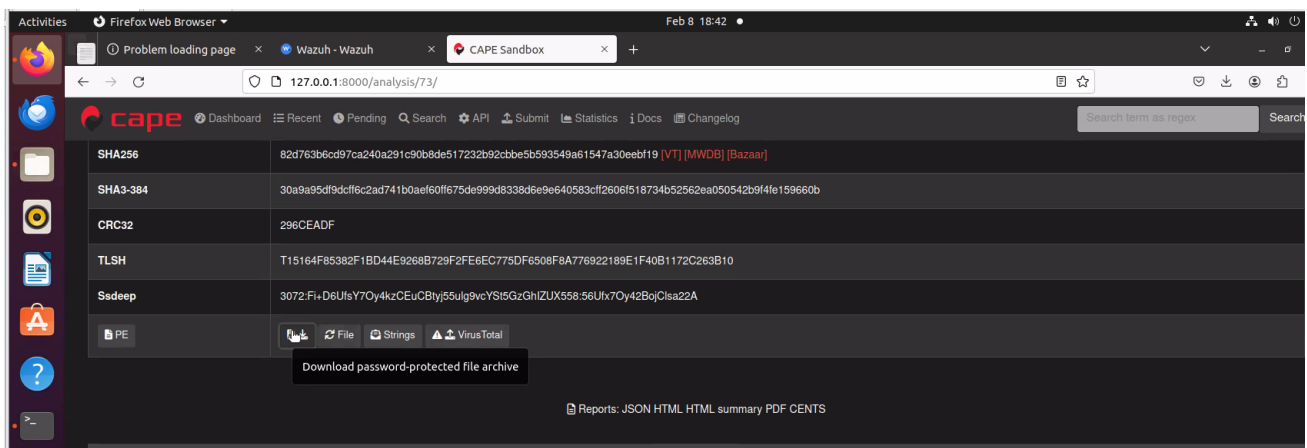

```
root@ubuntu: /home/cape
sudo] password for cape:
root@ubuntu:/home/cape# curl -so ~/unattended-installation.sh https://packages.w
zuh.com/resources/4.2/open-distro/unattended-installation/unattended-installati
n.sh && bash ~/unattended-installation.sh
2/12/2024 19:12:09 INFO: Starting the installation...
2/12/2024 19:12:09 INFO: Installing all necessary utilities for the installatio
....
2/12/2024 19:12:30 INFO: Done
2/12/2024 19:12:30 INFO: Adding the Wazuh repository...
2/12/2024 19:12:33 INFO: Done
2/12/2024 19:12:33 INFO: Installing the Wazuh manager...
2/12/2024 19:13:17 INFO: Done
2/12/2024 19:13:38 INFO: Wazuh-manager started
2/12/2024 19:13:38 INFO: Installing Open Distro for Elasticsearch...
2/12/2024 19:14:14 INFO: Done
2/12/2024 19:14:14 INFO: Configuring Elasticsearch...
2/12/2024 19:14:17 INFO: Configuration file found. Creating certificates...
2/12/2024 19:14:17 INFO: Creating the Elasticsearch certificates...
2/12/2024 19:14:17 INFO: Creating Wazuh server certificates...
2/12/2024 19:14:17 INFO: Creating Kibana certificate...
2/12/2024 19:14:17 INFO: Certificates creation finished. They can be found in ~
certs.
2/12/2024 19:14:17 INFO: Certificates created
```

a.7 Instalación kibana con elastic y wazuh

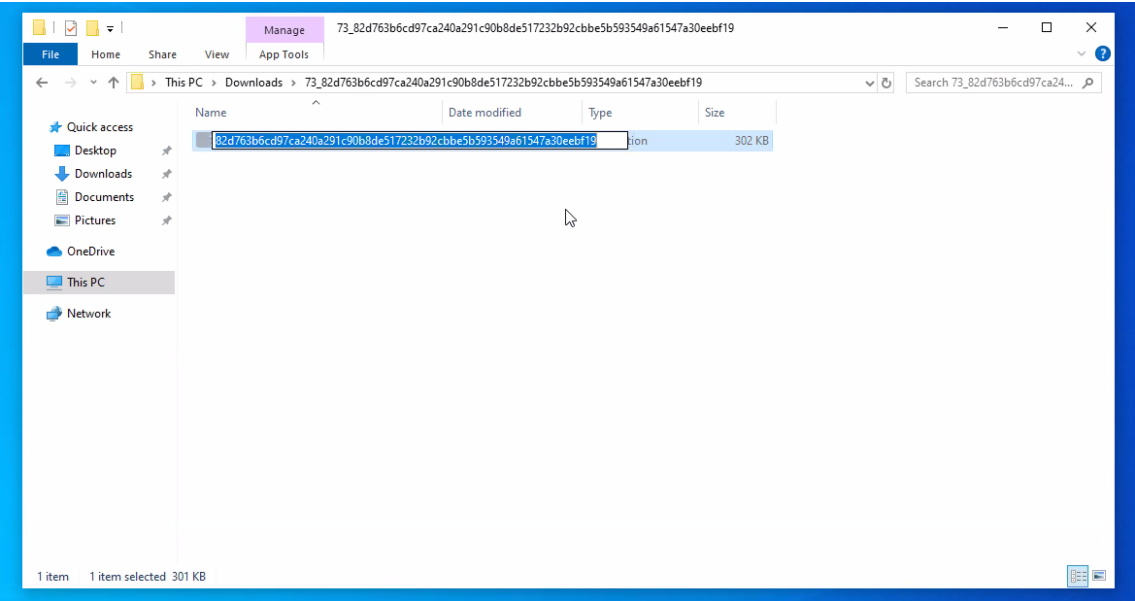
b. Capturas de pantalla relevantes



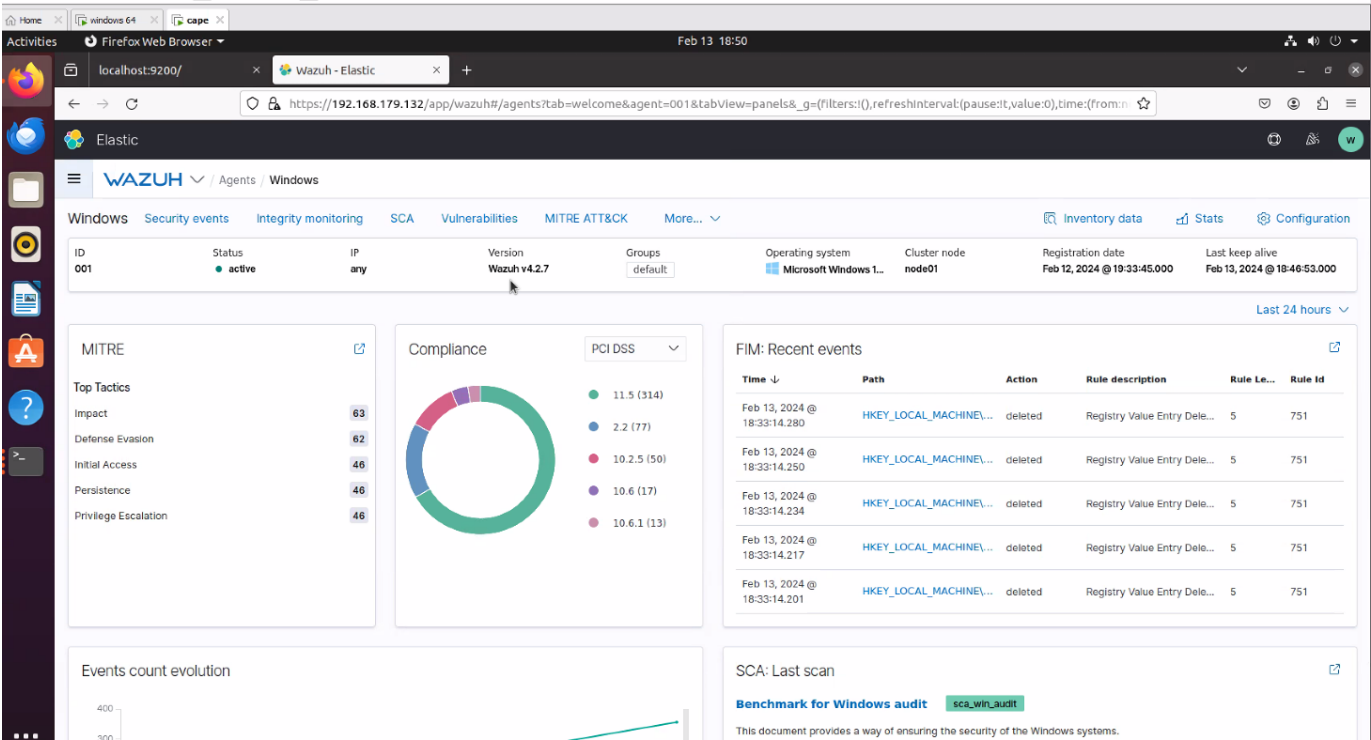
b.1 Snapshot



b.2 Descarga malware



b.3 Ejecución malware



b.4. Detección de wazuh

WAZUH / Modules / Windows / Security events				
subjectLogonId	>	Feb 14, 2024 @ 12:05:47.811	Registry Key Integrity Checksum Changed	5 594
data.win.eventdata.subjectUserName	>	Feb 14, 2024 @ 12:05:46.930	Registry Value Integrity Checksum Changed	5 750
data.win.eventdata.subjectUserSid	>	Feb 14, 2024 @ 12:05:46.930	Registry Value Integrity Checksum Changed	5 750
data.win.eventdata.subStatus	>	Feb 14, 2024 @ 12:05:46.930	Registry Key Integrity Checksum Changed	5 594
data.win.eventdata.targetDomainName	>	Feb 14, 2024 @ 12:05:46.986	Registry Key Integrity Checksum Changed	5 594
data.win.eventdata.targetLinkedLogonId	>	Feb 14, 2024 @ 12:05:34.269	Registry Value Integrity Checksum Changed	5 750
data.win.eventdata.targetLogonId	>	Feb 14, 2024 @ 12:05:34.269	Registry Key Integrity Checksum Changed	5 594
data.win.eventdata.targetUserName	>	Feb 14, 2024 @ 12:05:34.183	Registry Key Integrity Checksum Changed	5 594
data.win.eventdata.targetUserSid	>	Feb 14, 2024 @ 12:05:34.180	Registry Key Integrity Checksum Changed	5 594
data.win.eventdata.virtualAccount	>	Feb 14, 2024 @ 12:05:33.323	Registry Key Integrity Checksum Changed	5 594
data.win.eventdata.workstationName	>	Feb 14, 2024 @ 12:05:17.320	Registry Key Integrity Checksum Changed	5 594
data.win.system.channel	>	Feb 14, 2024 @ 12:05:13.520	Software Protection service scheduled successfully	3 60642
data.win.system.computer	>	Feb 14, 2024 @ 12:05:08.774	Windows Logon Success	3 60106
data.win.system.eventID	>	Feb 14, 2024 @ 12:05:08.730	Windows Logon Success	3 60106
data.win.system.eventRecordID	>	Feb 14, 2024 @ 12:05:08.664	License Activation (slui.exe) failed	5 60646
data.win.system.eventSourceName	>	Feb 14, 2024 @ 12:05:08.634	The database engine attached a database	3 60798
data.win.system.keywords	>	Feb 14, 2024 @ 12:05:08.601	The database engine has completed recovery steps	3 60809
data.win.system.level	>	Feb 14, 2024 @ 12:05:08.586	The database engine is replaying log file C:\Winnt\system32\wins\150.log	3 60808

b.5. Log generados smokeloader

>	Feb 14, 2024 @ 12:34:26.192	CVE-2023-27043 affects Python 3.7.2 (32-bit)	7
>	Feb 14, 2024 @ 12:34:26.181	CVE-2023-36632 affects Python 3.7.2 (32-bit)	10
>	Feb 14, 2024 @ 12:34:26.170	CVE-2023-40217 affects Python 3.7.2 (32-bit)	7

b.6. Vulnerabilidades destacadas

```

root@ubuntu: /var/ossec/integrations
GNU nano 4.8 custom-misp.py Modified
#!/var/ossec/bin/python3
# MISP API Integration
import sys
import os
from socket import socket, AF_UNIX, SOCK_DGRAM
from datetime import date, datetime, timedelta
import time
import requests
from requests.exceptions import ConnectionError
import json
import ipaddress
import hashlib
import re

pwd = os.path.dirname(os.path.dirname(os.path.realpath(__file__)))
socket_addr = '{0}/queue/sockets/queue'.format(pwd)

def send_event(msg, agent = None):
    if not agent or agent["id"] == "000":
        string = 'i:misp:{0}'.format(json.dumps(msg))
    else:
        string = 'i:{0} [{0}] [{0}] {0}'.format(agent["id"], agent["name"], agent["ip"] if "ip" in agent else "any", json.dumps(msg))
    sock = socket(AF_UNIX, SOCK_DGRAM)
    sock.connect(socket_addr)
    sock.send(string.encode())
    sock.close()

false = False
# Read configuration parameters
alert_file = open(sys.argv[1])
# Read the alert file
alert = json.loads(alert_file.read())
alert_file.close()
# Now alert output if MISP Alert or Error calling the API
alert_output = {}
# MISP Server Base URL
misp_base_url = "https://**your misp instance**/attributes/restSearch/"
# MISP Server API Auth Key
misp_api_auth_key = "**Your API Key"
# API - HTTP Headers
misp_apicall_headers = {"Content-Type": "application/json", "Authorization": "f{misp_api_auth_key}", "Accept": "application/json"}
# Construct event for Windows/Sysmon for Linux and Sysmon Event ID
event_source = alert["rule"]["groups"][0]
event_type = alert["rule"]["groups"][2]
# Regen Pattern used based on SHA256 length (44 characters)

```

b.7 Integración script misp

Add auth key

Auth keys are used for API access. A user can have more than one authkey, so if you would like to use separate keys per tool that queries MISP, add additional keys. Use the comment field to make identifying your keys easier.

User

alumno_admin@keepcoding.io

Comment

Allowed IPs

0.0.0.0/0

Expiration (keep empty for indefinite)

YYYY-MM-DD

☐ Read only (it will be not possible to do any change operation with this token)

Submit Cancel

b.8 Generar key en MISP

```

root@ubuntu: /var/ossec/integrations
GNU nano 4.8                                custom-misp.py                                Modified
if not agent or agent["id"] == "000":
    string = '1:misp:{0}'.format(json.dumps(msg))
else:
    string = '1:[{0}] ({1}) {2}->misp:{3}'.format(agent["id"], agent["name"],
    sock = socket(AF_UNIX, SOCK_DGRAM)
    sock.connect(socket_addr)
    sock.send(string.encode())
    sock.close()
false = False
# Read configuration parameters
alert_file = open(sys.argv[1])
# Read the alert file
alert = json.loads(alert_file.read())
alert_file.close()
# New Alert Output if MISP Alert or Error calling the API
alert_output = {}
# MISP Server Base URL
misp_base_url = "https://13.48.162.234/attributes/restSearch/"
# MISP Server API AUTH KEY
misp_api_auth_key = "gECgwGf0jyGKSrHfbFHqv21RwbvH5Ez3Bs4q5lSR"
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^_ Replace    ^U Paste Text^T To Spell   ^_ Go To Line
  
```

b.9 Añadir url MISP y key generada

```
root@ubuntu: /var/ossec/integrations
cape@ubuntu:~$ sudo su
[sudo] password for cape:
root@ubuntu:/home/cape# cd /var/ossec/integrations
root@ubuntu:/var/ossec/integrations# nano custom-misp.py
root@ubuntu:/var/ossec/integrations# nano custom-misp.py
root@ubuntu:/var/ossec/integrations# ls -llh
total 40K
-rw-r--r-- 1 root root 8.3K Feb 14 17:17 custom-misp.py
-rwxr-x--- 1 root ossec 4.3K May 30 2022 pagerduty
-rwxr-x--- 1 root ossec 1.1K May 30 2022 slack
-rwxr-x--- 1 root ossec 3.8K May 30 2022 slack.py
-rwxr-x--- 1 root ossec 1.1K May 30 2022 virustotal
-rwxr-x--- 1 root ossec 6.3K May 30 2022 virustotal.py
root@ubuntu:/var/ossec/integrations# chown root:ossec custom-misp.py
root@ubuntu:/var/ossec/integrations# chmod 750 custom-misp.py
root@ubuntu:/var/ossec/integrations# ls -llh
total 40K
-rwxr-x--- 1 root ossec 8.3K Feb 14 17:17 custom-misp.py
-rwxr-x--- 1 root ossec 4.3K May 30 2022 pagerduty
-rwxr-x--- 1 root ossec 1.1K May 30 2022 slack
-rwxr-x--- 1 root ossec 3.8K May 30 2022 slack.py
-rwxr-x--- 1 root ossec 1.1K May 30 2022 virustotal
-rwxr-x--- 1 root ossec 6.3K May 30 2022 virustotal.py
root@ubuntu:/var/ossec/integrations#
```

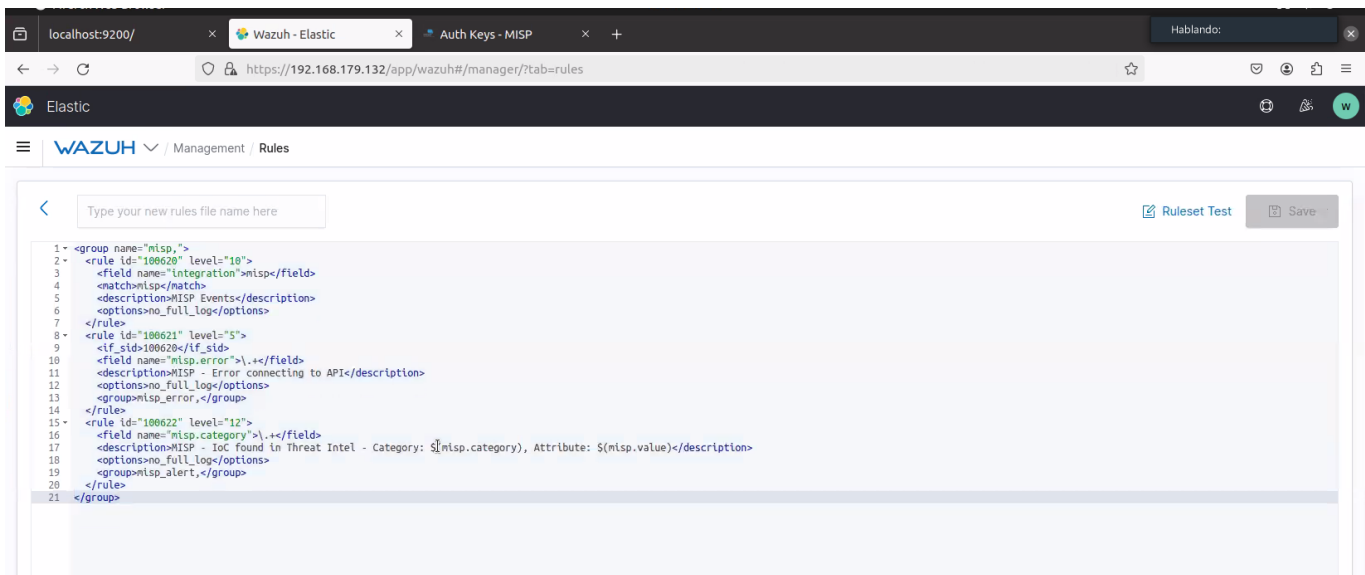
b.10. Integración reglas misp

```
root@ubuntu: /var/ossec/integrations
GNU nano 4.8 /var/ossec/etc/ossec.conf
<!--
Wazuh - Manager - Default configuration for ubuntu 20.04
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>no</logall>
    <logall_json>no</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>ossecm@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>
  </global>
</ossec_config>

[ Read 371 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace  ^U Paste Text ^T To Spell  ^_ Go To Line
```

b.11. Integración reglas MISP en configuración Wazuh



b.12. Reglas MISP en el SIEM Wazuh

Attributes

◀ previous		next ▶													
										Related		Feed			
Date	Event	Org	Category	Type	Value	Tags	Galaxies	Comment	Correlate	Events	hits	IDS	Distribution	Sightings	A
2024-02-04	1	Keepcoding	Payload delivery	sha256	015e3f63a4b336ac9dbfa5304eaf2954277a5d2501d52f5b8b67767c9e8d72c			288D.exe					Inherit event		(0/0/0)
2024-02-04	1	Keepcoding	Payload delivery	sha256	12cbb662f357a3be5dac4e19a58c7079cfc6c180ff52db827640f1a3b74c75d			B7C1.exe					Inherit event		(0/0/0)
2024-02-04	1	Keepcoding	Payload delivery	sha256	06809e78c80df9384578a7a7a12e702443dcd16b88e1f1886b5a73e8805e5424			explorer.exe					Inherit event		(0/0/0)
2024-02-04	1	Keepcoding	Payload delivery	sha256	391e4bc51d7940b48e4064b85104b6a9753c9051fbd3136537ef81bb297ad83			82d763b6cd97ca240a29.exe					Inherit event		(0/0/0)
2024-02-04	1	Keepcoding	Payload delivery	sha256	b7d2c8c40e8a23f0ca789a313a87738b9a53d9011c49784f414336cb5c260edd			B65B.exe					Inherit event		(0/0/0)
2024-02-04	1	Keepcoding	Payload delivery	sha256	ad31543080a03b325fa104d5571ae67b92a90cb99e9e2977ef3f68d77a42671			3928.exe					Inherit event		(0/0/0)
2024-02-04	1	Keepcoding	Payload delivery	sha256	ab112b4b9e691371c579bed0247e2be859cf364ea5c3e420cb2da00b91a8bbf			explorer.exe					Inherit event		(0/0/0)
2024-02-04	1	Keepcoding	Payload delivery	sha256	8ee79c9bc42d30eb7e71032336446d62471cfe82eb3c35db23d95bd211f416b1			8769.exe					Inherit event		(0/0/0)
2024-02-04	1	Keepcoding	Payload delivery	sha256	971bd8f6522c953541d5a9d154329735aa1d47e52d0808f8193ba0553c9c2e			AppLaunch.exe (injecting process - A89F.exe)					Inherit event		(0/0/0)
2024-02-04	1	Keepcoding	Payload delivery	sha256	34d54ad691bb9d9a4c752afbb80b258c904d6e4dca9f788673f53647a0800f9			8769.exe					Inherit event		(0/0/0)

b.13 Atributos MISP del Smokeloader


```
PS C:\Users\Administrator\Desktop> cd C:\Users\Administrator\Desktop\sysmon
PS C:\Users\Administrator\Desktop\sysmon> ./sysmon64.exe -i sysmon.xml

System Monitor v15.14 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Error: Failed to open xml configuration: sysmon.xml
PS C:\Users\Administrator\Desktop\sysmon> ./sysmon64.exe -i sysmonconfig/export.xml

System Monitor v15.14 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Error: Failed to open xml configuration: sysmonconfig/export.xml
PS C:\Users\Administrator\Desktop\sysmon> ./sysmon64.exe -i sysmonconfig-export.xml

System Monitor v15.14 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.50
Sysmon schema version: 4.90
Configuration file validated.
Sysmon64 installed.
SysmonDrv installed.
Starting SysmonDrv.
SysmonDrv started.
Starting Sysmon64..
Sysmon64 started.
PS C:\Users\Administrator\Desktop\sysmon>
```

b.14. Instalación Sysmon

Microsoft-Windows-Sysmon%4Operational Number of events: 1,134

Level	Date and Time	Source	Event ID	Task Ca...
Information	17/02/2024 17:49:11	Sysmon	11	File cre...
Information	17/02/2024 17:49:11	Sysmon	11	File cre...
Information	17/02/2024 17:49:11	Sysmon	11	File cre...
Information	17/02/2024 17:49:11	Sysmon	11	File cre...
Information	17/02/2024 17:49:11	Sysmon	11	File cre...
Information	17/02/2024 17:49:11	Sysmon	11	File cre...
Information	17/02/2024 17:49:11	Sysmon	11	File cre...
Information	17/02/2024 17:49:11	Sysmon	11	File cre...
Information	17/02/2024 17:49:11	Sysmon	11	File cre...
Information	17/02/2024 17:49:10	Sysmon	1	Proces...
Information	17/02/2024 17:49:05	Sysmon	22	Dns qu...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...
Information	17/02/2024 17:48:40	Sysmon	11	File cre...

Event 22, Sysmon

General Details

QueryResults: -
Image: C:\Windows\System32\svchost.exe
User: NT AUTHORITY\LOCAL SERVICE

Log Name: Microsoft-Windows-Sysmon/Operational

Source: Sysmon Logged: 17/02/2024 17:49:05

Event ID: 22 Task Category: Dns query (rule: DnsQuery)

Level: Information Keywords:

User: SYSTEM Computer: DESKTOP-2SJSUJ

OpCode: Info

More Information: [Event Log Online Help](#)

b.15. Logs generados por Sysmon

