Desarrollo de la herramienta MISP para inteligencia de ciberamenazas

Guillem Adolf Mandel Guasch

Resumen— MISP (Malware Information Sharing Platform) es una plataforma de inteligencia de amenazas de código abierto para compartir, almacenar y correlacionar Indicadores de Compromiso (IOCs) de ataques dirigidos e información de vulnerabilidades. No sólo para almacenar, compartir y colaborar en análisis de malware, sino también para utilizar la información de esos indicadores para detectar y prevenir futuros ataques, fraudes o amenazas contra infraestructuras TIC, organizaciones o personas. Durante un incidente de ciberseguridad, los IOCs son pistas y pruebas de una violación de datos. Estas huellas digitales pueden revelar no sólo que se ha producido un ataque, sino a menudo, qué herramientas se utilizaron en el ataque y quién está detrás de ellas. Al ser de código abierto, nos permitirá la integración, programación, creación de reglas y revisión de conexiones para prevenir esos posibles ataques, pero para ello, uno de los principales intereses sobre esta plataforma es obtener y compartir IOCs.

Palabras clave—MISP, amenaza, IOC, vulnerabilidad, malware, automatización, código abierto, SIEM, compartir información.

Abstract— MISP (Malware Information Sharing Platform) is an open source threat intelligence platform for sharing, storing and correlating Indicators of Compromise (IOCs) of targeted attacks and vulnerability information. Not only to store, share and collaborate on malware analysis, but also to use the information from those indicators to detect and prevent future attacks, fraud or threats against ICT infrastructures, organizations, or individuals. During a cybersecurity incident, IOCs are clues and evidence of a data breach. These digital footprints can reveal not only that an attack has occurred, but often, what tools were used in the attack and who is behind them. Being open source, will allow us to integrate, program, create rules and review connections to prevent these possible attacks, but to do so, one of the main interests on this platform is to obtain and share IOCs.

Index Terms—MISP, threat, IOC, vulnerability, malware, automation, open source, SIEM, information sharing.

1 Introducción

En el mundo del IT, el término automatización se usa para generar instrucciones y procesos reproducibles que reduzcan la necesidad de intervención humana en los sistemas informáticos. La automatización de la ciberseguridad por su parte, combina herramientas y procesos para ejecutar actividades que puedan incluir la recopilación y correlación de datos de diferentes sistemas, y la colaboración en la coordinación de los ciclos de vida de respuesta a incidentes y de su gestión. Esto es posible gracias a la inteligencia artificial, cuyo algoritmo está diseñado para tomar decisiones y correlacionar datos.

Por otra parte, los equipos de seguridad se enfrentan continuamente a lidiar con alertas de seguridad 24/7. MISP permite a una organización tener una forma estructurada de almacenar datos sobre las amenazas que ha experimentado, como las IPs, los dominios y las direcciones de correo electrónico, así como con todo comportamiento que ha aprendido sobre esas amenazas. El formato consiste en que la organización tiene un historial de búsqueda de eventos de amenazas y la plataforma conecta de manera automática cualquier dato histórico con los nuevos eventos introducidos en el sistema. Es un motor de búsqueda y un repositorio de los datos y acciones que la organización ha tomado acerca de esos eventos, lo cual puede hacer que una organización sea cada vez mucho más rápida e inteligente cuando se tratan nuevas alertas [1].

En cuanto a la base de datos, su estructura aporta una importante ventaja adicional: la posibilidad de combinar tu base de datos MISP con las bases de datos MISP de otras organizaciones en una única base de datos de gran tamaño y con capacidad de búsqueda. Por otra parte, compartir todos estos datos presenta nuevos retos, ya que no toda la información debería ser compartida con todos. Aquí es donde aparece el término de comunidad, donde uno puede escoger qué compartir, con quién, y hasta dónde puede llegar. Una buena idea puede ser juntarse con otras organizaciones que experimentan amenazas similares. MISP también permite que una organización ingiera información sobre amenazas procedente de fuentes de datos públicos o de otras fuentes de confianza como la policía o los investigadores de seguridad [2].

En este proyecto vamos a emplear Indicadores de Compromiso (IOC). Son datos que surgen a partir de la actividad en un sistema que brindan información sobre el comportamiento de una amenaza y permiten contextualizar los incidentes, clasificar los mismos y que constituyen una postura reactiva. La presencia de *malware*, firmas, *exploits*, vulnerabilidades y direcciones IP son las pruebas típicas que quedan cuando se produce una infracción. Dicho de otra manera, es como llegar a la escena de un crimen y tratar de recrearlo básandonos en las pruebas dejadas. En el caso de una violación de la información, los IOCs pueden

incluir una variedad de pruebas electrónicas dejadas atrás, como un dominio, una dirección IP codificada, un hash, un nombre de archivo, etc. Sin embargo, estos IOCs cambian constantemente, lo que hace imposible un enfoque proactivo para asegurar la empresa. Dado que los IOCs representan un método reactivo para rastrear a los cibercriminales, para cuando se encuentra uno, hay una alta probabilidad de que ya haya sido comprometido [3].

Algunas de las características y requisitos que cumple la plataforma MISP son los siguientes:

- Una eficiente base de datos de IOCs que permiten almacenar información sobre muestras de *malware*, incidentes y atacantes.
- Una correlación automática que encuentre relaciones entre atributos, indicadores de *malware*, campañas de ataques y/o análisis de datos.
- Utilizar funcionalidades avanzadas de filtrado para cumplir con la política de compartición de cada organización, junto con una una interfaz de usuario intuitiva para que los usuarios finales puedan crear, actualizar y colaborar en los eventos y atributos.
- Una interfaz gráfica para navegar sin problemas entre los eventos y sus correlaciones. Una funcionalidad de gráfico de eventos para crear y ver las relaciones entre objetos y atributos [4].

A continuación, hablaremos sobre los avances en ciberseguridad y su automatización, presentaremos los objetivos, la metodología y planificación que se ha llevado a cabo, y seguidamente, se explica el desarrollado e implentación de esta herramienta. Finalmente, se muestran los resultados que se han obtenido y las conclusiones presentadas, junto con los futuros avances en los que ya se está trabajando.

2 ESTADO DEL ARTE

Hoy en día, en las empresas donde se trabaja en ciberseguridad, ya sea una compañía dedicada, una consultora con su departamento o un centro de operaciones (SOC), no se acostumbra a ver herramientas encargadas de llevar a cabo tareas rutinaras y que a menudo demandan mucho tiempo. Gracias a la gran cantidad de la información que manejan los sistemas corporativos detectar una anomalía de forma manual puede tomar gran cantidad de tiempo, lo que representa también una gran ventaja para los atacantes.

El desarrollo de la automatización y optimización de plataformas web para equipos de ciberseguridad es algo reciente. Desafortunadamente, el robo de datos sensibles y la intrusión no autorizada a sistemas corporativos, es hoy en día un negocio muy lucrativo.

El proyecto MISP es la principal plataforma de inteligencia de amenazas de código abierto en combinación con un estándar abierto para la inteligencia de amenazas. Van surgiendo oportunidades para que todos los interesados (desarrolladores, colaboradores y usuarios) del proyecto MISP compartan sus experiencias, aprendan nuevos casos de uso y mejoren sus capacidades de inteligencia de amenazas, así como debatir abiertamente sobre el uso actual de MISP, los desarrollos futuros y la integración con los ecosistemas de seguridad en general [5].

En cuanto a la compartición de datos, por un lado, actualmente los datos que se almacenen están inmediatamente disponibles entre socios: se almacena la identificación del evento en el sistema y se informa mediante notificaciones de correo electrónico firmadas y encriptadas.

Por otro lado, compartir con máquinas mediante la generación de reglas, exportaciones de texto o CSV, MISP permite importar automáticamente los datos en el sistema de detección, lo que permite un resultado más rápido de intrusiones.

Por último, si se ejecuta MISP internamente, los datos también podrán cargarse y descargarse automáticamente desde y hacia instancias MISP alojadas externamente. Gracias a esta automatización y al esfuerzo de los demás, ahora se dispone de valiosos IOCs sin ningún trabajo adicional.

3 OBJETIVOS

3.1 Objetivo Principal

El objetivo principal es el estudio e implementación de MISP para integrar y procesar la salida hacia los gestores de eventos e información (SIEMs) de clientes reales.

3.2. Objetivos específicos

Este trabajo consta de varios objetivos secundarios que complementarán y darán soporte al objetivo principal:

3.2.1. Compartir y obtener IOCs.
Información distribuida y segura. Se mejorará la prevención de ataques obteniendo información de otras empresas de ciberseguridad que compartan sus IOCs.

3.2.2. Orquestar y automatizar la seguridad del MISP.
Estableceremos alertas automáticas y configuraremos sus sistemas para bloquear las amenazas que se hayan identificado con anterioridad.

3.2.3. Implementar procesos para módulos de expansión de MISP.
Se procederá a implementar procesos con Python para módulos de expansión para ampliar MISP con sus propios servicios o activar sus módulos ya disponibles.

3.2.4. Procesar la salida de SIEMs de clientes reales.
Obtendremos la salida de los SIEMs de cada uno de los clientes, implementando un MISP que proceda su salida.

- *E-mail de contacto: GuillemAdolf.Mandel@autonoma.cat*
- *Mención realizada: Tecnologiass de la Información*
- *Trabajo tutorizado por: Miguel Carpio (Departamento de Ingeniería de la Información y de las Comunicaciones)*
- *Curso 2021/22*

4 METODOLOGIA

La metodología aplicada en este trabajo es de tipo Agile, concretamente la metodología Kanban [6]. La visualización más básica de esta está compuesta por tres columnas: "Por hacer", "En proceso" y "Hecho". No requiere configuración y sirve como fuente de información durante el proceso para identificar los problemas.

Se ha procedido a la investigación de la plataforma y sus funcionalidades básicas, implementando su construcción en un entorno de pruebas gracias a la creación de equipos de máquinas virtuales para el despliegue de MISP y probar un par de bases de datos distintas.

Una de las incorporaciones son los servicios virtuales de GitLab. Con ellos, implementamos una nueva metodología: DevOps [7], centrada en la comunicación, colaboración e integración entre desarrolladores software y operadores profesionales de IT. Queremos obtener un mayor control, trazabilidad y una implementación automática para adaptarnos a nuevas versiones y a los cambios continuos que puedan surgir. La idea de realizar el deploy del sistema es de manera completamente automatizado. Este no solamente copia los cambios de forma automática en el servidor, sino que también está íntimamente conectado con la integración continua [8].

5 PLANIFICACIÓN

El trabajo se ha dividido en tres partes. Para agilizar la tarea, se ha diseñado un diagrama de Gantt (ver la Figura 1 del Apéndice A) para organizar y mostrar gráficamente todo el proceso del trabajo.

Primeramente, se planifica la investigación inicial sobre el software. Se establece un primer contacto gracias al alta de las primeras máquinas virtuales en donde se instalan las dos instancias de MISP. Cada una de ellas contiene una base de datos distintas para así poder realizar distintas pruebas tanto de integración como de posterior funcionamiento.

La segunda parte se centra en la arquitectura y todas las secciones que se han tratado en ella: investigación, desarrollo, programación e implementación de MISP, para cumplir con el objetivo de tener una aproximación muy cercana a la integración con el SIEM del cliente, para este caso Qradar, que también tendrá su MISP y su laboratorio interno junto a sus IOCs y sus bases de datos.

La tercera parte se centra en la modificación de la arquitectura y la integración de nuevos equipos con GitLab, ya que cada vez se van sumando más actividades y es ideal para este tipo de desarrollos, y con acceso a las diferentes herramientas web que ayudan a reportar información maliciosa (*IBM X-Force Exchange* [9], *VirusTotal* [10], *Abuseipdb* [11], etc.). Por último, la integración de la quinta y última máquina virtual, llamada *Feed Server*, que será una máquina similar a la segunda instancia de MISP por su base de datos pero que configuraremos para conectar con internet y será abierta para usuarios externos.

6 DESARROLLO

En este apartado se explica de manera detallada en que ha consistido todo el diseño e implementación de la plataforma MISP aplicada en una compañía para procesar la salida hacia el SIEM de un cliente real, conectando desde diferentes bases de datos, pasando por la red de la empresa y llegando así a la red del cliente junto al acceso abierto para todo usuario que quiera recopilar la información que crea necesaria para la seguridad de su entorno.

6.1 Entorno de trabajo

Primeramente, contamos con dos servidores virtuales donde van instaladas las instancias de MISP. Se han decidido que sean dos ya que contendrán bases de datos distintas y para el entorno de pruebas será mejor.

Las especificaciones de estos equipos son las siguientes: 4 cores, 8GB de RAM, 30 GB de disco (este espacio de memoria secundario se puede modificar a la larga, ya que depende del número de programas que se introduzcan en Gitlab y en la instancia de Docker que hemos creado) y dentro del mismo rango de red que el SIEM Qradar.

En cuanto a los lenguajes de programación, usamos Python y PHP para programar las distintas funciones, como por ejemplo la conexión con las diferentes APIs de herramientas web. Usamos también Apache para la conexión con las bases de datos, la aplicación de Visual Studio Code para el código y Gitlab, fácil de instalar, configurar, mantener y visualmente sencillo de comprender.

6.2 Arquitectura

En la Figura 1 del Apéndice B se puede apreciar la arquitectura completa y los diferentes componentes de MISP. Consta principalmente de dos partes:

- **Red interna de la empresa**. Se basa en cinco máquinas virtuales. Hemos decidido dedicar una máquina por servicio, ya que de este modo se encuentra más ordenado y en caso de fallo, mantenimiento o actualización, solo afectaría a uno solo. En cuanto a las tareas de desarrollo: por un lado, se han desplegado las instancias de MISP, la primera contendrá una base de datos sin filtrar y la segunda ya con los eventos filtrados, listos para proceder a filtrar por IOCs y enviar a cliente. Por otro lado, la idea ha sido ubicar Gitlab dentro de Docker y usarlo con al menos un *runner* para poder ejecutar las tareas CI/CD en esta misma instancia de Docker. La intención es poner un servidor proxy para redireccionar a los diferentes programas dentro de los containers, Gitlab y las diferentes tareas CI/CD desplegadas, con el objetivo de entregar más datos filtrados a las instancias de MISP. Todo ello se tendrá que poder acceder a la IP de la máquina virtual des de dentro de la VPN.

- **Red del cliente**: las diferentes aplicaciones y plataformas del cliente son principalmente dos. Por un lado, tenemos la conexión al SIEM del cliente, y este conectado a los firewalls que contendrán toda la información de los IOCs para que no se bloqueen todas esas alertas que el firewall haya denegado el tráfico (el *Threat Intelligence* lo que permite es ir haciendo *pulls* para extraer esos IOCs cada cierto tiempo).

6.2.1 Red de la empresa

A continuación, vamos a explicar detalladamente cada una de las partes que hemos integrado dentro de nuestra red interna.

6.2.1.1 Bases de datos.

Todo parte de los recursos o *feeds*, obtenidos de fuentes remotas o locales, que generen terceros con información y que podemos recoger para añadirla en nuestra instancia de MISP.

Esta información viene estructura en formato MISP, CSV o texto plano, y contiene indicadores (IOC) que pueden importarse automáticamente en intervalos predefinidos.

Importar estos *feeds* es muy sencillo, por lo que se puede reunir una gran cantidad de fuentes externas de información sin necesidad alguna de programación. Podremos seleccionar tantos *feeds* como deseemos según nuestras necesidades. En este caso, seleccionamos y habilitamos todos los *feeds* predefinidos para así tener más opciones de encontrar relaciones más adelante. Todos ellos, irán dentro de la primera máquina virtual que a la vez es una de las dos instancias de MISP que forman parte de esta arquitectura.

Para habilitar un *feed* para su almacenaniemto en caché, se debe marcar el campo de activación habilitado para beneficiarnos de estos *feeds* en nuestra instancia local de MISP (ver Figura 1). Es importante configurar correctamente según nuestras necesidades el valor que activa estos recursos, ya que, si no está activo, la correlación únicamente se nos muestra a nosotros, mientras que, si está activo, todas las correlaciones se muestan al resto de usuarios [12].

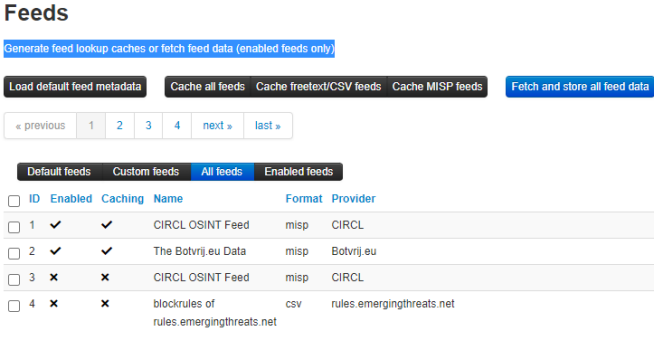

Fig. 1: lista de *feeds* por defecto

Una vez está todo correctamente configurado y tenemos los *feeds* preparados, ya podemos comenzar a crear eventos (ver la Figura 2). Un evento no es más que el registro de una muestra con sus IOC que hayamos descubierto o se nos haya remitido desde un SIEM, y así poderlo correlacionar con nuestra lista de *feeds*.

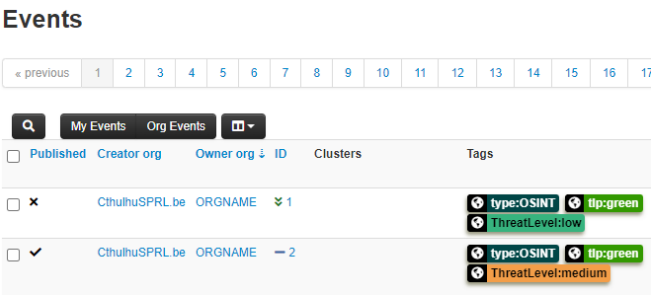

Fig. 2: lista de eventos

6.2.1.2 Instáncias de MISP1 y MISP2

La máquina de la primera instancia de MISP tiene que estar en la misma subred de MISP porque hay algunos programas que se van a comunicar con él.

MISP1 contiene todas esas listas de *feeds* y eventos que hemos visto en el apartado anterior, pudiendo almacenar toda información que creamos necesaria. También recibirá IOCs de otras herramientas como aplicaciones web, las cuales almacenan grandes cantidades de datos y serán útiles para tener en cuenta aún no habiendo sido reportados hasta el momento.

El próximo paso es filtrar esos eventos que hayan sido catalogados como maliciosos a partir de etiquetas o *tags* que podemos observar en la Fig 2. y enviarlos hacia la máquina de MISP2.

MISP2 contendrá los eventos filtrados como maliciosos y su función es extraer y filtrar esos IOCs repotados, a través también de tags, hacia el cliente.

6.2.1.3 Gitlab + Herramientas

En esta máquina tenemos una instancia autogestionada de Gitlab, es donde los desarrolladores crearán y realizarán cambios en el código para tener una mayor seguridad a la hora de hacer las actualizaciones que se consideren oportunas.

Además, en este servicio virtual se realizará en un *deployment* del código hacia la siguiente máquina usando *pipelines* [13]: el proceso de tomar el código del control de versiones (Gitlab) y ponerlo a disposición de los desarrolladores de la aplicación de forma automatizada. Se requiere una canalización automatizada que construya, pruebe y despliegue su aplicación, junto con la infraestructura necesaria para trasladar las nuevas adiciones de código desde el control de versiones a la producción.

Uno de los desarrollos para esta máquina ha sido conectar con herramientas de licencia gratuita como *IBM X-Force Exchange* (ver Figura 3), una plataforma para compartir inteligencia sobre amenazas que se puede utilizar para investigar sobre amenazas de seguridad. Los usuarios tienen acceso a todas las funciones: obtener datos de un IOC, comentar, compartir y realizar recopilaciones.

Fig. 3: Buscador de IOCs de *IBM X-Force Exchange*

6.2.1.4 Gitlab runner

La función principal de esta máquina es el envío de datos a las instancias de MISP a partir del deploy del código del servicio anterior. Dominios, URLs, IPs y otros tipos de IOC ya reportados como maliciosos serán datos filtrados enviados directamente a la instancia de MISP2, mientras que, en el caso contrario, los datos no catalogados como tal, pero que tengan relación al evento o caso de uso que estemos tratando en el momento, irán a parar a MISP1.

Está máquina contiene un Gitlab *runner* con un container de las aplicaciones (en el punto 7 de este proyecto se explica detalladamente su función), más en la figura 2 del Apéndice B se puede observar el flujo de trabajo de Gitlab.

6.2.1.5 Feed Server

Es una máquina virtual que contendrá datos para que los usuarios externos puedan acceder a ellos. Está conectada tanto a internet como a la instancia de MISP2, siendo muy similar a esta, ya que también contendrá la información filtrada como maliciosa. Habrá un flujo de datos constantes por las dos bandas para que cada cierto tiempo programado se actualicen los datos y se eliminen aquellos eventos caducados.

Contendrá también los *feeds* de MISP y dependiendo del porcentaje de los IOCs reportados, irán a parar a una de las dos instancias de MISP, por lo que también estará conectada a la máquina de MISP1.

Hay que tener unos límites de filtrado coherentes. Dicho de otra manera, hasta Microsoft tiene IPs y dominios reportados con un 20% de maliciosidad, por lo que ese porcentaje no puede ser demasiado bajo ya que sino lo llenaríamos de IOCs reportados y saturaríamos el servicio. Otro apunte importante es que la máquina Feed Server es la que nutre a los elementos de seguridad de clientes, por lo que, si hay IOCs erróneos, podría afectar a la infraestructura.

Fig. 4: vista actual de la VM Feed Server

En la Figura 4 podemos observar como se muestra la página principal, donde podemos, por el momento, realizar dos funciones:

La primera es poder subir un archivo con IPs y el programa parseará todas las IPs existenentes. El programa te devolerá una vista con una tabla para cada IP con los correspondientes resultados e insertará los datos nuevos a la base de datos.

La segunda función es poder descargar la lista de IPs con una cierta severidad (*Low, Medium, High, Very High*). El programa te descargará un CSV con todas las IPs con la severidad seleccionada.

6.2.2 Red del cliente

Recordemos uno de los objetivos gracias a la implementación de un MISP: procesar la salida hacia los SIEMs para cada uno de los clientes.

Un SIEM (*Security Information and Event Managment*), es un sistema de seguridad capaz de detectar, responder y neutralizar las amenazas informáticas. Permite tener control absoluto sobre la seguridad informática de la empresa y, al tener administración total sobre todos los eventos que suceden segundo a segundo, resulta más fácil la detección y actuación de forma inmediata frente a cualquier tendencia o patrón fuera de lo común [14].

Los IOCs filtrados a partir de la instancia MISP2 serán enviados al SIEM a través de intervalos generados cada cierto tiempo (*pulls* cada 30 minutos), teniendo en cuenta que encontraremos eventos caducados y que ya no serán necesarios mantener. La conexión a los SIEMs del cliente será a través del servidor *TAXII* y las instancias de MISP del cliente. Estas también enviaran la información de los IOCs a los firewalls.

Hay que destacar que el SIEM no se comunica, no toma acciones, solo ingesta logs y no da ofensas/alertas de todos esos casos de uso que se configuran. Eso significa que, si aparecen hashes erróneos en el QRadar, p.ej. un hash de chrome.exe el cual es un falso positivo, no habrá problema.

7 INTEGRACIÓN CONTINUA

En el desarrollo de software, los servicios de control de versiones son esenciales para administrar los repositorios Git [15] de su proyecto: Gitlab es un servicio web de control y desarrollo de software colaborativo basado en Git.

Es una única aplicación para todo el ciclo de vida del desarrollo de software. Desde la planificación del proyecto y la gestión del código fuente hasa el CI/CD, la monitorización y la seguridad.

Hemos ubicado Gitlab dentro de Docker y con al menos un *runner* en la misma instancia para una mayor efectividad en la implementación de este proyecto.

Gitlab *runner* trata de un entorno al que Gitlab puede enviar tareas para su ejecución y recibir los resultados de estas. Dicho de otra manera, nosotros configuramos una serie de tareas en cada uno de nuestros proyectos, Gitlab las interpreta y transforma en *pipelines*, se las envía al *runner*, éste las ejecuta y devuelve los resultados de las tareas

a Gitlab, que a su vez los representa de manera gráfica sobre los mismos *pipelines*.

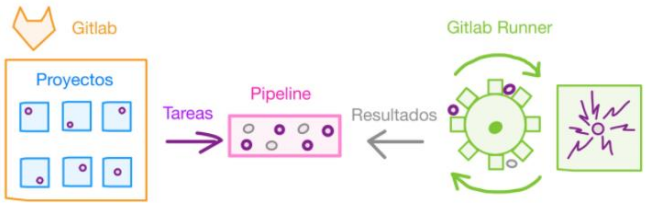

Fig. 5: Representación gráfica de Gitlab runner [16].

El servidor donde esté instalado nuestro *runner* debe poder tener una comunicación por red con el servidor donde esté instalado nuestro Gitlab.

Por motivos de seguridad, para registrar un *runner* necesitamos un *token*. Hay varios sitios en la interfaz de nuestro Gitlab desde los cuales podemos obtener dicho *token*.

Dependiendo del tipo de *runner* que registremos en nuestro Gitlab, este podrá ejecutar solo determinadas tareas. Sin embargo, gracias a las etiquetas o *tags*, podemos filtrar todavía más las tareas que nuestros *runners* pueden ejecutar. Básicamente, a la hora de registrar el nuestro podemos especificar una serie de palabras clave. Haciendo eso, podrá ejecutar únicamente las tareas que tengamos definidas con esas mismas palabras.

El último punto es definir el tipo de mecanismo que nuestro *runner* deberá usar para ejecutar las tareas. Como ya dijimos al principio de la entrada, un *runner* es la puerta a un entorno donde podemos ejecutar nuestras tareas. Dicho entorno puede ser el mismo servidor donde se está ejecutando el nuestro (shell), algún tipo de maquina virtual dentro del servidor (Docker) o un servidor remoto (SSH**).**

8 CYBER THREAT INTELLIGENCE

A través del conocimiento que aporta CTI podemos llegar a ser capaces de recolectar información sobre un cibercriminal con el fin de detectar cuál es su actividad maliciosa relacionada, que patrones suele utilizar y entender cuál es el comportamiento empleado detrás de sus ataques.

El fin del CTI no es recolectar únicamente indicadores de compromiso sobre amenazas, sino generar ese conocimiento mencionado alrededor del adversario con el objetivo de reducir el posible riesgo que pudiera ocasionar a la empresa u organización, además de anteponerse a sus ataques y contrarrestarlos. Para analizar una amenaza en su conjunto es vital detectar una serie de datos que ayuden a identificar el actor o grupo criminal detrás de un ataque. Por este hecho es bastante importante aplicar técnicas de análisis basados en hipótesis y evidencias a través de un proceso analítico de todos los datos recolectados [16].

Por medio del contexto podemos llegar a identificar el por qué, por quién y el cómo es efectuada la amenaza. Además, dicho contexto también puede facilitar información sobre qué tipo de empresas son objetivos y pueden llegar a verse afectadas frente a una amenaza concreta. Muchas de estas están dirigidas a un sector profesional específico, una tecnología vulnerable o incluso a un país en particular. Conociendo el contexto de una amenaza, muchas empresas pueden llegar a identificar si dicho incidente puede llegar a afectarles, priorizar de una manera más eficiente o bien saber cómo mitigar la amenaza.

8.1 TAXII

Trusted Automated Exchange of Intelligence Information (TAXII) es un protocolo de capa de aplicación para la comunicación e intercambio de información sobre ciberamenazas (CTI) a través del protocolo HTTPS, de forma sencilla y escalable].

TAXII permite a las organizaciones compartir CTI mediante la definición de una API y un conjunto de requisitos para los clientes y sus servidores. Como se muestra a continuación, TAXII define dos servicios principales para soportar una variedad de modelos de intercambio comunes.

Nosotros usamos el servicio llamado *Collection*, donde guardamos los atributos de CTI. Este servicio es una interfaz a un repositorio lógico de objetos CTI proporcionado por un servidor TAXII que permite a un productor alojar un conjunto de datos CTI que pueden ser solicitados para los consumidores: los clientes y los servidores TAXII intercambian información en un modelo de solicitud-respuesta.

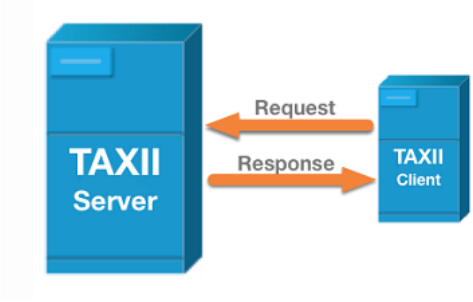

Fig 6: TAXII server: Collection model

El servidor puede ser utilizado por un cliente TAXII externo para recuperar datos de una organización de usuarios conectados al hilo y de cualquier comunidad o fuente a la que el usuario tenga acceso para conectarse a un servidor externo [17].

Una instancia del servidor de TAXII puede soportar una o más APIs. Específicamente, hemos usado la API de IBM X-Force para la conexión con las instancias de MISP del cliente y estas directamente con los firewalls, que recibirán sus peticiones. Entre los objetivos está el de crear algunos feeds personalizados y enriquecer los datos de inteligencia de amenazas.

9 RESULTADOS

Se ha conseguido cumplir con los objetivos propuestos al inicio de este trabajo y solo falta la aprobación del cliente para gestionar nuestros IOCs y procesar la salida hacia su MISP.

Hemos conseguido levantar toda la arquitectura e implementar las instancias de MISP junto al uso de Gitlab, con el resultado de la gestión en el desarrollo del código y de la obtención de IOCs gracias a la licencia de X-Force de IBM, una de las herramientas del Cloud.

Tenemos ya trabajando una lista de eventos y otra de IOCs filtrados, provinientes de varias bases de datos, para habilitar en los diferentes equipos, pasando por MISP2 y llegando a la red del cliente, aplicando los recursos necesarios de todos aquellos datos catalogados como maliciosos.

Fig. 7: Etiquetas de los eventos (***Red tag: high***) en MISP1

```
misp1 = misp_instance("https://[redacted]/",
                      "KcVrwSgyAHkL5hzDun8oAN8b4VaBQ8BDqGkKiIy7")
misp2 = misp_instance("https://[redacted]/",
                      "Yrch6gvV2COQ97Gk8PXEHr6yG3gL5utjZ7vD7ehK")

misp1.createThreatLevelTag()

filter = ['ThreatLevel:high']
a_month = relativedelta(months=6)
aux = datetime.now()-a_month
date_From = str(round(aux.timestamp()))

events = misp1.getFilteredEvents(filter, date_From)

misp2.push(events)
```

Fig. 8: Filtrado por alertas altas “high”

filename	Cybersecurity_EU_Common_Sec_and_Defence.doc
md5	ee1b63ac2915999ae0951f87a9a91958
link	https://twitter.com/h2jazi/status/1436391537416540160
ip-dst	206.198.151.187
hostname	iamnotthec2.ohl.io
md5	290d8e8524e57783e8cc1b9a3445dfe9
sha256	fd8a5313bf63f5013dc126620276fb4f0ef26416db48ee88cbaaca4029df1d73

Fig. 9: IOCs reportados en MISP2

9.1 GITLAB

Hemos realizado procesos con Python para módulos de expansión para ampliar la plataforma, que son los que nos han dado, entre otras cosas, el filtrado de datos para enviar la información de una máquina virtual a otra.

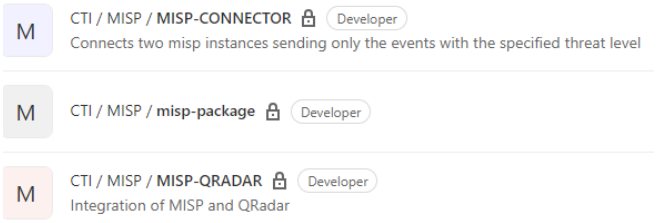

Fig. 10: proyecto ubicado en Gitlab

9.2 CYBER THREAT INTELLIGENCE

La inteligencia sobre amenazas reduce la presión de múltiples maneras:

- Identificando y descartando automáticamente los falsos positivos.
- Enriquecer las alertas con contexto en tiempo real, como las puntuaciones de riesgo personalizadas
- Comparando la información de fuentes internas y externas.
- Los usuarios identifican los riesgos 10 veces más rápido de lo que lo hacían antes de integrar la inteligencia sobre amenazas en sus soluciones de seguridad, lo que les da días más de tiempo de media para responder a las amenazas en un sector en el que incluso los segundos pueden ser importantes.

10. LÍNIAS FUTURAS

A partir de haber cumplido con los objetivos del proyecto, vamos a comentar una serie de mejoras interesantes de incorporar en un futuro breve para mejorar la arquitectura y el funcionamiento de la plataforma (ver Figura 3 del Apéndice B).

El siguiente paso será añadir una máquina virtual la cual estará conectada a las dos instancias de MISP, y recibirá los IOCs de MISP2. Esta máquina, contiene varios conceptos nuevos.

10.1 ELASTICSEARCH

Por un lado, tenemos Elasticsearch, un motor de analítica y análisis distribuido que se usará para poder escalar el nivel de búsqueda a una variedad de casos de uso mucho mayor: búsqueda de sitios web, aplicaciones, analíticas de logs, monitoreo de contenedores, etc.

Kibana [18] es una interfaz de usuario gratuita y abierta que te permite visualizar los datos de Elasticsearch y realiza funciones como rastrear compartir un enlace o exportar archivos CSV entre los miembros de un proyecto.

10.2 HONEYPOTS

Por la banda del cliente, se añadirá un servidor Honeypot. Es un equipo que se usa como cebo para atreaer atacantes, con el objetivo de detectar el ataque antes de que afecte a otros sistemas y trasladar dicha información hacia nuestra máquina de Elasticsearch.

La idea mediante la implementación de este sistema es hacerle creer a un atacante que está apuntando a un sistema real, sin embargo, estará desplegando sus actividades maliciosas en un ambiente controlado por nosotros. Crearemos datos falsos, como por ejemplo un usuario con un nombre atractivo para cuando un atacante se ponga a escanear y se encuentre a este, sea de los primeros que ataque.

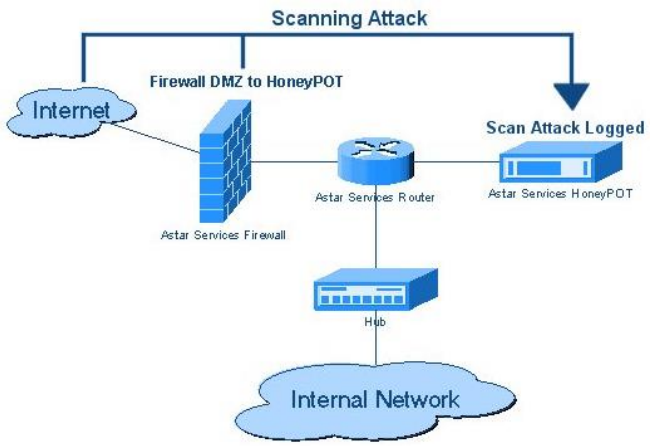

Fig. 11: sistema de seguridad Honeypot

Podemos implementarlos en el interior de la red del cliente, que puede servir para indicarnos que un atacante ya obtuvo acceso a nuestra red y está intentando realizar un movimiento lateral o bien, en el exterior de nuestra red, para detectar ataques externos. Por otra parte, podemos tener varios honeypot instalados en nuestra red y conformar lo que se conoce como un honeynet.

El análisis de la información generada en un Honeypot interno puede ayudarnos a reducir el tiempo de detección sobre el tiempo que lleva el atacante permaneciendo en nuestra red des de que consiguió entrar. Esta información debería servirle al equipo de respuesta ante incidentes o al SOC de la organización para implementar los controles o mitigaciones apropiadas [19].

11 CONCLUSIONES

Se ha logrado la implementación de MISP y está listo para la integración y proceso de salida hacia el SIEM del cliente.

Se ha podido compartir y obtener IOCs de las diferentes bases de datos de manera segura, también de las herramientas web gracias a la licencia gratuita de IBM X-Force.

Se ha automatizado la seguridad de MISP estableciendo alertas automáticas y se han añadido etiquetas para que el sistema bloquee las amenazas que se hayan identificado previamente como malicosas.

A la espera de configurar los sistemas para bloquear automáticamente las amenazas ya registradas, se ha procedido a implementar scripts en Python para módulos de expansión para ampliar MISP con sus servicios de GitLab y Docker.

MISP ofrece soluciones centrado en la seguridad de las empresas, hemos visto que es fácilmente escalable y ya hay las propuestas de futuro sobre la mesa para seguir añadiendo módulos y obtener así una mayor efectividad, así que no es un proyecto que ya haya terminado. La recogida de información se convierte en algo fundamental para las empresas ya que también permite mejorar las capacidades de investigación y ayuda a cumplir los mandatos de conformidad.

AGRADECIMIENTOS

Primeramente, me gustaría agradecer a mi tutor de trabajo Miguel Carpio por todas sus recomendaciones, soporte y ayuda que me ha ofrecido durante todo el proceso.

Segundo, a mi superior dentro de la empresa, Jose, por confiar en mi para este proyecto y a mis compañeros de trabajo por los ánimos y hacer de los turnos un tiempo más ameno.

Por último, también agradecer a mi familia por aguantarme en mis peores días y a mis amigos por el ánimo que me han dado en los momentos de bajona, sois mi soporte incondicional, gracias.

REFERENCIAS

[1] CIRCL, Team MISP Project | An Introduction to Cybersecurity Information Sharing | Available: https://www.misp-project.org/features.html [Accessed: September - 2021]

[2] Cynthia Wagner, Alexandre Dulaunoy | MISP – The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform | CIRCL – Computer Incident Response Centre, Luxembourg | October 2016.

[3] Crowdstrike | Indicators of attack versus indicators of compromise | Whitepaper Available: http://www.crowdstrike.com [EDR]

[4] MISP/misp-book: Threat Intelligence Sharing Platform – GitHub | Available: https://github.com/MISP/MISP

[5] MISP Instance requirements [Online]. Available: https://www.circl.lu/doc/misp/quick-start/ [Last modified: Sun Feb 28 2021].

[6] Rajat B. Wakode, Laukik P. Raut | Overview on Kanban Methodology and its Implementation | Department of Mechanical Engineering, Nagpur | Vol. 3, 2015

[7] DevOps [Online]. Available: https://www.paradigmadigital.com/techbiz/el-legendario-origen-del-movimiento-devops/

[8] Integración continua: eficàcia en proyectos de desarrollow web. Available: https://www.hostgator.mx/blog/deploy-en-programacion/

[9] IBM X-Force Exchange | https://exchange.xforce.ibmcloud.com/

[10] VirusTotal Analyze, "suspicious files and URLs to detect types of *malware*, automatically share them with the security community | Available: https://www.virustotal.com/gui/home/upload

[11] AbuseIPdb, "making the internet safer, one IP at time | Available: https://www.abuseipdb.com/

[12] MISP: Introducción e instalación. Available: https://fwhibbit.es/misp-introduccion-e-instalacion [May 15, 2018]

[13] Sonia Chhabra, Rishabh Sethia | Cloud DevOps CI – CD Pipeline | Computer Science, Sharda University | April 2021

[14] FireEye, What is SIEM. Available: https://www.fireeye.com/products/helix/what-is-siem-and-how-does-it-work.html

[15] Install self-managed Gitlab. Available: https://about.gitlab.com/install/?version=ce

[16] Md Sahrom Abu, Siti Rahayu Selamat | Cyber Threat Intelligence-Issue and Challenges | Indonesian Journal of Electrical Engineering and Computer Science, Malaysia | Vol.10, April 2018.

[17] Julie Connolly, Mark Davidson | The Trusted Automated eXchange of Indicador Information (TAXII) | MITRE Corporation | August 2012

[18] Neel Shah, Darryl Willick | A framework for social media data análitics using Elasticsearch and Kibana | part of Springer Nature | 2018

[19] Neha Titarmare, Nayankumar Hargule | An Overview of Honeypot Systems | Department of Computer Science & Engineering, India | Vol-7, Feb 2019.

APÉNDICE A

A.1. Diagrama de Gantt

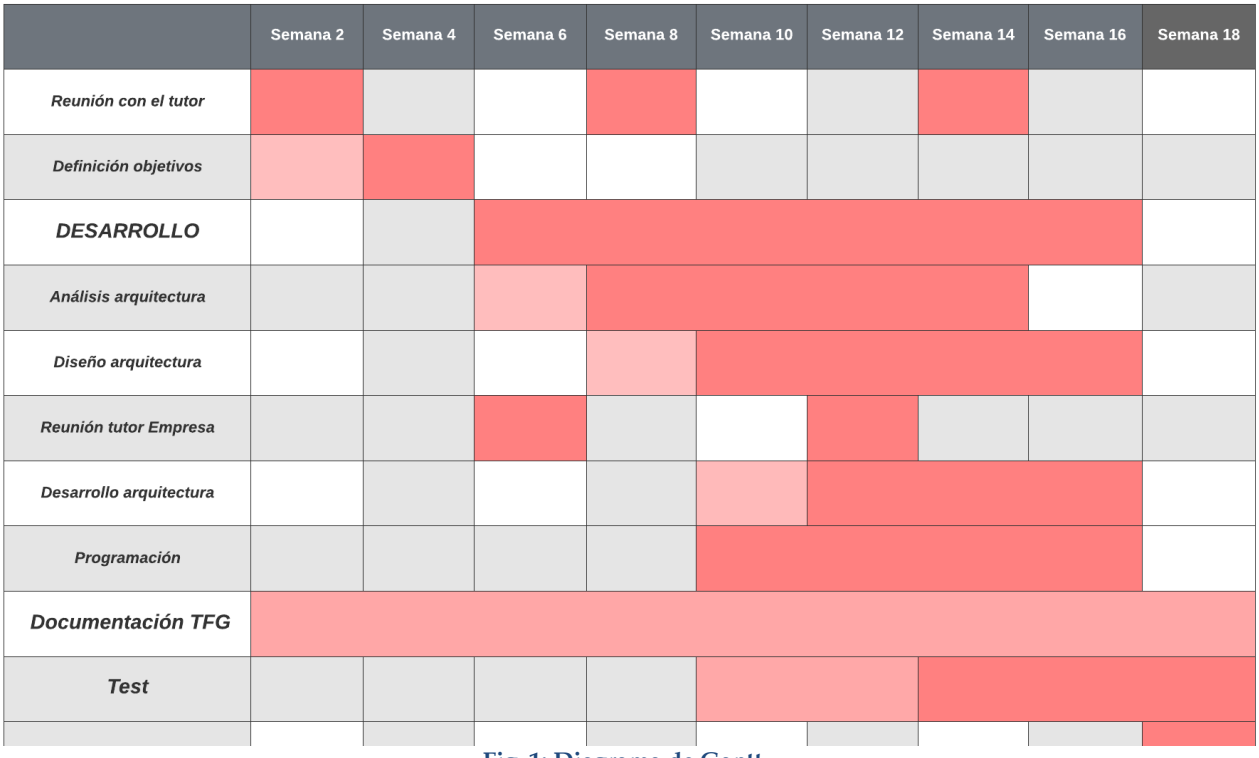

Fig. 1: Diagrama de Gantt

APÉNDICE B

B.1. Arquitectura MISP

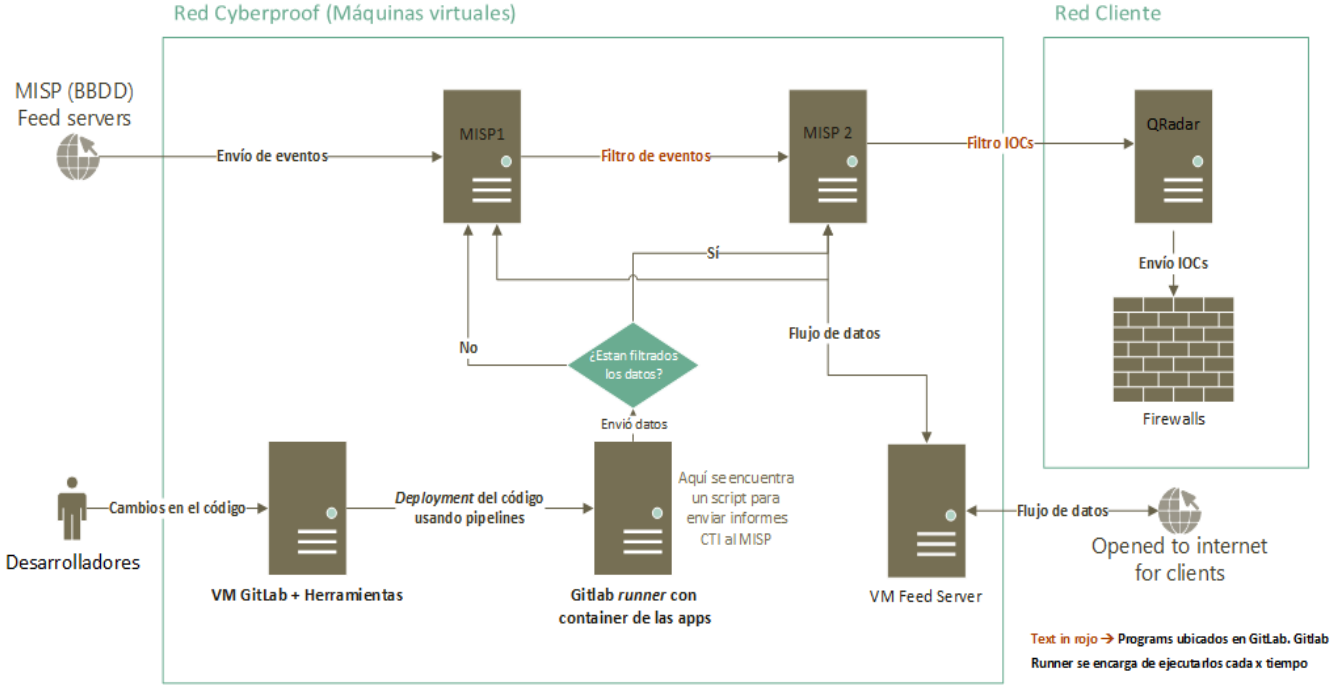

Fig. 1: Arquitectura de MISP

B.2. GitLab Workflow

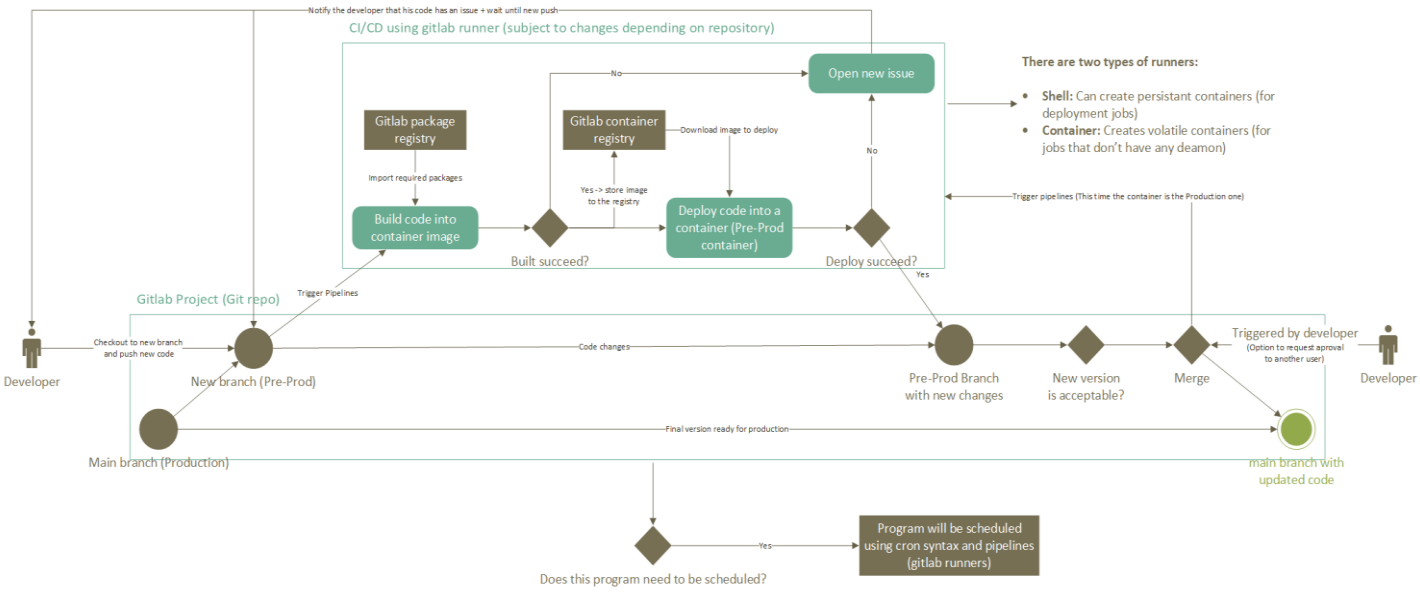

Fig. 2: GitLab Workflow

B.3. Futuros avances en la arquitectura de MISP

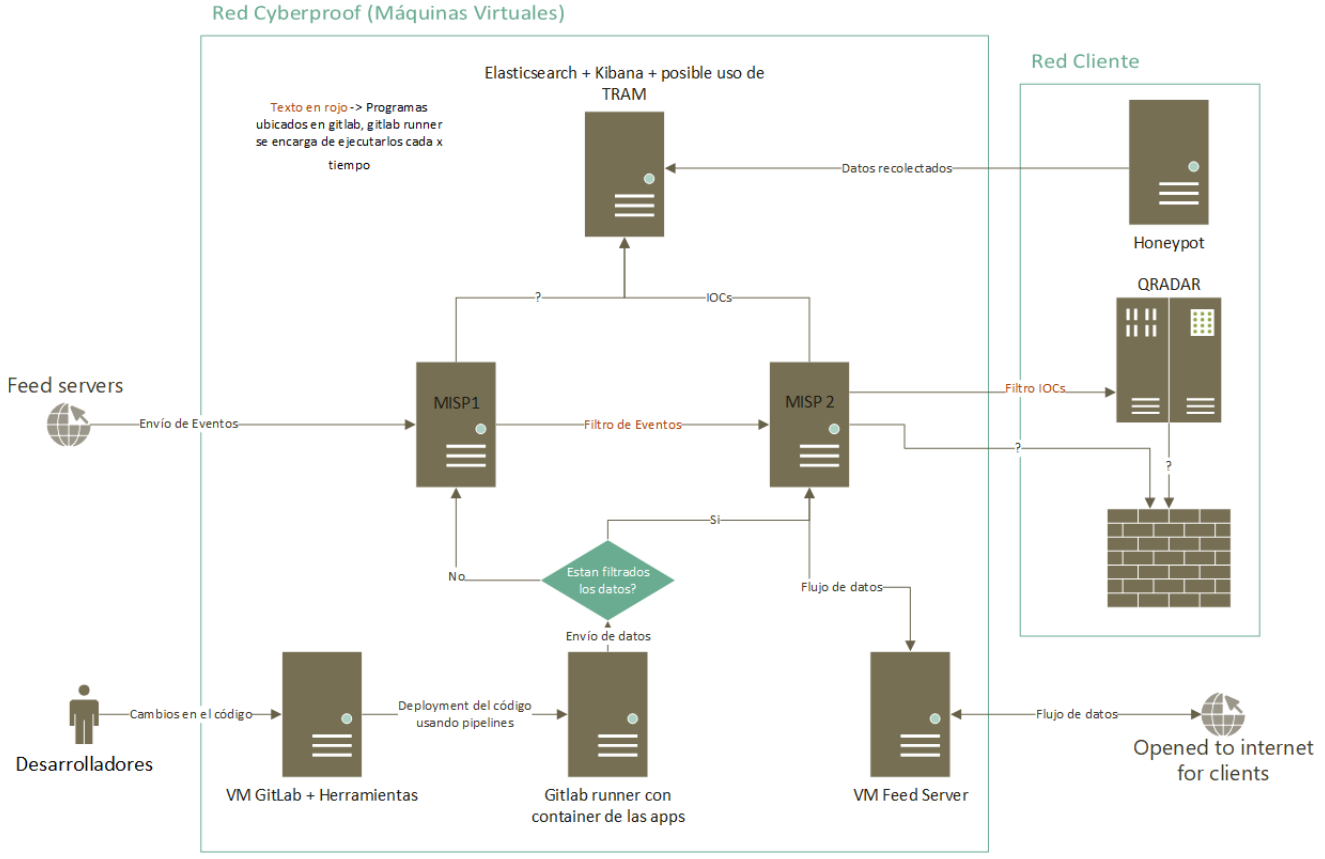

Fig. 3: Futuros avances. Aparición del ElasticSearch y del Honeypot