UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

Programa de Apoyo al Egreso de Profesionales en Actividad

TÍTULO: Dexter: Gestión automatizada de análisis forense en incidentes de seguridad

AUTOR: Juan Francisco Sirimarco

DIRECTOR ACADÉMICO: Paula Venosa

DIRECTOR PROFESIONAL: Magali Caruso

CARRERA: Licenciatura en Informática

Resumen

En la presente tesina, se desarrolla una introducción a un equipo modelo de respuesta ante incidentes, teniendo en cuenta las recomendaciones de distintos frameworks como NIST y FIRST. Se abordan los distintos servicios que un equipo de respuesta ante incidentes deberá considerar desde sus inicios.

Se presenta al equipo de Respuesta ante Incidentes de Mercado Libre (del cual formo parte actualmente) y al proceso de gestión de incidentes, describiendo la clasificación, el registro de incidentes, las métricas recomendadas y los distintos procedimientos de respuesta implementados.

Con el objetivo de reducir el tiempo de respuesta y análisis de un incidente de seguridad, se presenta y desarrolla una herramienta para automatizar y gestionar los análisis forense llamada "Dexter". La herramienta centraliza la información necesaria para el análisis forense de un incidente. Se describen y detalla la arquitectura, los distintos flujos para su uso, los desafíos encontrados en el camino del desarrollo y finalmente las recomendaciones para implementar Dexter en cualquier otro equipo de incidentes.

Palabras Clave

Análisis Forense, Automatización, Respuesta ante Incidentes, Gestión de incidentes, Enriquecimiento y centralizador de logs.

Trabajos Realizados

- *Revisión bibliográfica de Frameworks de gestión de incidentes.*
- *Presentación de los distintos procesos de gestión de incidentes y del equipo de Mercado Libre.*
- *Desarrollo de la herramienta Dexter para automatizar análisis forenses de incidentes.*

Conclusiones

- Gracias al desarrollo de la herramienta Dexter se lograron solucionar distintas problemáticas recurrentes al realizar un análisis forense, lo que permite reducir tiempos de respuesta y análisis en un incidente.
- Gracias a su escalabilidad, Dexter también facilitó la integración de nuevas fuentes de información y así centralizar los análisis en un mismo caso.
- Por último e importante a destacar es la trazabilidad y la posibilidad que brinda la herramienta de preservar evidencias forenses de un incidente ya analizado.

Trabajos Futuros

- *Como trabajo a futuro se propone sumar múltiples fuentes de información permitiendo ampliar y automatizar el análisis de los distintos escenarios de incidentes de seguridad.*
- *Para lograr esto, se propone también la migración hacia una arquitectura más escalable, solucionando de esta manera la problemática planteada de rendimiento.*
- *Finalmente, otra propuesta de trabajo a futuro, consiste en realizar las modificaciones necesarias para publicar Dexter como herramienta Open Source y que sea útil para cualquier otro equipo de respuesta ante inciden-*

Fecha de la presentación: **Agosto de 2021**

ÍNDICE GENERAL

1. Introducción

1.1 Motivación

La Seguridad Informática juega un rol fundamental en los últimos años debido al incremento constante de los ciberdelitos. Esto se ve reflejado constantemente en nuevos desafíos para toda empresa que necesita proteger sus datos e infraestructura. Debido a este incremento de ataques e incidentes, es necesario mejorar los tiempos de respuesta ante un incidente, para poder reducir el impacto que podrían causar.

Cuando se detecta un evento de seguridad, como puede ser una vulnerabilidad, un aumento de tráfico en la infraestructura, una alerta de Antivirus, etc; es necesario rápidamente contener el evento para reducir su impacto. Posterior a ello, se debe hacer un análisis forense para entender la superficie total del ataque, como por ejemplo, qué datos se comprometieron y qué acciones de mejora continua se podrían realizar para que no vuelva a suceder el mismo.

Durante mi experiencia realizando distintos análisis forenses de incidentes, nos encontramos junto al equipo con varias dificultades y problemáticas de distinta índole, como las siguientes:

- Números muy altos de usuarios y aplicaciones, por lo que el volumen de logs de tráfico a analizar es inmenso. Por ejemplo, en eventos relacionados a una vulnerabilidad web se podrían tardar días en conocer si se comprometieron datos.
- Debido al gran volumen de logs, las herramientas utilizadas para realizar consultas, frecuentemente devuelven múltiples errores. Por ejemplo, para casos de logs de tráfico utilizamos el servicio de Amazon llamado Athena[3].
- Varios de los incidentes ocurridos requieren consultar información y logs que se encuentran distribuidos en múltiples APIs o herramientas. Por lo tanto, se desperdicia tiempo en buscar y entender la API correcta para el evento ocurrido.

Teniendo en cuenta lo listado anteriormente, como equipo nos pareció necesario crear una herramienta que logre solucionar las dificultades experimentadas a la hora de trabajar con un incidente de seguridad.

1.2 Objetivos

A lo largo de la tesina se podrán identificar dos grandes conceptos, la Gestión de Incidentes en equipo y el desarrollo de una herramienta llamada Dexter.

En cuanto a Gestión de Incidentes se proponen los siguientes objetivos:

- Desarrollar una introducción a un equipo modelo de respuesta ante incidentes, describiendo la estructura y los distintos casos de uso a los que se da respuesta.
- Describir el equipo de Respuesta ante incidentes de Mercado Libre, su estructura y roles.
- Describir las distintas métricas, definiciones y procedimientos asociados al equipo.

Como objetivo general de la tesina, se propone el desarrollo de una herramienta que logre automatizar, enriquecer y gestionar el análisis forense de un incidente de seguridad informática ya sea ocurrido o en curso. De esta forma se logrará tener una rápida visión del incidente y así poder tomar una acción de contención rápida reduciendo el impacto del incidente.

Como objetivos específicos se plantean:

- Desarrollar la problemática recurrente encontrada en los análisis forenses.
- Analizar soluciones de análisis forenses existentes.
- Desarrollar una herramienta que permita resolver la problemática planteada.
- Aplicar un modelado de amenazas a la herramienta desarrollada.
- Realizar pruebas de funcionamiento de la herramienta en producción.
- Analizar resultados, ventajas/desventajas y futuras mejoras de la herramienta desarrollada.

1.3 Estructura de la tesina

La tesina se estructura de la siguiente forma:

En el capítulo 1, se resumen los objetivos y la motivación principal que dieron curso al desarrollo de la tesina.

En el capítulo 2, se realiza una introducción a un modelo de equipo de respuesta ante incidentes, basado en las recomendaciones de frameworks destacados y reconocidos por distintos equipos de seguridad. Además se detallan sugerencias relacionadas a las distintas fases de las que se compone un incidente de seguridad.

En el capítulo 3, se desarrolla el inicio y evolución del equipo de respuesta ante incidentes en Mercado Libre y todas las definiciones e implementaciones que sustentan a la gestión de Incidentes en la empresa.

En el capítulo 4, se presenta Dexter, una herramienta desarrollada con el objetivo de automatizar los análisis forenses en distintos escenarios de incidentes. En este capítulo se describe la herramienta desde su desarrollo, hasta su puesta en producción.

Finalmente en el capítulo 5, se describen las conclusiones finales de la tesina, tanto desde la Gestión de Incidentes, como del desarrollo de Dexter. Se realizan distintas recomendaciones para aplicar en otros equipos de respuesta ante incidentes. Se brindan distintas sugerencias para facilitar el desarrollo de una herramienta como Dexter en cualquier organización.

Finalizando el capítulo, se proponen varios puntos como trabajos a futuro.

2. Respuesta ante incidentes basado en frameworks

2.1 Introducción

En el presente capítulo vamos a introducir al armado de un equipo de respuesta ante incidentes basado en distintos framework como FIRST [1] Y NIST [2].

En la sección 2.2 se detallan cuáles son los servicios y objetivos que un equipo de respuesta ante incidentes deberá cumplir. En la sección 2.3 se describe cuales son los modelos posibles de equipo y cuales son los roles necesarios para responder de forma eficiente según ambos frameworks. En la sección 2.4 se detalla el principal servicio y misión al que un equipo de incidentes deberá responder, la gestión de incidentes de seguridad.

Luego en el apartado 2.5 se detalla cada fase que compone un incidente de seguridad, en cada fase se darán recomendaciones a tener en cuenta, como herramientas, tipos de alertas y escenarios posibles. En la sección 2.6 se recomiendan distintas métricas de incidentes necesarias para la maduración y crecimiento del equipo. Finalmente, en la sección 2.7 se describen y ejemplifican los posibles impactos a tener en cuenta en un incidente para su priorización.

2.2 Servicios de un equipo de respuesta ante incidentes

La gestión de incidentes es un conjunto ordenado de acciones que intenta prevenir que un incidente ocurra, y en caso que ocurra, intenta minimizar el impacto lo más rápido posible restaurando la operación de inmediato.

Dentro de un equipo de respuesta ante incidentes el FIRST define distintos servicios y funciones, dichos servicios pueden ser categorizados globalmente dentro de las siguientes áreas:

Figura 2.1. Servicios de un equipo de respuesta ante incidentes [1]

Gestión de eventos y alertas de seguridad: La gestión de eventos de seguridad tiene como objetivo identificar los incidentes de seguridad basándose en la correlación y el análisis de los eventos de una amplia variedad de eventos y fuentes de datos.
Los principales objetivos de esta área son:

- Monitoreo y detección
- Análisis de eventos

Gestión de Incidentes de seguridad: Se describe esta área y sus objetivos en la sección 2.4. Profundizaremos en esta área dado que es donde más puedo aportar desde mi experiencia.

Concientización de seguridad: La concientización comprende la capacidad de identificar, procesar, comprender y comunicar los elementos críticos de lo que está sucediendo dentro y alrededor del área de responsabilidad del equipo de respuesta ante incidentes.
Los principales objetivos de esta área son:

- Adquisición de datos
- Análisis y síntesis
- Comunicación

Gestión de vulnerabilidades: La gestión de vulnerabilidades incluye servicios relacionados con el descubrimiento, análisis y manejo de vulnerabilidades de seguridad nuevas o ya reportadas.También incluye servicios relacionados con la detección y respuesta a vulnerabilidades conocidas para evitar que sean explotadas.
Los principales objetivos de esta área son:

- Descubrimiento / investigación de vulnerabilidades

- Reporte de vulnerabilidades
- Análisis de vulnerabilidades
- Coordinación de vulnerabilidades
- Divulgación de vulnerabilidades
- Respuesta a la vulnerabilidades

Transferencia de conocimientos: esta área tiene como objetivo transferir todos los conocimientos que un equipo de respuesta ante incidentes tiene. Por ejemplo, identificar amenazas, tendencias y riesgos y así crear mejores prácticas operativas para ayudar a las organizaciones a detectar, prevenir y responder a incidentes de seguridad.
Los principales objetivos de esta área son:

- Entrenamiento y educación
- Ejercicios
- Asesoramiento técnico y asesoramiento de políticas

2.3 Modelos de equipo

Las estructuras posibles de un equipo de respuesta ante incidentes según el NIST [2] podrían ser:

- **Equipo de respuesta ante incidentes centralizado**: consiste en un solo equipo de respuesta ante incidentes que se encarga de todos los incidentes de la organización. Este modelo es efectivo para una organización pequeña que está centralizada en una misma región físicamente.

- **Equipo de respuesta ante incidentes Distribuido:** en este caso, la organización cuenta con varios equipos de respuesta ante incidentes, y cada equipo es responsable de un segmento físico o lógico de la organización. Este modelo es efectivo para grandes organizaciones que tienen sus recursos distribuidos en distintas regiones. Sin embargo, el equipo deberá ser parte de una sola entidad coordinada para que el proceso de respuesta ante incidentes sea consistente en toda la organización y la información pueda ser compartida entre los distintos sub equipos.

- **Equipo coordinado:** en este caso, se trata de un equipo de respuesta ante incidentes que brinda asesoramiento a otros equipos sin tener autoridad sobre estos. Este modelo puede ser pensado como un CSIRT[1] para CSIRTs .

Además los equipos de respuesta ante incidentes se clasifican en 3 modelos según el personal que los compone:

[1] CSIRT: Computer Security Incident Response Team

- **Empleados:** cuando la organización realiza todo el trabajo de respuesta ante incidentes.

- **Parcialmente tercerizado:** cuando la organización subcontrata una parte del equipo de respuesta ante incidentes

- **Completamente tercerizado:** cuando la organización subcontrata todo el trabajo de respuesta ante incidentes. Este modelo es utilizado cuando la organización no tiene suficientes empleados calificados. Es obligatorio en este caso, que la organización cuente con empleados supervisando el trabajo subcontratado.

El NIST [2] recomienda contar con los siguientes roles dentro del equipo de respuesta ante incidentes:

Manager y líder del equipo:

Deberá existir una persona encargada completamente del equipo de respuesta ante incidentes. La mayoría de los modelos generalmente cuentan con un Manager en el equipo y una o más personas que asumen su autoridad en caso de su ausencia.

El manager lleva a cabo distintas tareas de gestión, como actuar de enlace con otros equipos, managers o directores, manejar situaciones de crisis y asegurarse que el equipo cuenta con el personal, los recursos y las habilidades necesarias para contener un incidente.

Los managers deberán contar con buenas habilidades técnicas y excelentes habilidades de comunicación como para poder comunicar a una amplia audiencia un incidente en curso. Son completamente responsables de que las actividades de respuesta ante incidentes se lleven a cabo de forma correcta.

Líder técnico:

Algunos equipos pueden contar con un líder técnico que tenga fuertes habilidades técnicas y experiencia en respuesta ante incidentes y que asuma la responsabilidad por la calidad técnica del equipo.

Miembros del equipo:

Los miembros del equipo de respuesta ante incidentes deberán tener excelentes habilidades técnicas, como la de administrador de sistemas, administrador de redes, programación, soporte técnico y análisis forense.

Además, será fundamental que cada miembro del equipo tenga buenas habilidades para resolver problemas y cuente con pensamiento crítico.

Podría ser de ayuda contar con algunos miembros del equipo especializados en un área específica, como por ejemplo especialista en redes, análisis de malware o análisis forense.

Por último, es buena práctica sumar temporalmente al equipo a especialistas técnicos que no sean parte del equipo.

2.4 Gestión de Incidentes de seguridad

Esta área es el núcleo del equipo de respuesta ante incidentes por los servicios y funciones que cumple durante un ataque o incidente.

Ante un posible incidente reportado, el equipo deberá no solo recolectar y evaluar los reportes de seguridad recibidos, sino también analizar la información relevante y realizar un análisis técnico detallado del incidente y de los dispositivos utilizados.

Luego del análisis y confirmación de un incidente en curso, el equipo deberá actuar coordinadamente con los equipos involucrados tanto internamente como externamente, para seguir y aplicar todas las acciones mitigantes y luego recuperarse lo antes posible del incidente de seguridad ocurrido. Por último, se deberá evaluar la posible ejecución de un plan de escalamiento.

Según el Framework del FIRST, el área de gestión de incidentes deberá responder a los siguientes servicios *[1]*:

- Análisis de reportes de incidentes de seguridad
- Análisis de incidentes de seguridad.
- Análisis de evidencia forense.
- Mitigación y recuperación.
- Coordinación de incidentes de seguridad.
- Soporte en gestión de crisis.

Análisis de reportes de incidentes de seguridad: Este servicio tiene como objetivo recibir , procesar y clasificar los reportes de eventos de seguridad recibidos que pueden llegar principalmente desde el equipo de Detección de eventos y alertas de Seguridad Informática, desde un equipo interno no necesariamente relacionado a seguridad, como por ejemplo un equipo de desarrollo, un equipo de arquitectura o infraestructura, etc o de terceras partes reportando la detección de una anomalía.
Las principales funciones del servicio son:

- Recibir informes de incidentes de seguridad de la información
- Triage y procesamiento de incidentes de seguridad de la información

Análisis de incidentes de seguridad: Este servicio tiene como objetivo analizar y poder confirmar un incidente de seguridad junto a su potencial impacto, así como también poder identificar las vulnerabilidades, o debilidades que permitan un ataque o compromiso exitoso. Cuando el equipo valida que un incidente está ocurriendo, se deberá hacer un análisis para determinar en primera instancia el alcance del incidente, como por ejemplo cuáles son los recursos afectados , y que o quienes originaron el incidente , así como también cuales son

los métodos o vulnerabilidades que se están atacando. El resultado del alcance del incidente en conjunto con otras variables, se verá reflejado en el impacto del incidente.
Las principales funciones del servicio son:

- Coordinación de análisis.
- Análisis de las principales causas del incidente
- Recolección de información.
- Clasificación, priorización y categorización del incidente.
- Correlación con otros incidentes.

Análisis de evidencia forense: Este servicio tiene como objetivo analizar la evidencia relacionada a un incidente ya confirmado, considerando a su vez que es necesario preservar la evidencia forense. Este servicio está relacionado con entender las capacidades e intenciones de la evidencia recolectada, como un malware, un exploit, una copia de un disco rígido, logs, documentos,etc, así como también su propagación, detección, mitigación y neutralización del mismo. Aplica tanto a hardware, software y firmware. Además es importante considerar que cualquier dispositivo o evidencia debe conservarse y recopilarse sin ninguna modificación, manteniéndose de forma aislada. Algunas de las funciones a implementar de este servicio son:

- Análisis de medios.
- Ingeniería inversa.
- Análisis dinámico o tiempo de ejecución.
- Análisis comparativo.

Mitigación y recuperación: Este servicio tiene como objetivo contener el incidente lo que más se pueda para limitar los servicios, usuarios y dispositivos afectados, reducir las pérdidas y recuperarse del daño lo antes posible, evitar futuros ataques y pérdidas económicas eliminando vulnerabilidades y debilidades generales expuestas.
Las principales funciones de este servicio son:

- Establecer un plan de respuesta.
- Implementar medidas de contención.
- Restauración de sistemas.
- Soporte de otras entidades de seguridad para la contención.

Coordinación de incidentes de seguridad: Este servicio tiene como objetivo garantizar una distribución precisa de la información, mantener el flujo de la información y realizar un seguimiento del estado de las tareas proporcionadas a cada entidad o equipo que esté participando de la respuesta del incidente, midiendo y gestionando las posibles desviaciones causadas por retrasos.
Las principales funciones de este servicio son:

- Comunicación
- Distribución de notificaciones.
- Distribución de información relevante.
- Coordinación de actividades.
- Reportes
- Interacción con medios de comunicación.

Soporte en gestión de crisis: Este servicio tiene como objetivo obtener contactos de otros equipos especialistas que puedan ayudar a mitigar el incidente en un estado de crisis.

Las principales funciones del servicio son:

- Distribución de la información a los constituyentes.
- Informes de status al equipo de gestión de crisis.
- Comunicación pública de decisiones estratégicas.

2.5 Fases de un incidente

Un incidente consta de diferentes fases y aunque todas son necesarias algunas pueden estar incluidas como parte de otras o tratarse de manera simultánea. El NIST define las siguientes posibles etapas en un incidente [2]

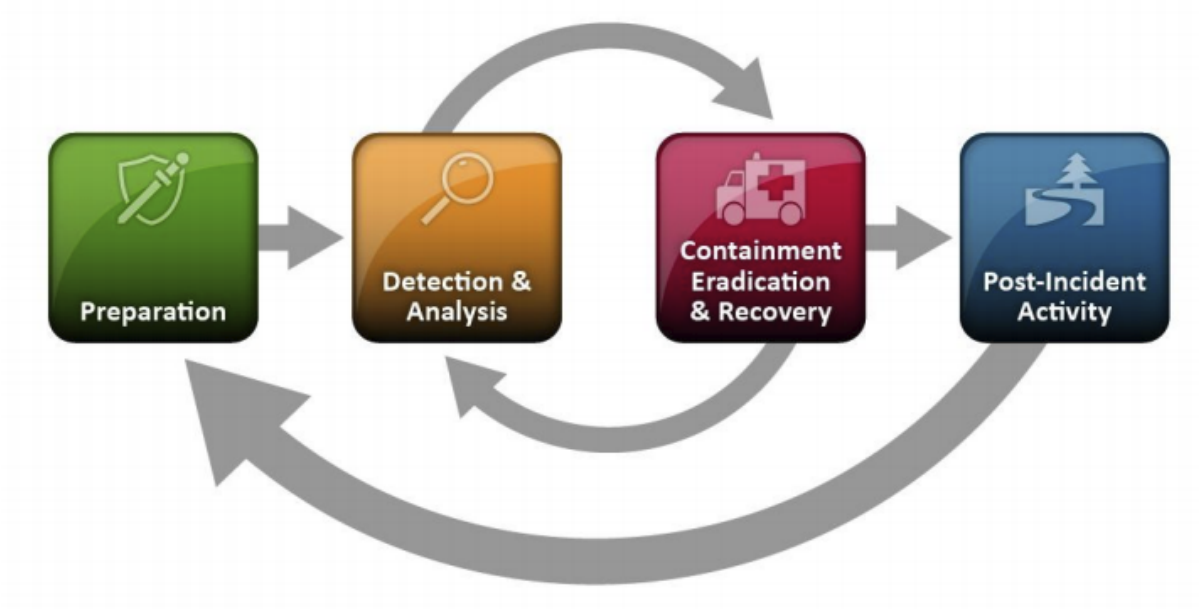

Figura 2.2. Ciclo de vida de un incidente

2.5.1 Preparación

La organización en esta fase será responsable de contar con distintos recursos y herramientas para prepararse efectivamente para responder ante un incidente. Estos recursos y herramientas podrían ser:

- **Información de contacto y guardias**: Correos, número de teléfonos, etc de miembros del equipo dentro de la organización, así como también contactos necesarios fuera de la organización. Por otro lado, es indispensable disponer de la información de las guardias de los otros equipos por posibles escalamientos.

- **Contar con un sistema para dar seguimiento al incidente**: Este sistema será el encargado de tener un registro de todos los incidentes, con su tipo, estado, impacto , datos afectados, reporte ejecutivo,etc de cada incidente.
- **War Room:** Se trata de un espacio o reunión, ya sea presencial o no, para centralizar las comunicaciones y coordinar el incidente en curso.
- **Almacenamiento seguro:** Necesario para guardar y asegurar las evidencias y el material sensible.
- **Hardware y software:** Dentro de esta categoría, es necesario contar con workstation para realizar las copias y análisis forenses, software para analizar imágenes de disco, un medio extraíble para obtener la evidencia de los sistemas, imágenes limpias de un SO para restauración, etc.
- **Documentación:** Es necesario contar con documentación de los SO, protocolos, puertos, antivirus,diagramas de red y assets críticos, etc.

En esta fase, estar preparados para responder y mantener el número de incidentes bajos puede ser clave para la protección del negocio de la organización. Si los controles de seguridad son insuficientes el incremento va a ser notable, y el equipo de respuesta ante incidentes va a estar sobrecargado.

2.5.2 Detección y Análisis.

El objetivo de esta fase es tener la capacidad de identificar o detectar cualquier incidente que pueda sufrir un organismo o entidad y con el menor impacto posible, por lo que es importante contar con un monitoreo completo. Se tiene como premisa que no todos los eventos o alertas detectadas son posibles incidentes.

Algunos de los pasos a llevar a cabo en esta etapa son:

- Registrar y monitorear los eventos de las redes, sistemas y aplicaciones.
- Recolectar información situacional que permita detectar anomalías.
- Disponer de capacidades para descubrir incidentes y comunicar los mismos a los contactos apropiados.
- Recopilar y almacenar de forma segura todas las evidencias.
- Compartir información con otros equipos internos y externos de forma bidireccional para mejorar las capacidades de detección.

.

2.5.2.1 Alertas

Existen varios indicadores que nos dan señales que un incidente puede estar ocurriendo en el momento actual como por ejemplo un antivirus detectando que un host de nuestra red

está infectado, o logs aplicativos advirtiendonos de múltiples fallos de login, o un administrador de red detectando un aumento de tráfico anómalo, entre otros.

También existen los precursores que van a realizar detecciones de posibles incidentes a futuro por ejemplo un IDS advirtiendonos de un escaneo de puertos hacia un servidor, o una detección de escaneo de vulnerabilidades hacia un servidor web, o un tweet que amenaza a nuestra organización

Los indicadores y precursores pueden originarse de distintas fuentes. Las categorías más importantes de fuentes que un equipo de respuesta ante incidente debería considerar son :

- **Alertas de un IDS:** se definen como IDS a aquellos productos que están diseñados para identificar eventos maliciosos y recolectar la mayor información posible relacionada a cada evento detectado. A veces pueden causar varios eventos considerados como falsos positivos.

- **Alertas de antivirus, y antispam:** se diseñan antivirus y antispam para detectar código malicioso y prevenir la infección de un host. Algunas organizaciones cuentan con más de un antivirus de distintos vendedores para lograr una mayor cobertura. Por otra parte, el software antispam detecta cuando un usuario intenta descargar un correo que contiene phishing, malware, etc.

- **Alertas de chequeo de integridad de archivos:** se diseña este software para analizar la integridad de archivos mediante el checksum. Detectan posibles alteraciones realizadas en un archivo que podría contener código malicioso.

- **Servicio de monitoreo tercerizado:** en este caso, algunas organizaciones tercerizan el monitoreo de servicios o recursos públicos como un DNS, un servidor FTP, o un sitio web.

- **Logs aplicativos, de servicio y de sistema operativo:** una de las fuentes más importantes para detectar distintos tipos de incidentes. Estos logs muestran todo tipo de información detallada, por ejemplo con estos logs podríamos saber si alguien accedió a un sistema o servidor y todas las acciones que realizó. Se debe hacer un chequeo periodico que estos logs se estén almacenando correctamente.

- **Logs de dispositivo de red:** logs de tráfico como por ejemplo de firewalls y de routers que pueden ser muy útiles como precursores o indicadores de incidentes.

- **Información de nuevas vulnerabilidades:** mantenerse atentos a noticias de nuevas vulnerabilidades y exploits publicados podría prevenir varios incidentes. Existen CERTS y organizaciones que publican y actualizan constantemente los últimos hechos ocurridos.

- **Información de incidentes en otras organizaciones:** otra de las fuentes clave es mantenerse en contacto con directivos de seguridad de otras organizaciones, y compartir la información de ataques o incidentes ocurridos para prevenirlos en otras

organizaciones. Se suelen compartir Indicadores de Compromiso (IoC) de cada ataque en sitios, redes o por correo para detectar , bloquear y prevenir un incidente.

- **Personas dentro de la organización:** administradores de red, de Sistemas, o cualquier colaborador de la organización podría compartir reportes de posibles ataques, vulnerabilidades o eventos de seguridad. Es necesario validar cuidadosamente cada reporte recibido.

- **Personas fuera de la organización:** usuarios que encuentran una vulnerabilidad en el sistema y deciden contactar a la organización para enviar el reporte. Estos casos pueden ser críticos dado que el usuario podría publicar en redes la vulnerabilidad encontrada.

En algunos casos sucede que es fácilmente visible la presencia de un incidente en curso cuando el indicador es muy claro, como por ejemplo la sustitución de la portada de una página web conocida como “defacement”. En otras situaciones al detectar un indicador más general como el cambio de un archivo de configuración requiere un análisis completo del equipo de gestión de respuesta ante incidentes para entender si se está llevando a cabo un incidente de seguridad y tomar las acciones adecuadas para contenerlo.

Muchas veces ocurre que se detecta una de las alertas o indicadores nombrados, pero no está relacionado a ningún proceso de seguridad, un claro ejemplo puede ser que una aplicación deje de responder o la respuesta no sea la esperada y la causa sea un deploy de versión con código erróneo que se realizó en las últimas horas.

Algunas recomendaciones para realizar un adecuado análisis de incidentes son:
- Entender el funcionamiento normal de las redes, sistemas y aplicaciones, o conocer qué experto podría ayudar a analizar en caso necesario.
- Crear una política de almacenamiento de logs aplicativos, de firewall, etc.
- Poder correlacionar logs de distintas fuentes e incidentes pasados.
- Mantener una base de conocimientos que contenga procesos y procedimientos de cómo actuar ante cada incidente.
- Filtrar los indicadores y precursores críticos para poder enfocarse en ellos.
- Buscar asistencia en otros equipos u otros CERTs en caso de necesitarla.

2.5.2.2 Documentación del incidente

Cuando el equipo de respuesta ante incidentes detecta que un incidente está ocurriendo, el Incident Handler debe empezar a registrar todo lo sucedido. Documentar cada paso sucedido y acción que se tome con su fecha correspondiente es necesario tanto para tener una amplia visión de lo ocurrido como para tener la evidencia necesaria en caso de que el incidente incluya abrir una causa en la justicia. Por ello, es necesaria una persona que documente toda la evidencia mientras que otra persona o equipo en paralelo contenga técnicamente el incidente.

Por otro lado, es necesario que el equipo cuente con un sistema de seguimiento de incidentes que logre mantener una base de datos con cada incidente actualizado. De cada incidente es necesario documentar:

- El estado actual de cada incidente (Abierto, en proceso, mitigado, cerrado,etc).
- Un resumen ejecutivo del incidente.
- Indicadores relacionados al incidente.
- Todas las acciones realizadas.
- Evidencias.
- Recursos e impacto que tuvo el incidente.
- Información de contacto de las partes involucradas.
- Tareas de remediación o mejora continua.

2.5.3 Contención, erradicación y recuperación

La etapa de mitigación o contención del incidente es una de las más importantes para reducir el impacto del incidente en curso. Para ello se debe elegir la estrategia que mejor se adecue que va a depender directamente del tipo de incidente con el que estemos tratando, por ejemplo la mitigación de un abuso de permisos va a ser totalmente diferente a contener un ataque de Denegación de Servicio.
Para lograr una mejor eficiencia en la contención, se recomienda el uso de procedimientos predefinidos para cada tipo de escenario posible. En cada procedimiento se debe detallar:

- Cómo preservar las evidencias.
- El daño potencial hacia los recursos.
- La disponibilidad de los servicios.
- Recursos y tiempo necesarios para implementar la estrategia de contención

Una vez contenido el incidente, la siguiente fase es la **erradicación y recuperación** del mismo, que puede constar de distintas tareas como:

- Identificar y restaurar todos los hosts afectados.
- Eliminar componentes restantes de Malware.
- Deshabilitar o resetear cuentas comprometidas.
- Parchear vulnerabilidades abiertas.
- Recuperar un host o servidor desde un Backup.
- Mejorar la seguridad perimetral, agregando nuevas ACL o rules.
- Mejorar sistema de monitoreo y logeo para incidentes similares.

La fase de erradicación y/o recuperación es específica de cada escenario que se presenta, a veces puede tratarse de una tarea, o a veces puede que con la contención sola el incidente ya pueda cerrarse.

2.5.4 Actividades post incidente

Como última instancia es clave generar un espacio de **lecciones aprendidas** días después que el incidente haya sido cerrado. En este espacio se tratarán todas las oportunidades de mejoras encontradas para evitar que un incidente de características similares vuelva a ocurrir.

Algunos de los temas a tratar y documentar en esta fase son:

- Describir cronológicamente el incidente y las acciones realizadas.
- Evaluar cómo reaccionó tanto el equipo de respuesta ante incidentes como todos los involucrados.
- Detallar qué herramientas, y/o acciones hubiesen aplicado para detectar, analizar y contener más rápido el incidente.
- Definir nuevas alertas y monitores que se van a generar para detectar el caso ocurrido.
- Detallar cómo el equipo responderá en futuros casos.

En esta reunión post incidente, es necesario involucrar a todas las personas necesarias tanto que participaron del incidente como personas que podrían cooperar en posibles acciones que surjan como lecciones aprendidas. La presencia de uno o dos moderadores que sigan la reunión y hagan una introducción de lo ocurrido va a evitar confusiones a futuro. Es necesario que todo lo charlado quede documentado como futuros pasos y acciones el equipo correspondiente, estos documentos son utiles tambien a futuro como casos de uso similares para nuevas personas en el equipo de respuesta ante incidentes.

2.6 Métricas recomendadas de Incidentes

Recolectar y generar métricas a partir de toda la información de los incidentes ocurridos en un tiempo prolongado es una buena práctica que trae diversos beneficios en un equipo de respuesta ante incidentes.

Por ejemplo, las métricas de tiempo de respuesta o contención en un incidente podrían utilizarse para justificar la necesidad de contratar nuevo personal para el equipo y para medir la eficiencia del equipo actual. Las métricas de los tipos de incidentes o vectores de ataque recibidos durante un mes, podrían indicar una posible debilidad en la organización que haya que mejorar.
Además, estos datos podrían compartirse al proceso de evaluación de riesgos para implementar controles adicionales.

Algunas de las métricas de incidentes recomendadas por el NIST [2] para construir son:

- **Cantidad de incidentes gestionados por categoría:** Para medir tanto la cantidad de trabajo realizado por el equipo, como para entender donde hace falta mejorar controles. Esta métrica no necesariamente responde a la calidad con la que el equipo responde.
- **Tiempo por incidente:** Para cada incidente el tiempo se puede medir de distintas formas:
    - Tiempo total en que se trabajó sobre el incidente.
    - Tiempo en rangos por cada etapa del incidente.
    - En cuanto tiempo el equipo respondió al reporte inicial del incidente
- **Efectividad de respuesta de incidentes:** Analizar la efectividad y calidad luego de de resolver un incidente, nos puede brindar a largo plazo distintas métricas útiles para el equipo, por ello se recomienda identificar los siguientes puntos por cada incidente resuelto:
    - Identificar los precursores e indicadores de los incidentes detectados para determinar cuán efectivo fueron.
    - Determinar si el incidente tuvo impacto luego de ser detectado, para determinar la efectividad de respuesta.
    - Determinar los vectores de ataque predominantes.
    - Determinar si el incidente está relacionado a un incidente anterior.
    - Calcular el impacto monetario, teniendo en cuenta la información comprometida tanto de negocio como de usuarios.

2.7 Priorización de incidentes

Establecer prioridades en la respuesta de incidentes es uno de los puntos más importantes en el proceso de gestión.
Cuando un incidente ocurre es necesario saber si debemos priorizarlo ante otros incidentes en curso. Los incidentes no deben atenderse por orden de llegada en caso de tener un equipo limitado, de lo contrario se deben priorizar en función de los siguientes factores:

- **Impacto funcional del incidente:** en caso que el incidente ataque al pilar *disponibilidad*, es necesario calcular el impacto de negocio de qué sistemas y/o servidores están siendo afectados o podrían estarlo en caso que el incidente se materialice.

- **Impacto de la información del incidente:** en caso que el incidente ataque al pilar *confidencialidad*, es necesario que un Incident Handler pueda clasificar la información expuesta (véase la figura 3.10), no solo en cantidad sino qué tipo de información se podría filtrar en caso que el incidente se materialice. Por ejemplo, si es información personal de usuarios finales, que impacte en una regulación propia de un país o si es información propia de la organización, como información estratégica de negocio, o información de sueldos de empleados, etc

- **Recuperación del incidente:** dependiendo de los recursos afectados y del tipo de incidente, es necesario calcular el esfuerzo necesario para la recuperación de un incidente con sus pasos y requisitos necesarios para ello, en algunos casos como el

de filtración de información no es necesario desperdiciar tiempo dedicado de recursos de la organización, al no poder recuperar la información previamente filtrada.

2.7.1 Criticidad de incidentes según su impacto.

Podríamos priorizar los incidentes, clasificándolos por criticidad teniendo en cuenta los distintos niveles de impacto nombrados. Si bien el impacto y la clasificación va a depender de cada organización, un modelo de ejemplo podría ser el siguiente:

Crítico

Son incidentes de seguridad que tienen al menos una de las siguientes características:

- El caso ha sido expuesto públicamente por un tercero o existe alta probabilidad que pueden llegar a ser públicos.
- Se trata de una fuga masiva de información de datos de tarjeta con alcance PCI[2].
- Se trata de una fuga de información masiva de datos PII[3] en donde los datos son principalmente Sensibles o Transaccionales.
- Se trata de un incidente de fuga de información que tiene impacto en los estados contables o afecta directamente a las Inversiones e Inversionistas de la Institución.
- Se considera que por sus características tiene definitivamente una sanción regulatoria o legal.
- Se trata de un incidente que tiene una afectación superior sobre la continuidad del negocio de cara al usuario.
- El incidente tiene una afectación mayor sobre las métricas core de la Institución.

Alto

Son incidentes de seguridad que tienen al menos una de las siguientes características:

- Se trata de un incidente que afecta en mayor medida a la continuidad del negocio de cara al usuario.
- Tiene una alta probabilidad de tener un impacto negativo en la imagen de la Institución si se hiciera público.
- Se trata de una fuga de información masiva de datos PII en donde los datos son

[2] P*CI: Es un estándar de seguridad de datos para la industria de tarjetas de pago.*
[3] *PII: La información de identificación personal es cualquier dato que podría identificar potencialmente a un individuo específico.*

principalmente limitados.

- Se trata de una fuga de información no tipificada como PII ni financiera pero su cálculo de impacto dió Alto durante el la evaluación.
- Se considera que por sus características quizás pueda tener una sanción regulatoria o legal.
- El incidente tiene una afectación mayor sobre las métricas core de la Institución.

Medio

Son incidentes de seguridad que tienen al menos una de las siguientes características:

- Se trata de un incidente que tiene una afectación menor en la continuidad del negocio de cara al usuario.
- Se trata de un incidente de fuga de datos PII puntual de datos limitados o transaccionales.
- Se trata de una fuga de información datos de tarjeta con alcance PCI que se encuentra en un rango de 10 a 100 registros.
- El incidente tiene una afectación menor sobre las métricas core de la Institución.
- Tiene menor probabilidad de tener un impacto negativo en la imagen de la Institución si se hiciera público.
- Se considera que por sus características no tiene impacto sobre sanciones regulatorias pero puede tener impactos de carácter legal.

Bajo

Son incidentes de seguridad que tienen al menos una de las siguientes características:

- El incidente no tiene una afectación sobre las métricas core de la Institución.
- Se trata de un incidente que casi no tiene afectación sobre la continuidad del negocio de cara al usuario.
- Se trata de un incidente de fuga de datos PII puntual de datos limitados.
- Se considera que por sus características no tiene asociada una sanción regulatoria o legal.
- El incidente no es de carácter público ni tiene afectación sobre la imagen de la Institución.

2.7.2 Escalamiento de incidentes

El escalamiento de incidentes define los lineamientos a seguir para notificar a los equipos y/o roles intervinientes, a fin de activar un plan de comunicación correspondiente y definición de acciones que disminuyan el impacto que pudiera ocasionar un incidente de seguridad informática.

Cualquier incidente tratado por el equipo de Respuesta ante Incidentes podría ser escalado, sin importar el tipo y/o impacto del mismo.

Al escalar un incidente, un comité interno deberá analizar la situación ocurrida a un nivel táctico, y validará si el caso requiere una comunicación externa debido a su impacto regulatorio, legal o sobre la imagen de una institución.

3. Gestión de incidentes *en Mercado Libre*

3.1 Introducción

En el presente capítulo se detalla cómo es la gestión de incidentes en Mercado Libre, cómo está compuesto el equipo y distintas definiciones que se realizaron en el mismo para su crecimiento.

En la sección 3.2 se describe como el equipo está formado y distribuido, cuales son los distintos roles y cómo interactúan entre ellos.

Luego en la sección 3.3 y 3.4 se detalla como es la taxonomía que adoptamos actualmente para registrar los distintos tipos de incidentes y los vectores de ataque posibles relacionados.

En la sección 3.5 se presenta la herramienta de registro de incidentes desarrollada internamente, en la cual participé como analista de calidad de Software (QA).

En el apartado 3.6 se muestran las distintas métricas definidas en el equipo, clasificadas en métricas de Gestión, Técnicas y Ejecutivas.

En la sección 3.7 se describen los protocolos y procedimientos que definimos y documentamos en el equipo para responder ante distintos escenarios recurrentes.

Por último, en la sección 3.8 se desarrollan distintas recomendaciones a la hora de gestionar incidentes o iniciar un equipo de respuesta ante incidentes.

3.2 Estructura del equipo y roles

Mercado Libre es una empresa a nivel regional que se encuentra distribuida en un total de 18 países, actualmente cuenta con aproximadamente 20.000 empleados, es una empresa de gran capacidad que además cuenta con infraestructura dividida en varios países.

A finales del año 2018, se formalizó el equipo de respuesta ante incidentes, con el objetivo de contar con un equipo exclusivamente dedicado a gestionar y responder los incidentes que atentaban contra Mercado Libre y sus distintas unidades de negocio. El equipo inició con 2 personas y en Abril del año 2019 me incorporé al equipo.

El equipo tuvo un gran crecimiento y evolución en el transcurso de los últimos 2 años, a continuación se detalla el modelo y estructura del equipo según lo visto en la sección 2.4.

Actualmente, en el equipo de respuesta ante incidentes tomamos un **modelo distribuido** donde contamos con subequipos distribuidos en distintos países, como Argentina, Chile, Brasil, Colombia y México, y teniendo el equipo coordinador central en Argentina.

Respecto al personal del que se compone el equipo de respuesta ante incidentes, el equipo está completamente **compuesto por empleados pertenecientes a la organización**, es decir no hay ningún miembro del equipo tercerizado.

En cuanto a la disponibilidad del equipo de respuesta ante incidentes, es de 24/7 donde existe una guardia rotativa para incidentes que puedan ocurrir fuera de horario laboral y además la guardia está dividida en los distintos equipos de cada país.

El equipo de respuesta ante incidentes se compone actualmente de los siguientes roles :

- **Manager**: Referente ante incidentes a nivel directivo. Participación activa en los planes de escalamiento y comité de riesgos.

- **Project Leader**: Responsable del seguimiento y comunicación en cada incidente, encargado de que los miembros del equipo puedan tomar y responder los incidentes ocurridos, organizando a su vez las reuniones de enlace necesarias para el seguimiento de cada incidente. En conjunto al manager y líder técnico, forma nuevos sub equipos

- **Technical Leader**: Un referente que acompaña al equipo ante cualquier duda técnica que surja dentro de un incidente. Además vela por la calidad técnica del equipo.

- **Incident Handler**: Los miembros de este equipo serán los responsables de seguir un incidente desde que se alerta hasta que se cierra.

- **Forensic**: Los miembros de este equipo serán los encargados de realizar los análisis forenses de los distintos incidentes, para conocer en detalle si hubo un compromiso de datos.

- **Detector**: Los miembros de este equipo serán los encargados de analizar todas las alertas generadas de las distintas fuentes de la organización, que podrán ser escaladas a incidentes.

- **Threat Intelligence:** Los miembros de este equipo se encargan de investigar las amenazas externas dirigidas a la organización.

- **Antiphishing:** Equipo encargado de detectar y mitigar ataques de phishing que atentan contra Mercado Libre.

- **Threat Hunting:** Los miembros de este equipo se encargan de buscar amenazas proactivamente e internamente en la organización.

Luego existen los siguientes equipos que si bien tienen objetivos distintos a Incidentes, trabajan directamente y complementariamente con el equipo de respuesta ante incidentes dia a dia:

- **Riesgos**: Encargado de realizar la gestión dinámica de riesgos, y de velar por la maduración de los procesos de seguridad.

- **Security Awareness:** Equipo responsable de la capacitación y concientización de ciberseguridad para empleados internos y externos.

Estos roles dentro del equipo de respuesta ante incidentes, interactúan entre sí de distintas formas, en el siguiente gráfico se pueden visualizar los posibles flujos interactivos entre los distintos roles:

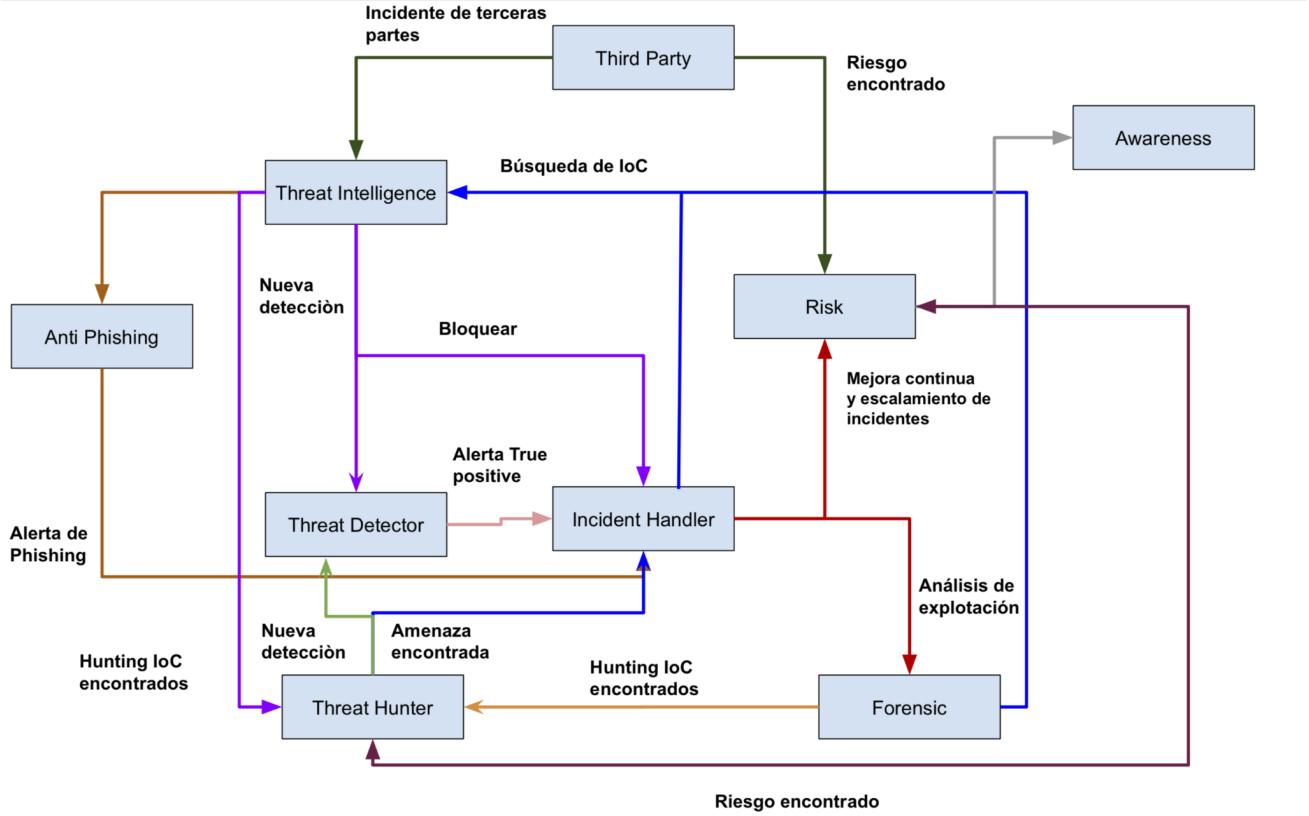

Figura 3.1 Posibles flujos - Equipo de incidentes en Mercado Libre

3.3 Clasificación de Incidentes

Ante la necesidad de poder estandarizar en Mercado Libre definimos una taxonomía de incidentes en base a las recomendaciones de ENISA [19] y del CERT de Estados Unidos “CISA” [20] , de forma que se ajuste a los eventos e incidentes más recibidos.

En Mercado Libre tomamos como definición que un evento se transforma en incidente, si se concreta exitosamente el incidente y existe un impacto directo hacia alguno de los 3 pilares de seguridad:

- Confidencialidad
- Integridad
- Disponibilidad

Veamos algunos ejemplos de eventos e incidentes según la definición:

Escenario 1: Un Malware es descargado desde un correo, solo si este logra ejecutarse exitosamente se categoriza en Incidente, caso contrario es un evento de Seguridad.

Escenario 2: Un ataque de denegación de servicio, se lo considera incidentes sólo si genera una caída total o parcial en una o varias aplicaciones.

Escenario 3: Una filtración de contraseñas o keys, se la considera incidente solo si se detecta la utilización de las cuentas filtradas.

A continuación se listan todas las categorías de incidentes posibles con sus respectivos subtipos y vectores de ataque:

Tipo de Incidente	Subtipo	Descripción	Vector de Ataque
Malicious Code	**Ransomware**	Software que se incluye o inserta intencionalmente en un sistema con un propósito dañino. Normalmente es necesaria la interacción del usuario para activar el código.	● Email ● Medios externos ● Uso inadecuado ● Web
	Skimming		
	Other Malware		
Information Gathering	**Scanning**	Esta categoría incluye cualquier actividad que busque identificar computadoras, puertos abiertos, protocolos, servicios o cualquier combinación para luego explotarlo o realizar un uso ilícito de la información obtenida.	● Otro
	Crawler		
	Scrapper		
Availability	**DoS**	Son ataques que, mediante el agotamiento de recursos, impiden parcial o totalmente el uso autorizado de redes, sistemas o aplicaciones.	● Desgaste
	DDoS		

		Si el origen del ataque se produce desde un único punto de origen es DoS y si se produce desde múltiples puntos de origen es DDoS.	
	Outage	Cuando la disponibilidad de un servicio se ve afectada por acciones locales (destrucción, etc.), por casos de fuerza mayor, fallas espontáneas o errores humanos, sin que haya malicia o negligencia grave	● Uso inadecuado ● Error humano
Vulnerable	**Web**	Vulnerabilidades web halladas (puede ser mediante herramientas automatizadas) teniendo de referencia el top ten de OWASP.	● Web ● Impersonalización ● Error de código ● Vulnerabilidades sin parche ● Mala configuración
	Infrastructure	Resolvers abiertos, impresoras legibles en todo el mundo, vulnerabilidades detectadas mediante herramientas automatizadas, firmas de virus no actualizadas, etc.	● Impersonalización ● Mala configuración ● Producto desactualizado
Intrusions	**Compromising an account**	Compromiso de la cuenta de un usuario o empleado.	● Desgaste ● Web
	Application Compromise	Compromiso de un sistema o una aplicación (servicio). Esto puede haber sido causado de forma remota por una vulnerabilidad nueva o conocida, pero también por un acceso local no autorizado. También incluye ser parte de una botnet.	
Fraud	**Abuse**	Implica el abuso de privilegios o de políticas de uso aceptable, es decir, cuando un individuo realiza un uso ilegítimo de los recursos de la organización a los que tiene acceso.	● Uso inadecuado
	CopyRight	Uso de una marca registrada o de dominios similares que puedan inducir a confusión respecto de la afiliación del titular de la marca con la promoción que se esté realizando.	● Otro
	Social Engineer	Todo engaño, soborno o amenaza que se realiza para recopilar información confidencial.	● Email ● Web ● Otro

	Checker	Checker de tarjetas.	• Desgaste
Data Breach	**Privacy**	Exposición de información de usuarios de la organización. La información puede ser de tipo PII, PCI o de ninguna de ambas clases. Puede haber ocurrido en la infraestructura de nuestra organización o en la de un tercero con una relación contractual	• Mala configuración • Error humano • Cadena de suministro • Otro
	Proprietary	Incidentes que exponen información del negocio o interna de nuestra organización. Por ejemplo: planes estratégicos.	
Other	**Other**	En caso que un incidente no pueda ser categorizado en ninguno de los tipos anteriores.	• Cualquier vector de Ataque

Figura 3.2: Clasificación de incidentes.

3.4 Vectores de ataque

El vector de ataque es el método utilizado para lograr un acceso no autorizado a un sistema. Los vectores de ataque permiten a los ciberdelincuentes explotar las vulnerabilidades del sistema para obtener acceso a datos confidenciales, información personal (PII), etc.

- **Web:** Un ataque ejecutado desde un sitio web o una aplicación basada en web.
    - Ej: Ataque Cross-Site Scripting usado para robar credenciales, o una redirección a un sitio que explota vulnerabilidades del navegador e instala malware.
- **Email:** Uno de los vectores más comunes, donde el ataque se concreta vía un archivo adjunto que termina ejecutando malware o un link hacia un sitio falsificado.
    - Ej: Explotar código disfrazado como un documento adjunto o un enlace a un sitio web malicioso en el cuerpo del mensaje de un correo electrónico.
- **Desgaste:** Un ataque que emplea métodos de fuerza bruta para comprometer, degradar o destruir sistemas, redes o servicios.
    - Ej: DoS destinado a impedir el acceso a una aplicación; un ataque de fuerza bruta contra un mecanismo de autenticación, como contraseñas o firmas digitales.

- **Medios externos:** Un ataque ejecutado desde un medio removible o dispositivo periférico.
    - Ej: Código malicioso que se propaga a un sistema desde una unidad flash infectada.
- **Impersonalización:** Un ataque que implica el reemplazo de contenidos o servicios legítimos con un sustituto malicioso.
    - Ej: Spoofing, Man in the Middle, rogue wireless access points, y SQL injections.
- **Uso inadecuado:** Cualquier incidente/evento que surja de la violación de una politica de uso aceptable de la organización de un usuario autorizado (excluyendo las categorías anteriores).
    - Ej: El usuario instala el software para compartir archivos que lleva a la pérdida de datos sensibles; o un usuario realiza actividades ilegales en un sistema.
- **Pérdida de Equipo:** La pérdida o el robo de un dispositivo informático o medios utilizados por la organización.
    - Ej: Una laptop o dispositivo móvil perdido.
- **Error humano:** Error humano que causa la caída de un sistema o servidor, o la filtración de información sensible.
    - Ej: Exposición de keys en repositorios Git Públicos.
- **Mala configuración:** Configuración de un programa que provoca un comportamiento no deseado y que afecta a la seguridad de un sistema. Los errores de configuración pueden darse en aplicaciones web, sistemas operativos, servidores.
    - Ej: Uso de cuentas predeterminadas, servicios o puertos abiertos innecesariamente, servidores sin contraseñas configuradas, etc.
- **Error de código:** Errores en el código de un programa que hace que el mismo produzca resultados incorrectos, inesperados, o que se bloquee. Pueden tener un impacto en la seguridad.
- **Vulnerabilidades sin parchear:** Errores de seguridad conocidos que no han sido parcheados.
- **Producto desactualizado:** Un Software, programa o sistema operativo desactualizado.
- **Ataque en cadena:** Ataques o fugas de información originadas a partir de terceras o cuartas partes de confianza.
- **Desconocida:** Causa del ataque no identificada.
- **Otro**: Un método de ataque que no encaja en ningún otro vector y que no es desconocido.

3.5 Sistema de Registro de Incidentes

Un sistema de registro de incidentes mantiene un listado actualizado de todos los incidentes y eventos ocurridos. Es necesario que las medianas y grandes organizaciones cuenten con un sistema de registro de incidentes ya sea propio o tercerizado por todos los beneficios que trae:

- Mantener un histórico de incidentes ocurridos.
- Tener un acceso centralizado a todos los incidentes.
- Tomar como modelo la resolución de incidentes ya ocurridos.
- Que los ejecutivos y otros equipos estén actualizados del estado de un incidente.
- Crear distintas métricas de incidentes.
- Correlacionar Incidentes.

En Mercado Libre contamos con un sistema de registro y seguimiento de incidentes desarrollado internamente. En el mismo registramos por cada incidente los siguientes campos:

- Nombre y Descripción del Caso
- Clasificación del Caso por Tipo, Subtipo y Vector de ataque.
- Estado del caso.
- Analista del equipo asignado.
- Label: Identificación de Evento e Incidente de Seguridad Informática.
- Impacto inicial.
- Si el caso es de público conocimiento o no.
- Si fue explotado el evento.
- Si hay evidencia para analizar o no.
- Si tuvo una monetización asociada.
- Cuál fue el atacante y la fuente de origen.
- Si hay IPs asociadas al caso.
- Pilar afectada por el caso (CID) - Confidencialidad, Integridad y Disponibilidad.
- Exposición de la información.
- Tecnología interna utilizada.
- Identificación de afectación de métricas organizacionales.
- Si hubo afectación del servicio y su duración en tiempo.
- Tipo y Cantidad de usuarios afectados.
- Sitio organizacional comprometido.
- Ambiente comprometido.
- Fecha y hora de ocurrencia.
- Fecha y hora de detección.
- Fecha y hora de contención.
- Fecha y hora de inicio y fin de análisis forense.
- Fecha y hora de escalamiento (si el caso fue escalado).
- Si los datos implicados fueron PCI.
- Si los datos implicados fueron PII.
- Los privilegios asociados, en caso de tratarse de una key filtrada.
- La cantidad de datos implicados y su tipo.
- Equipo responsable de la contención del incidente.

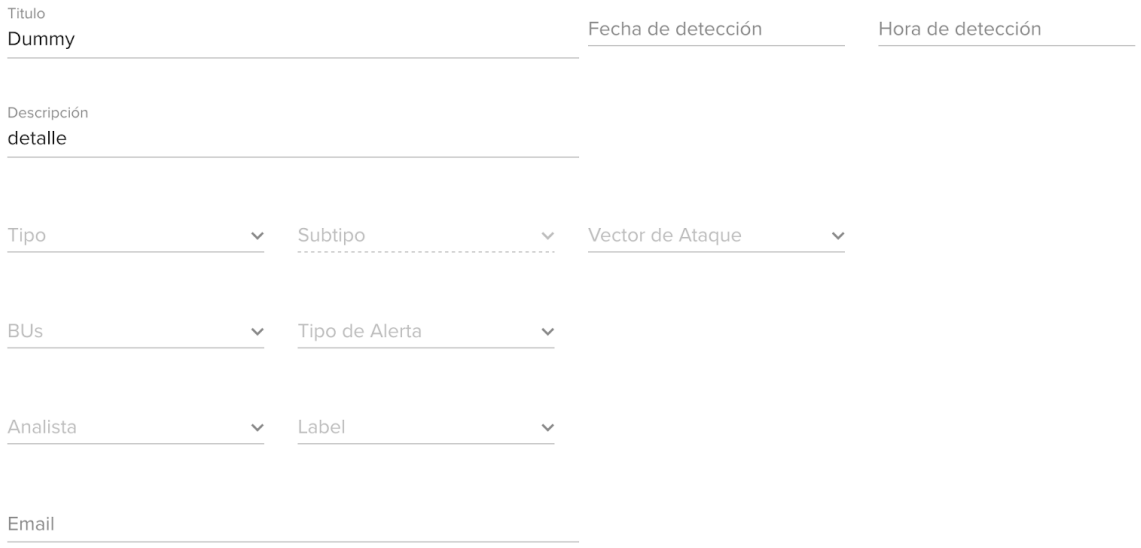

Figura 3.3: Nuevo incidente - Herramienta de registro de incidentes.

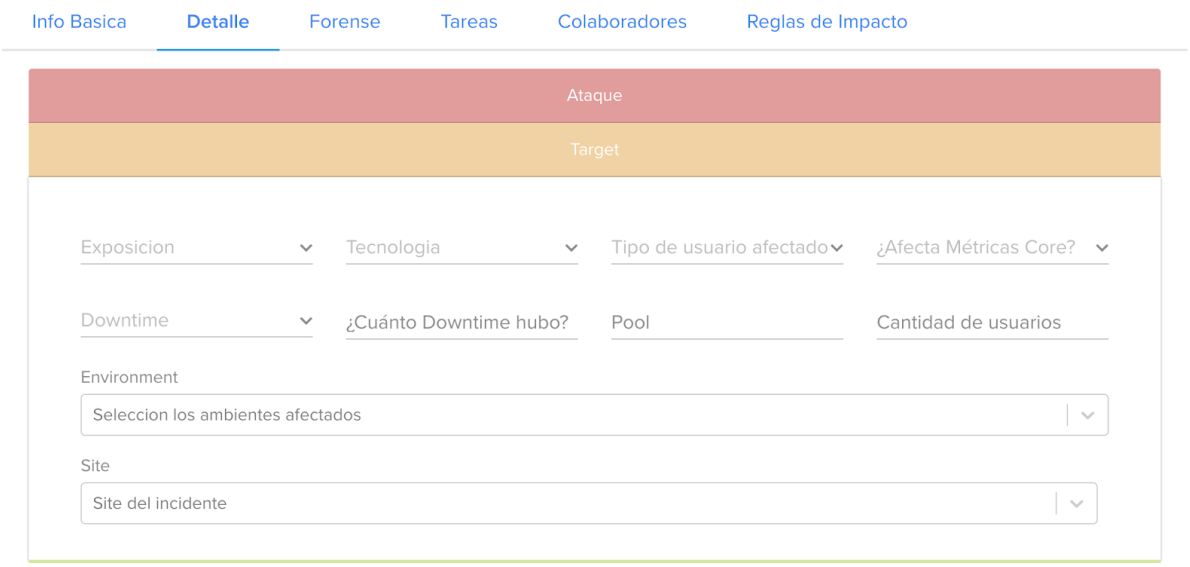

Figura 3.4 Detalles de un incidente - Herramienta de registro de incidentes.

En esta herramienta adicionalmente podemos encontrar un segmento llamado *Forense* con distintos documentos que suman información al caso, tales como:

- Evidencias del análisis forense realizado.
- **Documento Resumen Ejecutivo:** Detalle ejecutivo del caso y sus resultados posteriores al análisis.
- **Documento de Forense**: Detalle de las tareas realizadas durante el análisis forense por parte del miembro del equipo de respuesta ante incidentes.

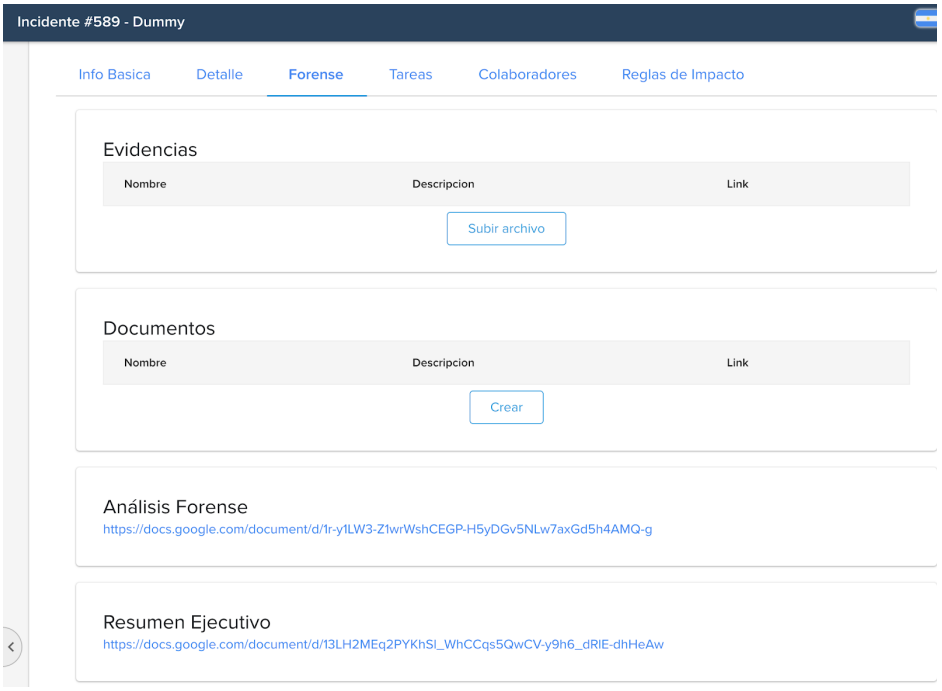

Figura 3.5 Documentos asociados - Herramienta de registro de incidentes.

A su vez, dicha herramienta provee de una sección de “Tareas” donde adicionalmente el equipo de respuesta ante incidentes puede cargar sus asignaciones tanto para sus propias actividades como las de otros equipos, que tengan responsabilidades sobre las tareas de mejora continua a realizar en base al caso.

3.6 Métricas definidas

En Mercado Libre optamos por definir distintas métricas de incidentes para poder medir la efectividad del equipo. Actualmente contamos con las siguientes métricas definidas:

3.6.1 Métricas de Gestión

Definimos las siguientes métricas generales de Gestión en Mercado Libre:

- **Cantidad de Alertas, Eventos e Incidentes:** Esta métrica la utilizamos por un lado para visualizar la cantidad general de alertas, eventos e incidentes por cada mes en las que el equipo estuvo involucrado realizando una investigación. Por otro lado se utiliza como comparativa de la cantidad de alertas que se transformaron en eventos y/o incidentes.
- **Cantidad de eventos e incidentes por tipo:** Esta métrica nos permite visualizar cuales son los tipos de eventos o incidentes que nos impactaron según la taxonomía definida.

- **Cantidad de eventos e incidentes por país:** Esta métrica nos permite visualizar cuál o cuáles fueron los países más afectados por los eventos e incidentes de cada mes.
- **Cantidad de eventos e incidentes por unidad de negocio afectada:** Esta métrica nos muestra la unidad de negocio más afectada por los distintos eventos e incidentes, la cual nos puede sugerir la implementación de nuevos controles de seguridad en dicha unidad de negocio.
- **Cantidad de eventos e incidentes por origen o fuente:** Esta métrica nos muestra las herramientas desde las que más recibimos las alertas o eventos ocurridos en un lapso de tiempo.

3.6.2 Métricas ejecutivas

Definimos las siguientes métricas ejecutivas:

- **Personal Information Leaks:** Aplica solo en los incidentes de tipo Data Breach que involucren información personal identificable (PII). La métrica muestra la cantidad de incidentes con **exposición interna** de información PII vs la cantidad de incidentes con **exposición externa** de Información PII.
- **Downtime because DDoS:** Aplica solo en los incidente de tipo DDoS y refleja el **tiempo total bajo ataque** de denegación de servicio vs **tiempo total de caída de aplicaciones o servicios de Mercado Libre.**
- **Malware:** Aplica solo en los incidentes de tipo Malicious Code y refleja la cantidad de incidentes de tipo Malicious Code que **afectan a usuarios finales** vs cantidad de incidentes de tipo Malicious Code que **afecta a la infraestructura de Mercado Libre**.

3.6.3 Metricas Tecnicas

Consideramos métricas técnicas a las que nos dan un valor representativo de cómo el equipo de incidentes responde en cuanto a la contención, análisis forense y respuesta al reporte inicial de cada incidente.
Definimos las siguientes métricas técnicas:

- **MTTC (Mean time to Contain):** Promedio del tiempo de contención de incidentes y eventos que tengan como responsable de contención al equipo de IRT.
- **MTTF (Mean time to Forensic):** Promedio del tiempo de análisis forense de los distintos tipos de incidentes y eventos.
- **MTTA (Mean Time to acknowledge):** Promedio del tiempo de respuesta al reporte inicial de cada incidente y evento.

Una vez que pudimos tomar un muestreo representativo durante el periodo de un año de cada métrica definida anteriormente, definimos valores esperados con el objetivo de mejorar la eficiencia del equipo. Estos valores dependen de las 4 clasificaciones posibles: Crítica , Alta, Media y Baja.

En la siguiente tabla se resumen los valores acordados como objetivos.

Métrica	Promedio obtenido 2020	Promedio esperado 2021
MTTF High	5 días	3 días
MTTF Medium	12 Días	9 días
MTTF Low	6 días	14 días
MTTC High	3 días	3 días
MTTC Medium	13 días	6 días
MTTC Low	18 días	10 días

Figura 3.9: Métricas de contención y forense de incidentes.

3.7 Protocolos y procedimientos de análisis y respuesta

Con el objetivo de estandarizar prácticas y mejorar el equipo, en Mercado Libre optamos por definir y documentar distintos escenarios de Incidentes tanto para contener como para realizar el análisis forense.
Uno de los objetivos de la iniciativa es optimizar los tiempos de análisis forense, y poder tener documentado tanto el flujo , como las distintas herramientas que utilizamos para responder.
Hasta el momento definimos protocolos de respuesta y forense para los siguientes tipos de incidentes:

Protocolo DDoS:

Se define un protocolo en el que se detallan los mecanismos para poder mitigar un ataque de Denegación de Servicio, como por ejemplo:

- Bloquear una o varias IP.
- Bloquear un usuario.
- Bloquear un header específico.

Por otro lado, se definen casos de uso de análisis forense post incidente, para detallar qué fue lo que ocurrió, como por ejemplo:

- Qué aplicaciones fueron atacadas.
- Cantidad de tráfico en un determinado tiempo.
- Identificar usuarios e IPs detrás del ataque.
- Con qué dispositivos se realizó el ataque.

- Desde que país o países se realizó el ataque.

Protocolo Leak Secrets Keys/Access Tokens:

Un escenario recurrente que puede suceder como evento de seguridad, se relaciona a la filtración de access tokens en sitios públicos, poniendo en riesgo las cuentas asociadas. Por eso se decidió realizar un protocolo para responder ante estos casos. El protocolo se puede resumir en los siguientes pasos:

- Obtener información detallada de los access token filtrados.
- Envío de Push Notification al dueño de la aplicación, advirtiendo el Leak y recomendando renovar sus credenciales.
- Luego de 3 días cumplidos, chequeamos si se renovaron las credenciales y caso contrario volvemos a enviar un comunicado de vencimiento.
- Para las cuentas críticas , luego de 7 días cumplidos renovamos las credenciales y enviamos notificación con dicha acción.

Protocolo Vulnerabilidades Web:

Otro de los escenarios posibles es el de vulnerabilidades web explotables a través de la aplicación de Mercado Libre o de Mercado Pago. Realizar el análisis forense de estas vulnerabilidades es un trabajo que consume mucho tiempo y recursos. Es por ello que optamos por definir un protocolo de análisis forense para las vulnerabilidades Web, enfocado en la información asociada que podría filtrarse si un ciberdelincuente explota la vulnerabilidad.

El protocolo general de Vulnerabilidades Web está dividido en varios procedimientos.

Cada procedimiento se relaciona con la posible afectación diferente tipo de información como puede ser: información personal identificable (PII) de un usuario, o la posibilidad de obtener dinero desde la cuenta de un usuario afectando así la información transaccional.

En la siguiente tabla se detalla el significado de cada procedimiento.

Procedimiento	Detalle
Procedimiento Sensible	Si la vulnerabilidad a realizar el análisis forense permite la filtración de datos PII de un usuario relacionado a: religión, origen racial, salud, genética, política, preferencias sexuales.
Procedimiento Transaccional	Si la vulnerabilidad a realizar el análisis forense permite la filtración de datos PII de un usuario relacionado a: compras, gastos, datos PCI (nro de tarjeta, cvv, cc, pin, pan), movimientos de dinero, score crediticio, carrito, favoritos, preguntas, estado

	contable.
Procedimiento limitado Alto	Si la vulnerabilidad a realizar el análisis forense permite la filtración de datos PII de un usuario relacionado a: usuario, nombre o apellido.
Procedimiento limitado Crítico	Si la vulnerabilidad a realizar el análisis forense permite la filtración de datos PII de un usuario relacionado a: contraseña, correo, número de teléfono, documento de identidad o dirección.
Procedimiento ATO	Cuando la vulnerabilidad permite al atacante cometer fraude o robo de dinero a través de la apropiación de la cuenta de un usuario
Procedimiento negocio Crítico	Si la vulnerabilidad a realizar el análisis forense no involucra filtración de datos PII, pero tiene impacto crítico hacia el negocio.

Figura 3.10: Detalles procedimiento forense de vulnerabilidades.

Dependiendo de cómo se relaciona la vulnerabilidad que estamos analizando a cada procedimiento, determinamos la cantidad mínima de logs en semanas y meses que se deberán analizar para concluir con un análisis forense completo.

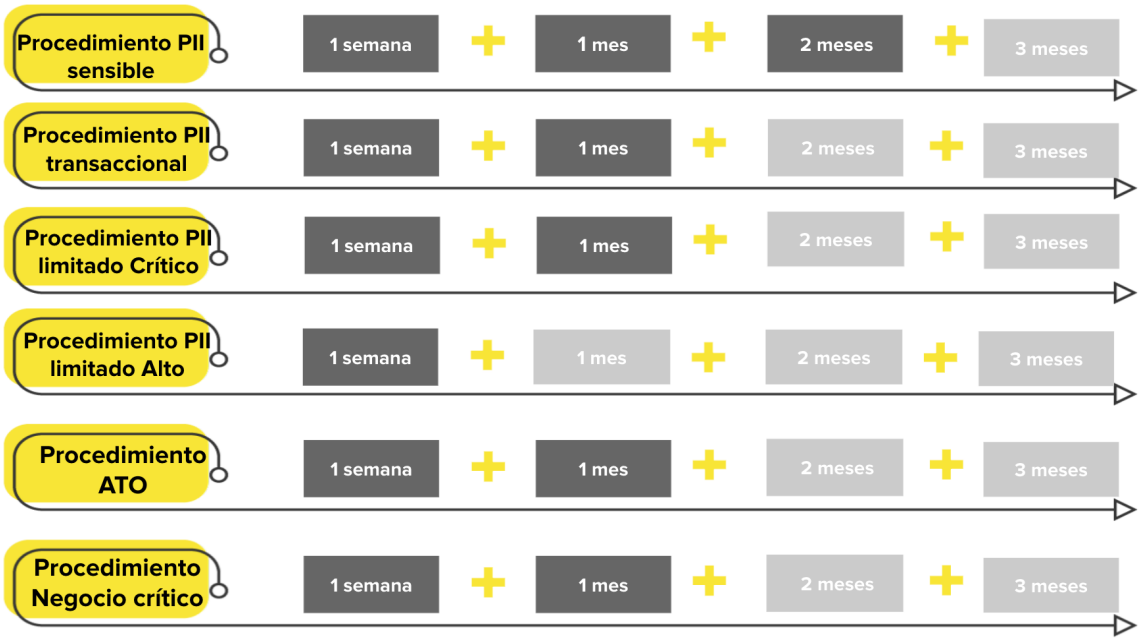

Figura 3.11 Protocolo forense de Vulnerabilidades.

3.8 Recomendaciones en gestión de Incidentes

Al pertenecer al equipo de Respuesta ante Incidentes desde el inicio, presencié la evolución del mismo.
A continuación destaco algunos puntos que me parecen importantes mencionar como parte de la conclusión y que desde mi experiencia recomendaría a todo equipo de gestión de incidentes:

Estandarización y documentación base: Contar con documentación que incluya distintos factores como la definición de evento e incidente, tipos de incidentes posibles, cálculo de impacto, etc. Generar este tipo de documentación ayuda a establecer las bases del equipo desde las etapas tempranas de creación del equipo.

Fuentes de logs y herramientas: Identificar y recopilar todas las herramientas y logs relacionados a la infraestructura de la organización que ayudarán a analizar y mitigar incidentes. Clasificarlas según los distintos escenarios posibles.

Sistema de registro de incidentes: Uno de los puntos más importantes es contar con un seguimiento de cada incidente y a su vez el listado histórico de todos los incidentes ocurridos. Es indispensable para cualquier equipo contar con un sistema donde se puedan registrar todos los incidentes junto a un reporte y los campos necesarios para tener un registro completo del incidente.
El Registro de incidentes trae múltiples beneficios al equipo:

- Cualquier persona de la organización puede conocer los incidentes ocurridos.
- Es posible contar con métricas a partir de los incidentes ocurridos.
- Ayuda a correlacionar eventos e incidentes.
- Permite estandarizar eventos e incidentes.
- Posibilita el almacenamiento de evidencias.

Difusión del equipo: Según el tamaño de la empresa puede ser necesario difundir la existencia del equipo y contar con una lista de distribución del tipo incidentes@ para recibir de forma centralizada reportes de eventos. Otra buena práctica, es notificar a los equipos y líderes involucrados en un incidente, para que pueda mantenerse actualizado del estado del incidente.

Protocolos de análisis y respuesta: Armar protocolos y procedimientos de análisis forense o de contención ante distintos escenarios ayuda a automatizar ciertos flujos y estandarizar respuestas de los incidentes más frecuentes. A su vez ayuda a los nuevos miembros del equipo a adquirir conocimientos de resolución de incidentes de manera más clara y a actuar siguiendo pasos claros y concretos en pleno incidente.

Establecer Métricas: Armar métricas de incidentes forma parte de la maduración en el equipo. Por un lado, con metricas tecnicas el equipo puede establecer un SLA[4] para responder, mitigar o realizar un análisis forense. Por otro lado, con las métricas se pueden detectar patrones y visualizar dónde se están focalizando los incidentes en periodos determinados, generando así la necesidad de implementar nuevos controles de seguridad donde sea necesario.

División de roles: El equipo de Incidentes requiere conocimientos en múltiples áreas, dependiendo del incidente que estemos tratando, conocimientos de Malware, redes, arquitectura, programación,etc. En caso de poder contar con un equipo compuesto por varios miembros, se sugiere dividir el equipo en distintos roles y especialistas en cada área. De esta forma el equipo mejorará el tiempo de respuesta ante los distintos tipos de incidentes.

[4] Un acuerdo de nivel de servicio, SLA (Service Level Agreement), es un acuerdo escrito entre un proveedor de servicio y su cliente con objeto de fijar el nivel acordado para la calidad de dicho servicio.

4. Dexter

4.1 Introducción

En el presente capítulo se describe el desarrollo de una herramienta llamada “Dexter”, creada para gestionar y automatizar análisis forenses de eventos e incidentes.

En la sección 4.2 se describe el objetivo principal de la herramienta desarrollada.

En el apartado 4.3 se detallan los problemas recurrentes detectados en el equipo que originaron la idea de desarrollo de la herramienta.

Luego, en la sección 4.4 se describen las tecnologías seleccionadas para el desarrollo, con sus respectivas ventajas.

La sección 4.5, detalla la arquitectura de Dexter, primero se introduce a Fury, la plataforma interna donde se ejecuta Dexter. Luego se muestra cada componente que conforma la arquitectura de Dexter y cómo interactúa entre sí.

Luego de la primera versión desarrollada de Dexter productiva, se detectó una problemática que se detalla en el apartado 4.6.

A partir de un caso de uso, se muestra el flujo de la aplicación desde que se crea un caso nuevo, hasta que se realiza el análisis de logs con una Tool en la sección 4.7.

En el apartado 4.8 se explican las principales características que permiten que Dexter sea eficiente, seguro y preciso al obtener el resultado de un análisis.

En la sección 4.9 se cuentan distintos casos de éxito en el equipo de respuesta ante incidentes, donde Dexter se utiliza para eventos o incidentes reales.

Finalmente, con la motivación de poder replicar la herramienta Dexter en otra organización, se detalla en la sección 4.10 un paso a paso general con ejemplos.

4.2 Objetivo

El proyecto Dexter tiene como objetivo principal, desarrollar una herramienta que permita automatizar, simplificar y mejorar el análisis de los distintos tipos de incidentes y eventos en Mercado Libre. Además, poder centralizar los logs e información necesaria, reduciendo así notoriamente el tiempo de demora de análisis de un evento o incidente de seguridad.

4.3 Problemáticas detectadas

Durante los años transcurridos en Mercado Libre contamos con distintos tipos de incidentes a analizar, sin embargo, durante los años 2019 y 2020 distinguimos principalmente 2 tipos de eventos e incidentes que el equipo necesitaba dar una rápida respuesta.

Por un lado, los distintos ataques de denegación de servicio [12] que atentaban contra la disponibilidad de distintas aplicaciones. En estos ataques el equipo necesita entender rápidamente qué está ocurriendo para poder contener rápidamente el incidente, así como también poder brindar un análisis forense correcto, luego que el incidente fue contenido.

Por otro lado, recibimos reportes de vulnerabilidades Web desde el equipo de Application Security que atentan al sitio de Mercado Libre y Mercado Pago. En estos casos el equipo de respuesta ante incidentes, deberá realizar un análisis forense adecuado para saber si hubo explotación de la vulnerabilidad reportada y en ese caso obtener un reporte completo con los datos afectados.

Con el análisis de estos 2 escenarios de incidentes, nos encontramos con distintas problemáticas a la hora de realizar el análisis tanto forense como de contención.
A continuación se listan algunas de las problemáticas detectadas:

Volumen de logs: Mercado Libre cuenta un tráfico promedio de 2.2 millones de requests por segundo aproximadamente. Lo que hace complejo poder consultar los logs de tráfico de las miles de aplicaciones que podrían ser atacadas en un incidente, se requiere realizar múltiples consultas con la mayor cantidad de filtros posibles [17] del ataque para acotar la búsqueda y obtener resultados claros del ataque.

Errores de Athena: Athena [3] es un servicio de AWS (Amazon Web Services) que nos permite consultar a través de SQL los logs de tráfico que se almacenan generalmente en otros servicios propios de AWS como el llamado S3 [4].
El servicio de Athena suele tener múltiples errores en el frontend al ejecutar distintas consultas, perdiendo así la ejecución de la misma y haciendo que su uso diario sea complejo.

Información descentralizada en APIs: En Mercado Libre contamos con distintas APIs que nos brindan información de usuarios, IPs, aplicaciones, etc. Esta información es útil a la hora de un incidente, dado que nos permite enriquecer con distintas fuentes tanto el origen como el destino del ataque en curso. Al tener la información distribuida en múltiples APIs, se dificulta y se desperdicia tiempo al consultar cada dato que necesitamos.

Repetición de análisis: Otra de las problemáticas encontradas, fue que usualmente estamos realizando tareas repetitivas al ejecutar las mismas consultas al realizar el análisis de logs, esto no solo sucede en los Incidentes de DDoS (Ataque de Denegación de servicio) sino también al realizar el análisis forense de una vulnerabilidad Web, donde buscamos anomalías de tráfico hacia la aplicación y url donde se encontró la vulnerabilidad web.

A partir de las distintas problemáticas planteadas en la sección anterior, nace la idea de desarrollar un proyecto que automatice y simplifique las tareas diarias de análisis forense del equipo.

4.4 Lenguajes seleccionados

Dexter se compone de un backend desarrollado en lenguaje Python, seleccionamos este lenguaje, debido a que cumplía con nuestros requisitos y se ajustaba al tamaño de la aplicación a desarrollar. Python [5] es un lenguaje de programación interpretado de código abierto, es multiplataforma y multiparadigma. Una de sus principales ventajas es la baja curva de aprendizaje para aprender respecto a otros lenguajes, posee una gran calidad y simpleza en su sintaxis.

Para el Frontend, utilizamos **React.JS** [6], uno de los lenguajes más utilizados y con mayor soporte en Mercado Libre para realizar desarrollos internos. React es una biblioteca de JavaScript para construir interfaces de usuario, es declarativo y está basado en componentes. Estos componentes se crean encapsulados, manejan estados y se pueden reutilizar fácilmente, haciendo que la escritura y lectura de código sea sencilla.

4.5 Arquitectura

4.5.1 Fury

Dexter se ejecuta y gestiona dentro de Fury. Fury se trata de **una plataforma como servicio (PaaS)** interna que permite abstraer a cada desarrollador de la infraestructura necesaria, para que su aplicación se ejecute correctamente. Cada vez que un desarrollador implementa o crea una versión nueva de una aplicación, Fury se encarga de crear todos los componentes necesarios para el deploy de la aplicación.
Fury se encarga de crear una red [7] de instancias, junto a un load balancer [8] para distribuir equitativamente el tráfico. De esta manera logra que la infraestructura de cada aplicación compuesta por instancias **EC2** [10] sea **balanceada** y **autoescalable** [9]. A su vez la utilización de **Docker** abstrae al desarrollador de la instalación de todas las dependencias necesarias.

Fury también administra un repositorio de **Git** por aplicación, una canalización de **CI / CD** con diferentes estrategias de implementación, logs de aplicaciones en el servicio de **Kibana** y recopilación de métricas en el servicio de **DataDog**

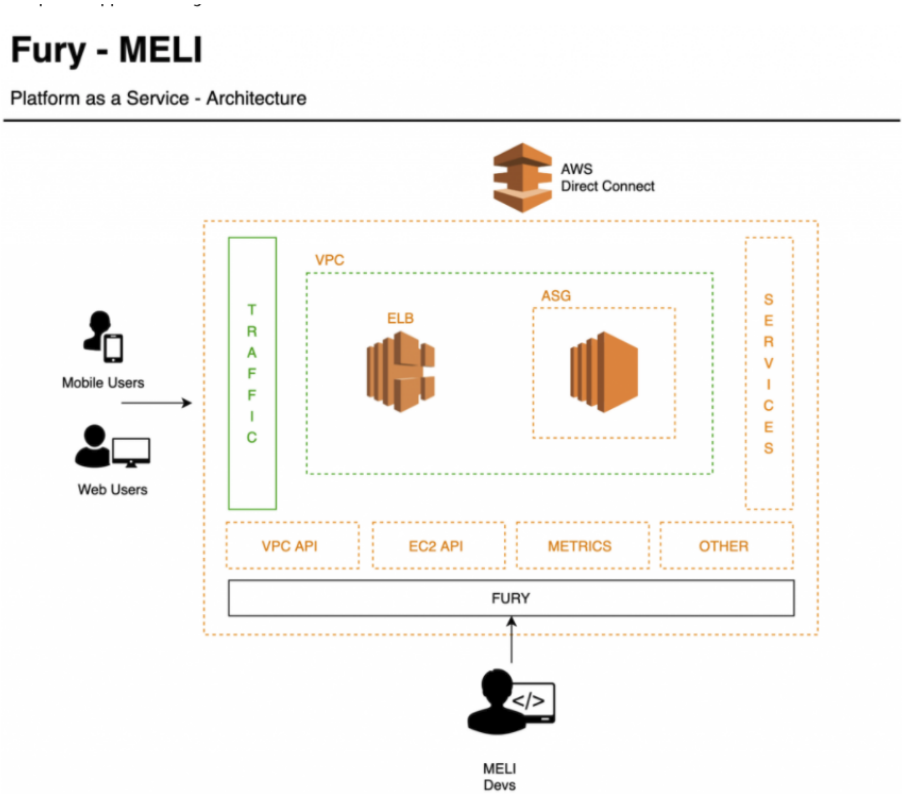

Figura 4.1: Arquitectura Fury [11]

4.5.2 Arquitectura Dexter

Dexter se compone y utiliza principalmente los siguientes componentes y aplicaciones:

Capa de tráfico: Llamamos capa de tráfico a los logs de cada request realizado hacia las aplicaciones de Mercado Libre. En cada log podemos ver información detallada de cada request, como por ejemplo:

- IP Origen
- URL
- Status Code
- Método HTTP
- HTTP Referrer
- Timestamp
- Aplicación Destino
- Cookies

Precisamente son logs de los distintos servidores nginxs que se encuentran detrás de cada aplicación. Estos logs son almacenados en Buckets S3 [4] durante 3 o 6 meses dependiendo si son logs de aplicaciones internas o externas. Esto se define por la política de retención de logs y cumplir con las regulaciones necesarias de Mercado Libre y Mercado Pago.

Finalmente, para consultar estos logs utilizamos un servicio de Amazon llamado Athena [3] que mediante SQL nos permite realizar querys para obtener los logs almacenados.

Data Extractor: Se trata de una API interna que nos permite abstraernos de lógicas comunes al realizar consultas en Athena. De esta forma definimos una consulta con argumentos mediante el formato JSON, luego podremos ejecutar la consulta guardada desde nuestra aplicación con los parámetros necesarios. Finalmente, la ejecución devuelve un enlace a un archivo en donde se almacenan los resultados de la ejecución.

Servicios de Fury:

- **Big-Queue**: se trata de un sistema de cola de mensajes compuesto por topics y consumers, donde los topics representan colas donde se almacenan los mensajes recibidos, y los consumers (asociados a un topic) pueden ejecutar diferentes lógicas al recibir un mensaje.
- **Job**: permite que cierta lógica de código se ejecute cada vez que ocurre una condición particular.
- **Object Storage:** es un servicio que permite almacenar, recuperar y administrar cualquier tipo de objeto en Amazon S3.
- **DB:** Dexter cuenta con 2 esquemas de bases de datos MySQL, una para la versión productiva de Dexter y otra para la versión de Test. Ambas se componen de un cluster de un servidor Master y dos servidores esclavos.

Dexter: Dexter se resume en los siguientes componentes y clases:

- **Case**: Un caso en Dexter se corresponde a un evento o incidente que se desea analizar, actualmente Dexter se enfoca en automatizar los eventos de tipo Availability y Vulnerable. Cada caso se compone de una o varias Tools para realizar el análisis.

- **Tool**: Se define una tool en Dexter como una consulta con uno o varios parámetros que se ejecutan con el objetivo de obtener logs o información sobre un evento ocurrido. Las consultas comúnmente se ejecutan sobre la api de Data Extractor que va a ser la encargada de ejecutarla sobre Athena.

- **Parser:** El Parser se compone de distintas clases que se van a encargar de realizar los siguientes pasos:
    a. Obtener los resultados almacenados en la api de Data Extractor.
    b. Fusionar los resultados, en caso que se hayan dividido en varios días.
    c. Enriquecer los resultados con distintas APIs internas (véase la sección 4.8.3).
    d. Guardar los resultados en el almacenamiento propio de Dexter.

- **Enrich APIs:** A través de distintas APIs internas, Dexter enriquece los logs obtenidos en la capa de tráfico

En el siguiente diagrama se muestran los componentes detallados y la interacción entre cada uno.

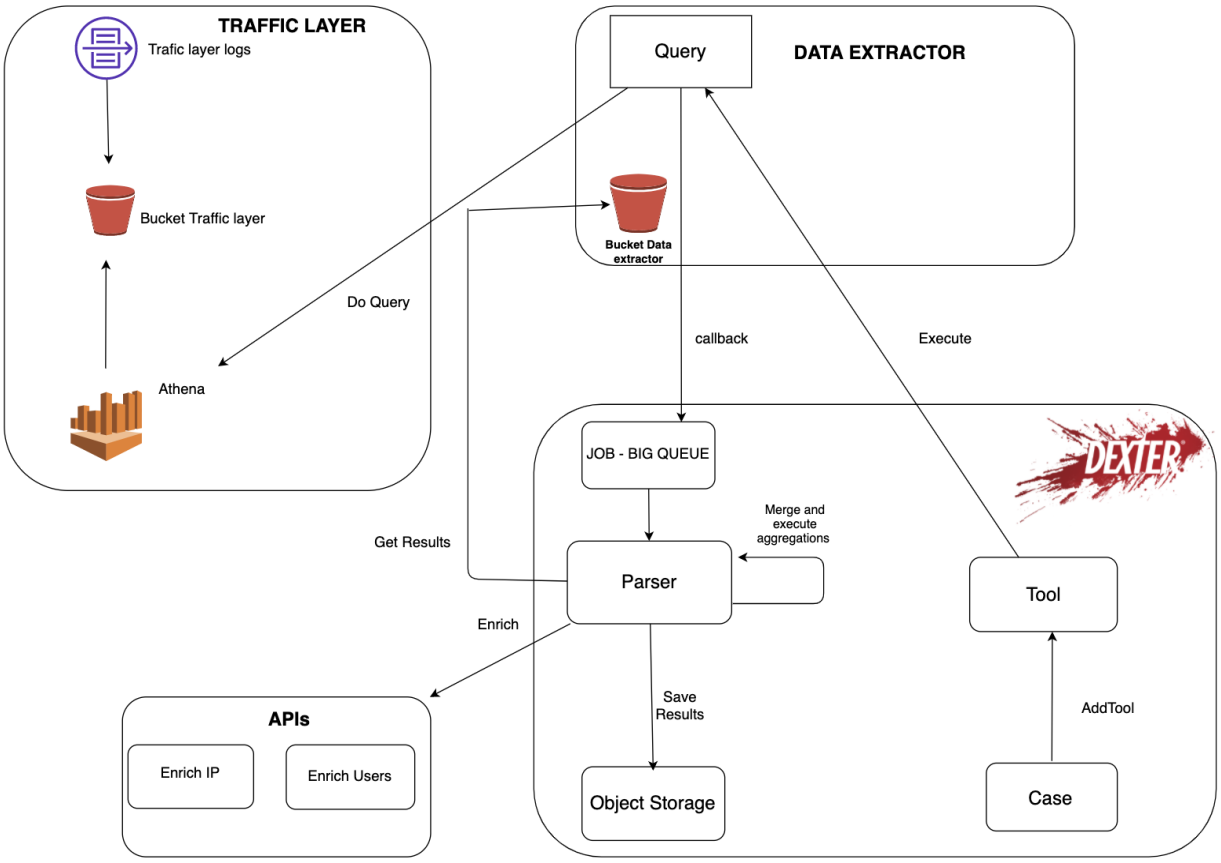

Figura 4.2: Diagrama de Arquitectura Dexter

4.6 Costo de consultas vs tiempo de procesamiento.

Uno de los grandes desafíos que nos encontramos en el desarrollo de Dexter es el de costo de consultas a Amazon vs tiempo de procesamiento de logs.

Amazon cobra actualmente 5 USD por TB de datos escaneados [16] mediante el servicio Athena. El cobro de la consulta se basa en la cantidad de datos escaneados por consulta.

Una de las principales fuentes de logs consultadas con Dexter es la capa de tráfico, anteriormente mencionada (véase la sección 4.5.2). Debido a la cantidad de aplicaciones y tráfico que se almacena, los archivos de logs son archivos que suelen superar los TB de información, por lo cual, cada consulta ejecutada supone un desafío de costos. Por ejemplo, cuando realizamos una consulta por Athena para obtener logs en un periodo de 15 días para una sola aplicación puntual de Mercado Libre, se escanean aproximadamente 5 TB de logs, se demora aproximadamente 30 minutos y la consulta tiene un costo de 25 USD.

Debido al desafío de costos, se decidió realizar la menor cantidad de consultas posibles en Athena y realizar el procesamiento o agregaciones de logs a través de Dexter. De esta forma, Dexter recibe archivos menos procesados y de gran tamaño pero esto conlleva una reducción de costos de consultas con Athena considerable.

Esta decisión, derivó en otra problemática relacionada al tiempo que puede demorar el procesamiento de logs en archivos grandes, como por ejemplo cuando se realizan las agregaciones o fusiones de archivos al analizar varios días de logs, estas consultas podrían tardar desde varios minutos a horas. El desafío a futuro es encontrar un equilibrio entre el tiempo de procesamiento y el costo de las consultas.

4.7 Flujo Dexter

En la siguiente sección se van a mostrar los distintos flujos y casos de uso posibles dentro de Dexter, desde que se inicia la aplicación hasta que se analiza un evento específico.

4.7.1 Listado de Casos

Cuando Dexter se inicia, en la primera pantalla o Home se pueden visualizar todos los casos creados asociados a un evento o incidente de seguridad. Se pueden aplicar varios tipos de filtros para buscar un caso específico:

- Filtro por dueño del caso
- Filtro por categoria.
- Filtro por fecha de creación.
- Filtro por nombre del caso.

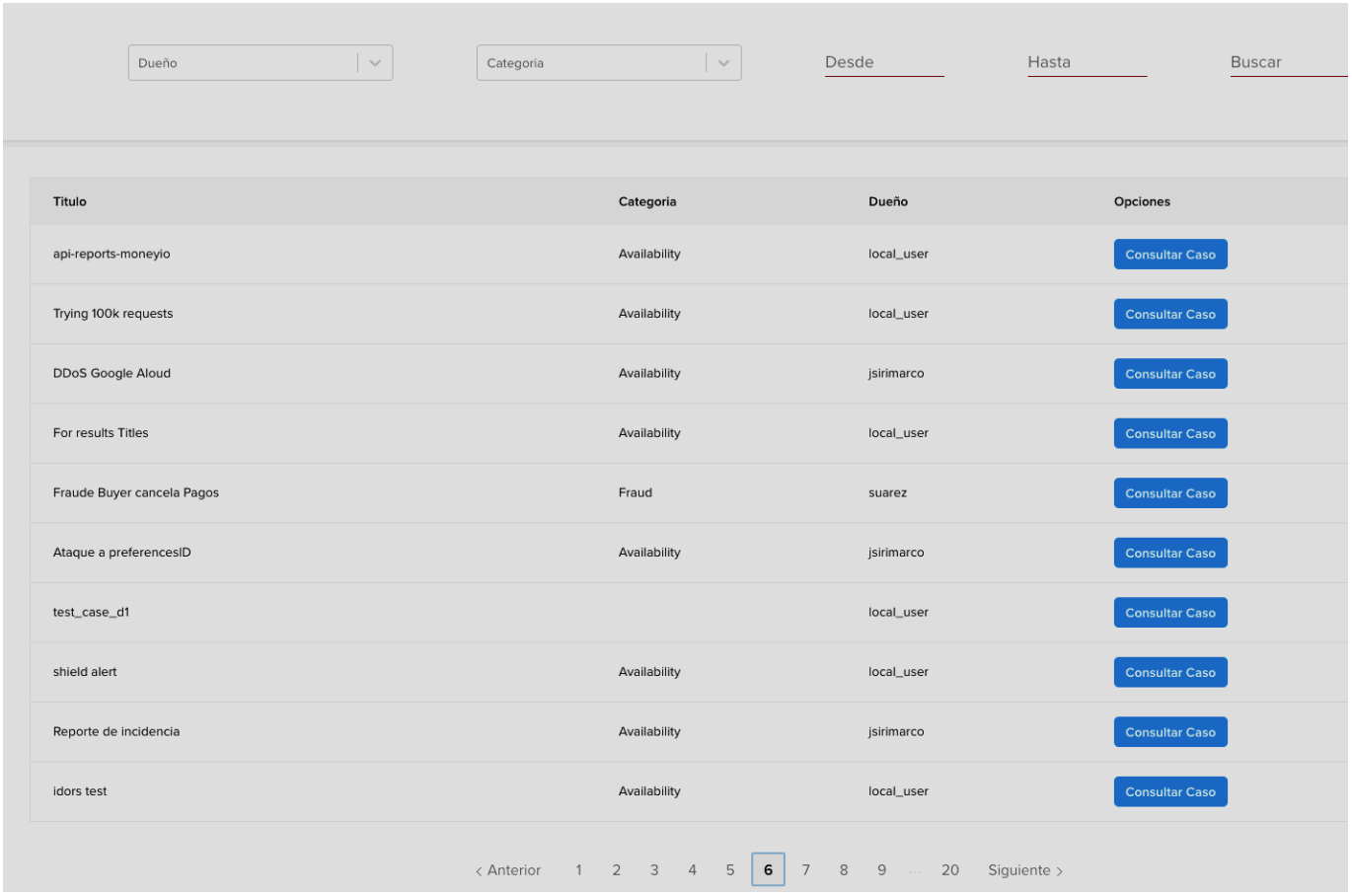

Figura 4.3: Dexter - Listado de casos.

4.7.2 Nuevo Caso

Para crear un caso nuevo nos dirigimos al menú de la izquierda “New Case”, que nos redirige a la pantalla de creación de caso. En esta pantalla Dexter nos solicita 3 campos:

1. **Descripción**: Nombre del caso para poder identificar el evento o incidente de seguridad. Con esta descripción podremos buscar el caso luego en el Home mediante los filtros.
2. **Categoría de Investigación**: Este campo requiere el tipo de Evento o Incidente que estamos tratando, dependiendo su Taxonomía (véase la sección 3.3). Las categorías actuales posibles son:
    - Vulnerability
    - Availability.
    - Data Breach.
    - Fraud.

3. **Subtipo de Investigación**: Cada tipo de incidente a su vez está asociado a un subtipo de incidente que nos va a permitir seleccionar las herramientas más adecuadas para realizar el análisis forense. Ejemplo: Actualmente para eventos de tipo Vulnerable podremos seleccionar como subtipo IDOR y para eventos de tipo Availability podremos seleccionar DDoS.

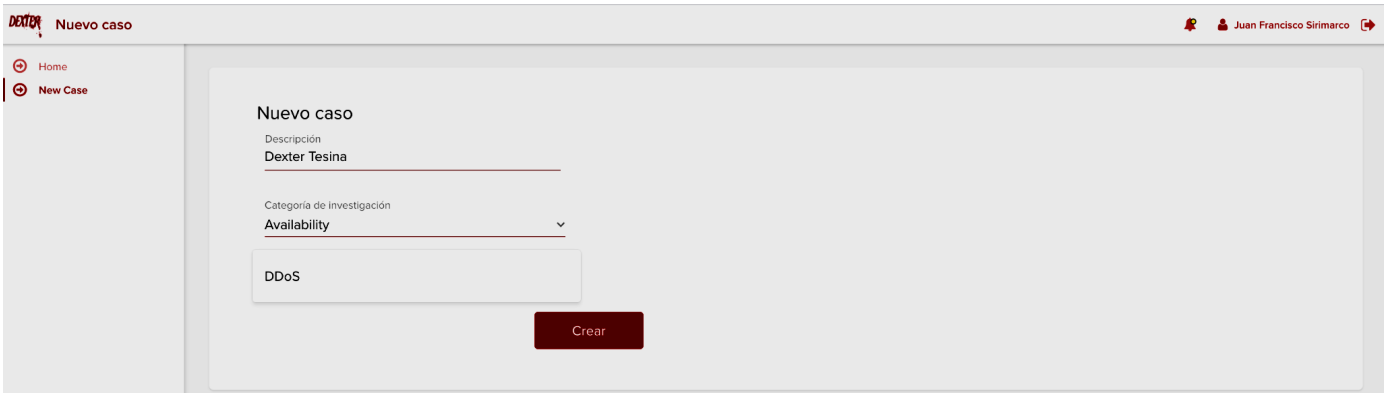

Figura 4.4 Dexter - Nuevo caso.

4.7.3 Agregar Tool a un caso

Esta vista nos va a permitir asociar una o varias Tools al caso creado previamente para luego poder ejecutarlas.
Existen distintas Tools por cada tipo de Incidente:

Availability:

- **Host Attacked:** Se utiliza cuando detectamos una anomalía de tráfico sobre un dominio completo o múltiples aplicaciones que pertenecen a un mismo dominio. De esta forma obtenemos una perspectiva general de lo que está ocurriendo bajo todo un dominio.
  Parámetros requeridos:
  - Host: Cualquier dominio que esté bajo la infraestructura de Fury perteneciente a Mercado Libre. Ejemplo: www.mercadopago.com.uy
  - Fechas: From y To
  - Scope
- **Application Attacked:** Se utiliza cuando nos informan que una aplicación puntual está sufriendo un aumento de tráfico. Al utilizar esta Tool, podemos visualizar un resumen del análisis de logs de una aplicación específica.
  Parámetros requeridos:
  - Aplicación: Nombre de cualquier aplicación interna que corre bajo la infraestructura de Fury. Ejemplo: home-landing
  - Fechas: From y To
  - Scope

- **IP Attack:** Cuando tenemos una o varias IPs que están realizando un posible ataque de Disponibilidad, utilizamos esta Tool para ver todo el tráfico relacionado a una sola IP.
  Parámetros requeridos:
  - IP: La IP origen del atacante. Ejemplo: 190.200.116.113

        - Fechas: From y To
        - Scope
    - **User Attack:** Utilizamos esta Tool cuando queremos investigar la actividad de tráfico de un usuario en particular.
      Parámetros requeridos:
        - User ID: Identificador único de usuario dentro del ecosistema de Mercado Libre. Ejemplo: 12345678
        - Fechas: From y To
        - Scope

Vulnerability:

- **IDOR Search Tool:** Utilizamos esta herramienta para realizar análisis forense de una vulnerabilidad web de tipo IDOR[5]
  Parámetros requeridos:
    - Fechas: From y To
    - Host: Cualquier dominio que esté bajo la infraestructura de Fury perteneciente a Mercado Libre. Se corresponde con el dominio donde se encuentra la vulnerabilidad IDOR Ejemplo: www.mercadopago.com.br
    - Method: Método HTTP, por el cual se explota la vulnerabilidad que estamos analizando. Ejemplo: PUT
    - URL: Endpoint específico donde se encuentra la vulnerabilidad IDOR. Ejemplo: /api/bank_account?user_id=%
    - Scope

Cada tool requiere de distintos parámetros para poder ejecutarse, exceptuando por los siguientes parámetros que se requieren en la mayoría de las Tools:

- From: Fecha desde donde se inicia la búsqueda de Logs.
- To: Fecha hasta donde se realiza la búsqueda de logs.
- Scope: Distinción entre logs de aplicaciones internas de Mercado Pago, Mercado Libre y aplicaciones públicas de Mercado Pago y Mercado Libre

[5] Insecure Direct Object Reference, también llamado IDOR. Se refiere a cuando una referencia a un objeto de implementación interna, tal como un archivo o llave de base de datos, se expone a los usuarios sin ningún otro control de acceso.

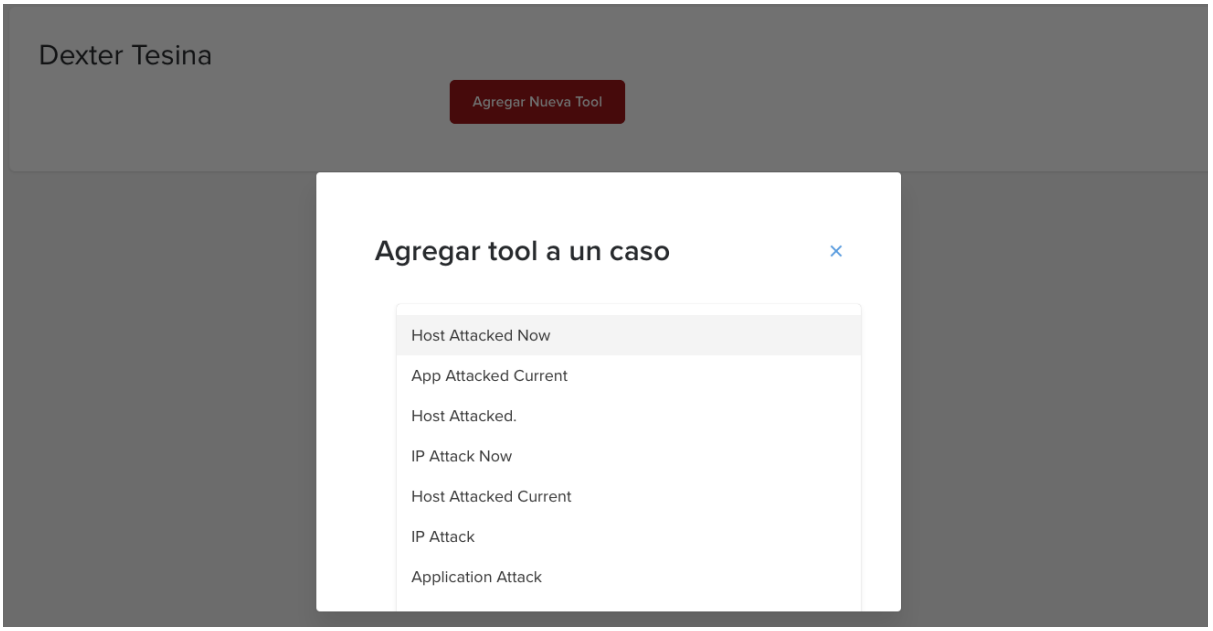

Figura 4.5: Dexter - Agregar Tool a un caso.

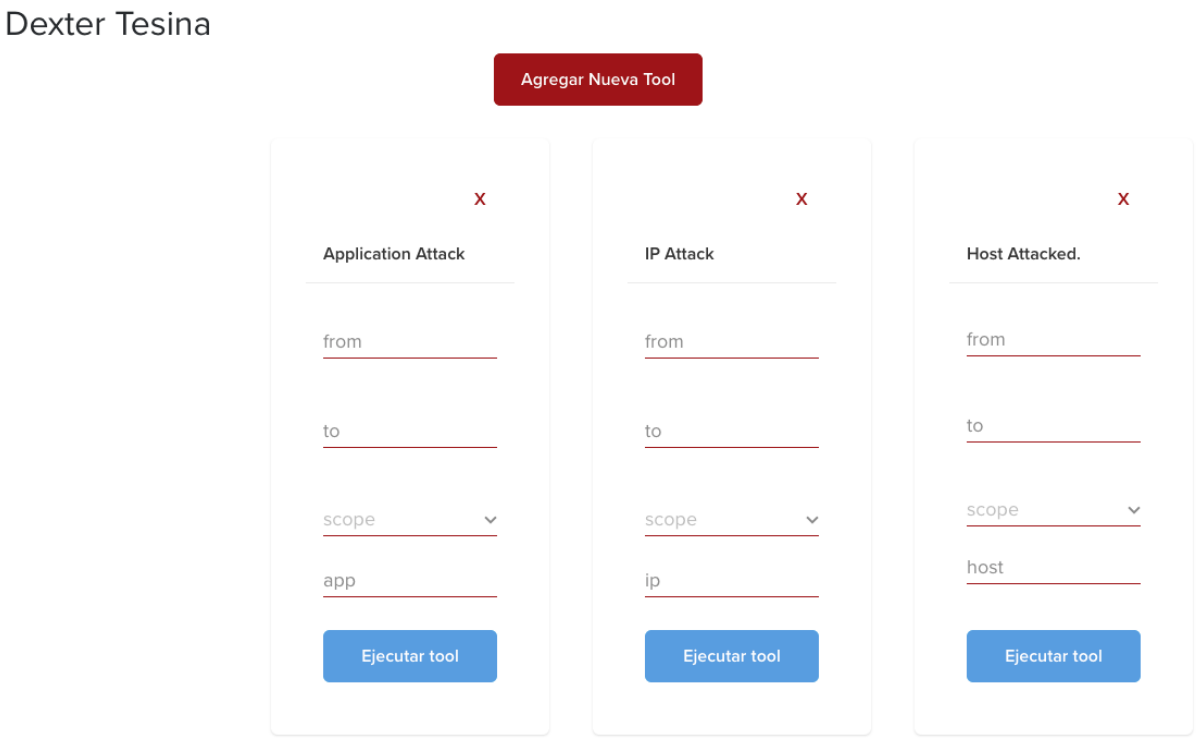

Figura 4.6: Dexter - Caso con Tools asociadas.

4.7.4 Ejecución de una Tool

Una vez asociada una o varias Tools, el siguiente paso es ejecutarlas para poder obtener una conclusión o un acercamiento al análisis forense que estamos llevando a cabo. El frontend nos muestra que se están ejecutando las consultas junto a los parámetros con los que se ejecutaron:

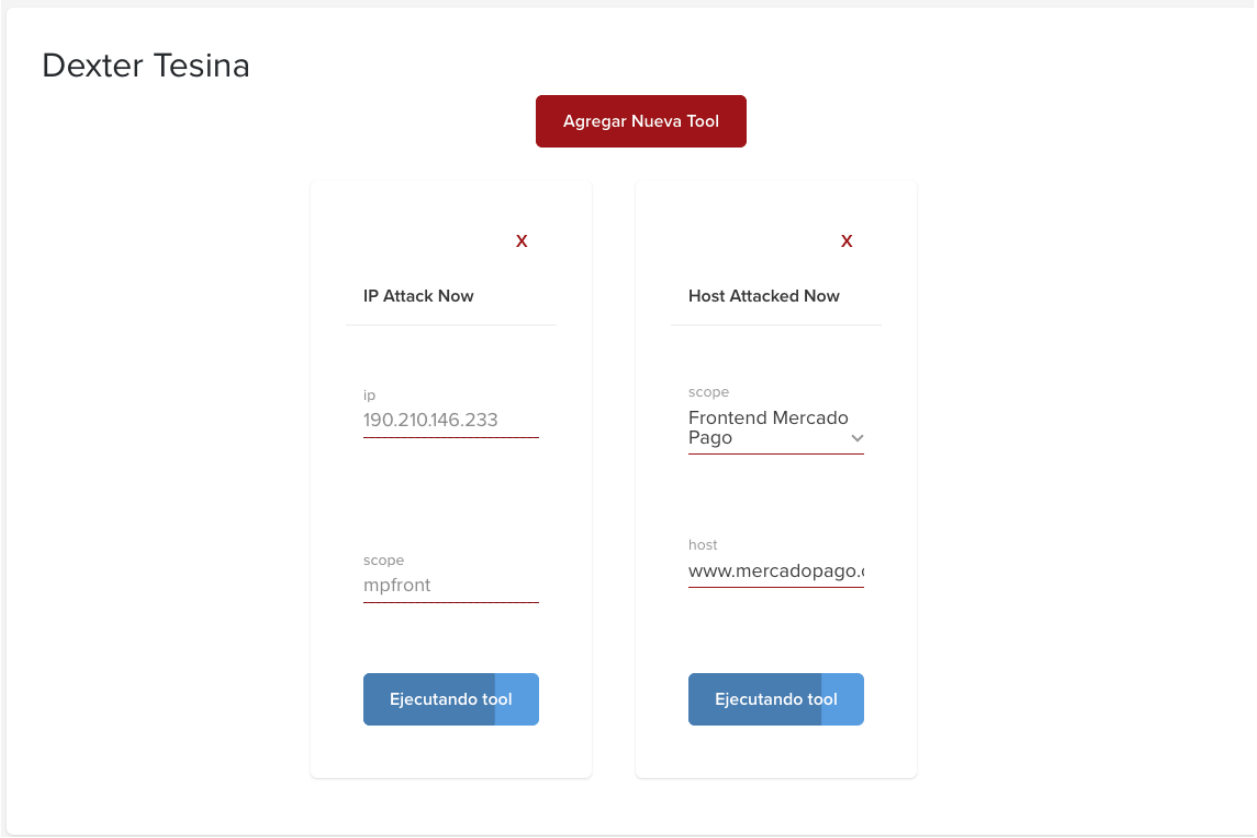

Figura 4.7: Dexter- Ejecución de Tools.

Cuando la ejecución finalice, un bot nos avisara por Slack[6] que la ejecución finalizó (véase la sección 4.8.1). Los botones debajo de cada Tool pasarán a estar en color Rojo permitiendo la visualización de los resultados.

[6] Slack es la herramienta de comunicación interna del equipo.

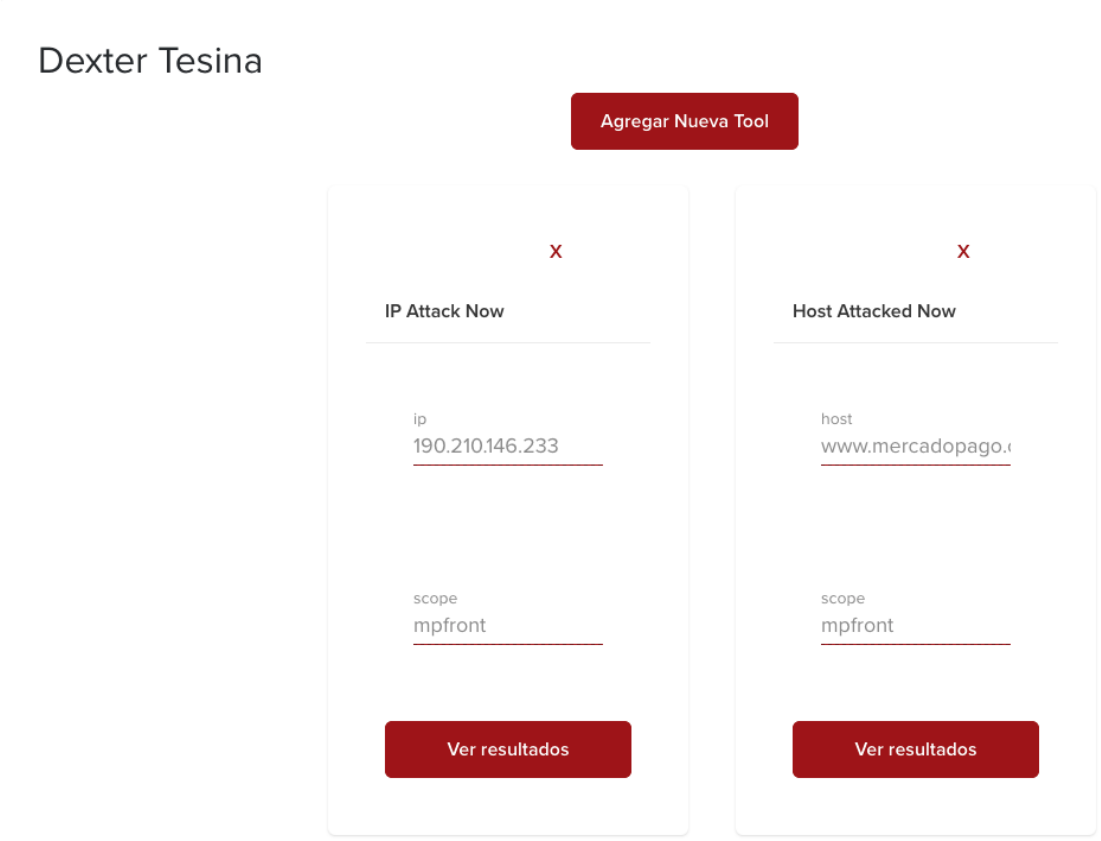

Figura 4.8: Dexter - Ejecución de tools finalizada.

4.7.5 Visualización de resultados

Finalmente accedemos a la visualización de resultados, en la pantalla se podrán ver los logs que resultaron de la consulta que ejecutamos previamente.
En esta pantalla, vamos a ver varias pestañas que contienen las **agregaciones** de los logs (véase la sección 4.8.2)**.**
Las agregaciones junto a los enriquecimientos de información, nos dan una perspectiva completa del evento o incidente que estamos analizando

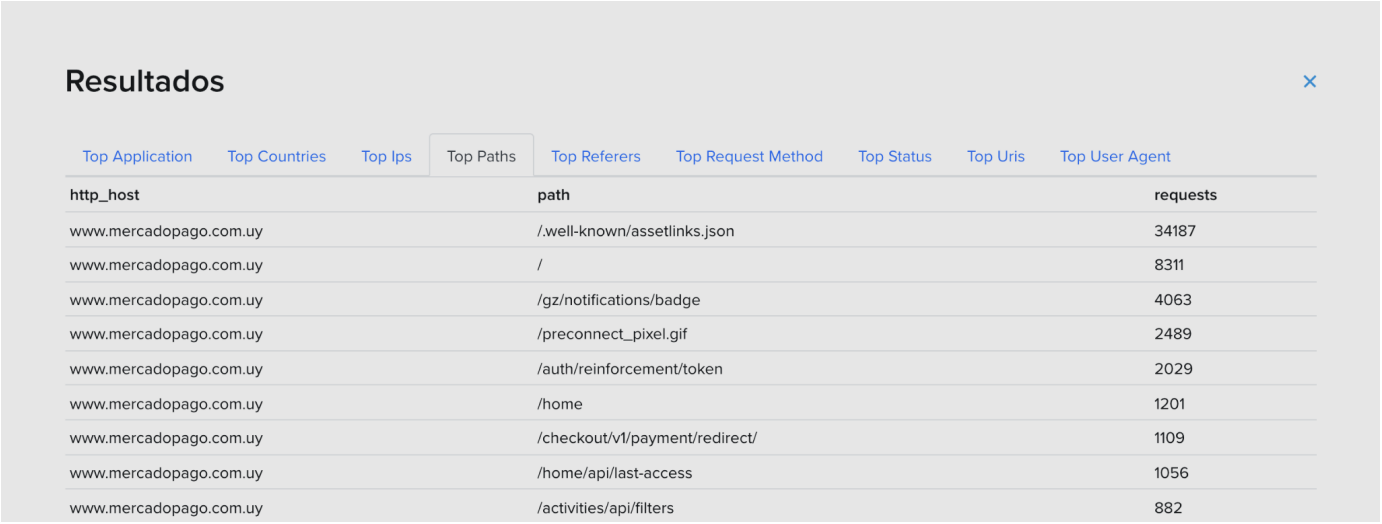

Figura 4.9: Dexter - Resultado de ejecución.

4.7.6 Eliminar Tool

Finalmente en caso que los resultados obtenidos, no sean los esperados, podremos eliminar una Tool asociada a un caso, ya sea que esté ejecutada o no. Esto nos permite limpiar el caso creado, asociar nuevas tools y ejecutarlas hasta obtener los datos buscados.

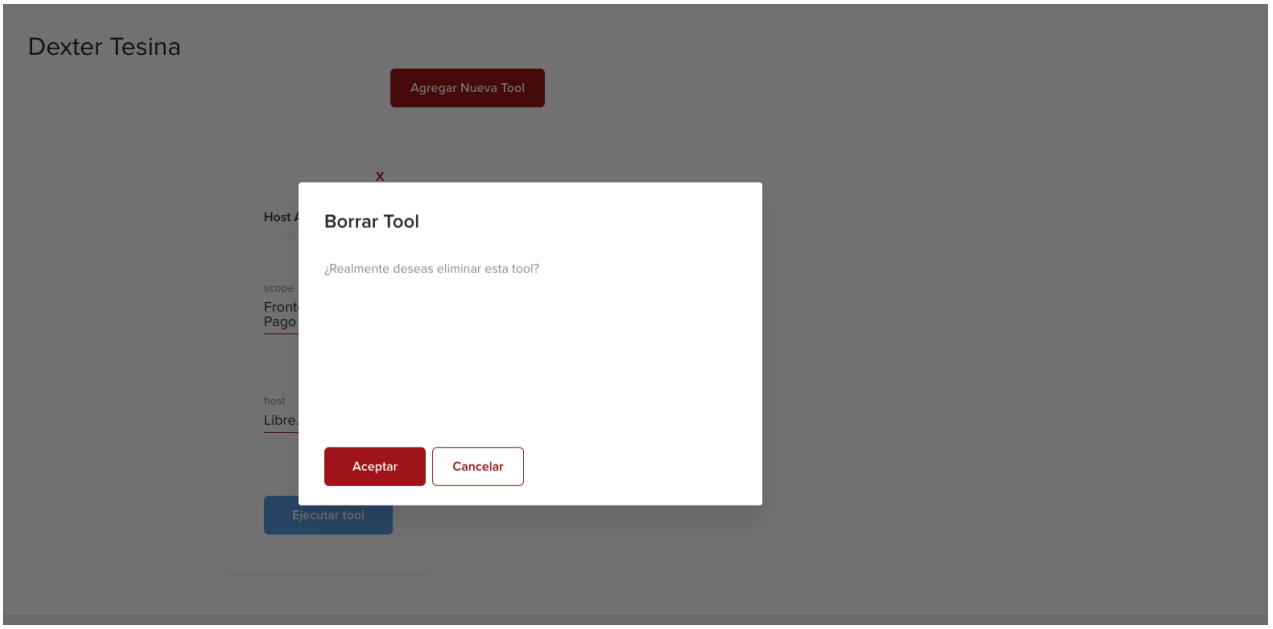

Figura 4.10: Dexter - Eliminar tool asociada.

4.8 Características adicionales

4.8.1 Aviso de Finalización por Slack

Mercado Libre cuenta con más de 8000 aplicaciones personalizadas que procesan un promedio de 2.2 millones de request por segundo que es almacenado en logs de tráfico. Debido a esto, realizar un análisis forense sobre estos logs no es una tarea sencilla ni rápida. Dependiendo de la aplicación y el tiempo que queremos obtener, esta tarea puede demorar varios minutos. Incluso cuando realizamos el análisis forense de una vulnerabilidad web expuesta (véase la sección 3.9), realizamos un análisis forense de meses, lo cual la ejecución de estas consultas podría llegar a demorar hasta 30 minutos.

Es por ello que una de las características implementadas es la integración con el servicio de Mensajería Slack. Cuando una consulta finaliza, un BOT nos avisa mediante un canal compartido con todos los usuarios de Dexter, que la ejecución ha finalizado, mostrando los parámetros ejecutados y un link para acceder directamente a los resultados obtenidos.

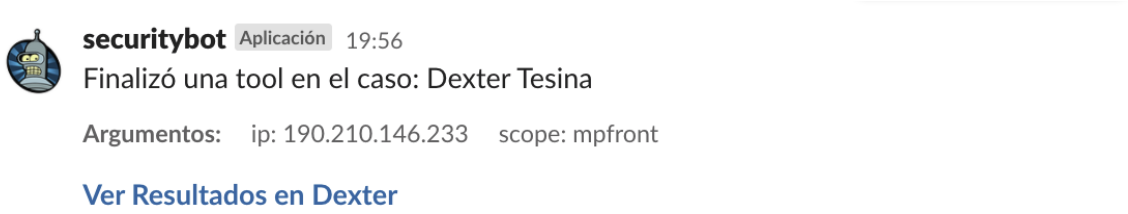

Figura 4.11:Notificación vía Slack de finalización.

4.8.2 Validación de inputs

Durante la etapa de desarrollo, se realizó un **Threat Model**[7][13] a Dexter.

Para llevar a cabo el Threat Model, primero se realizó una metodología de brainstorming junto con el equipo de Application Security donde se mostró la arquitectura y diagramas de Dexter, así como también los distintos inputs de la aplicación.

De esta forma, se generó un listado de hallazgos que luego fueron categorizados según el framework **STRIDE** [14] y probados exitosamente en Dexter.

Los hallazgos se pueden resumir en la siguiente tabla:

[7] Un Threat Model consiste en identificar y priorizar vulnerabilidades estructurales en la etapa de diseño de una aplicación, mediante el modelo de desarrollo.

Stride	Attack	Hallazgo	Impacto
Tampering	SQL Injection	Falta de validación en inputs de usuario.	Obtener Base de datos Completa.
Elevation of Privilege	SSRF Path Traversal	Error en concatenación de URLs en Middleware.	Escalar privilegios,solo en caso que Dexter se publique.

Figura 4.12: Hallazgos en Threat Model.

Solución: Como solución al hallazgo de Tampering se sugirió realizar una validación en cada entrada de usuario, mediante los valores posibles que cada campo puede aceptar. Por lo tanto, se implementó un validador, que va a chequear según la herramienta asociada, cuales son las posibles entradas que el usuario podría ejecutar, Impidiendo de esta forma una posible inyección SQL.

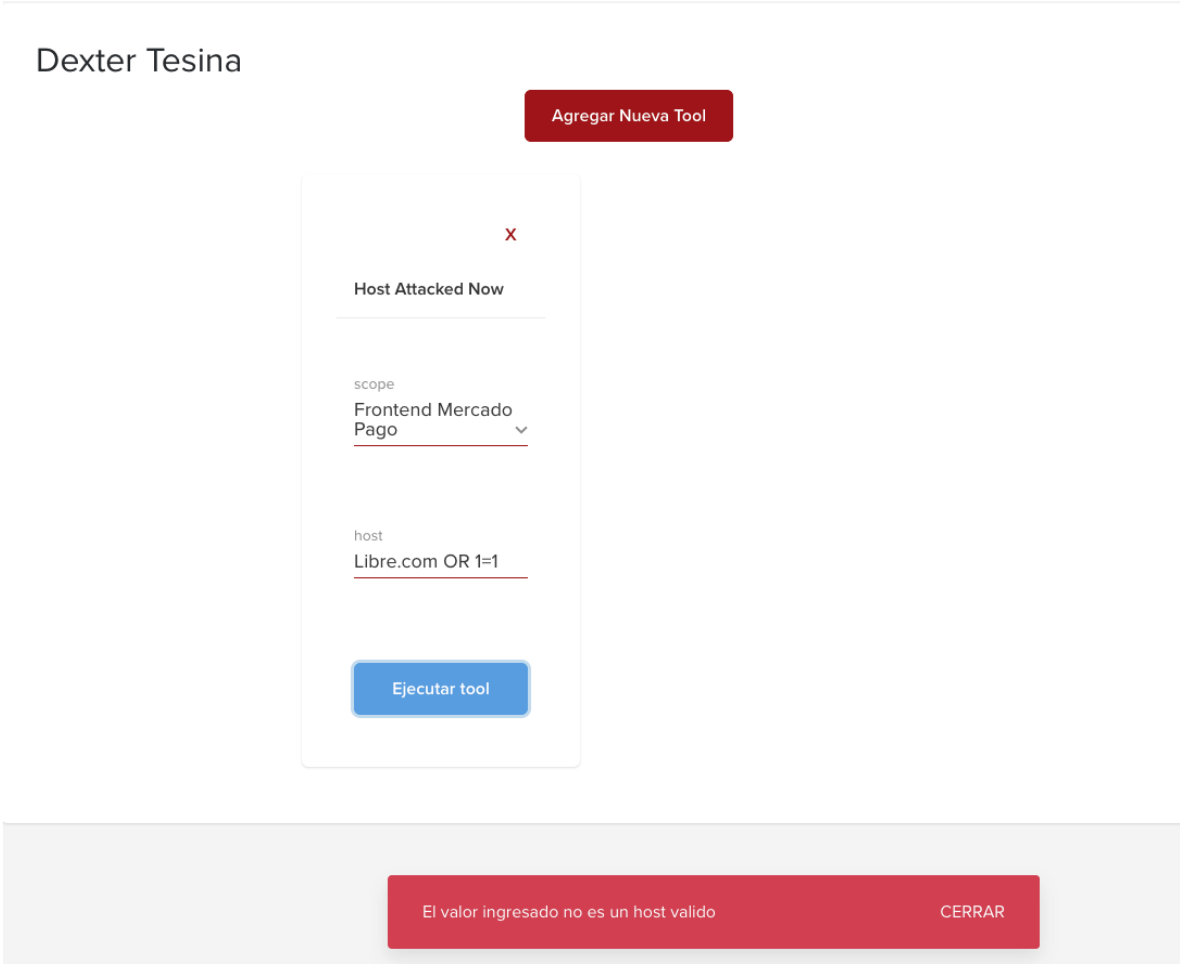

Figura 4.13: Validación de campos.

4.8.3 Agregaciones y enriquecimiento de Logs

Las agregaciones son los logs procesados de forma tal que nos muestre la información útil para el incidente o evento que estamos visualizando.

Podemos definir agregaciones de forma dinámica por cada tool creada, decidiendo así de forma sencilla qué información queremos procesar y mostrar en los resultados.

Por ejemplo, al analizar un ataque de Denegación de servicio del tipo Availability a partir de la tool “Host Attacked”, contamos con las siguientes agregaciones ya pre-configuradas:

- **Top Application**: Top de requests agrupados por nombre de aplicaciones interna.
- **Top By Time:** Top de requests agrupados por minuto.
- **Top Countries:** Top de requests agrupados por países.
- **Top IPs:** Top de requests agrupados por IPs.
- **Top Referers:** Top de requests agrupadas por Header Referer.
- **Top Request Method:** Top de requests agrupadas por los distintos Métodos HTTP.
- **Top Status:** Top de requests agrupadas por los distintos estados HTTP.
- **Top Uris:** Top de requests agrupadas por las urls.
- **Top User Agent:** Top de requests agrupadas por el header HTTP User Agents.
- **Top Users:** Top de requests agrupadas por usuarios de Mercado Libre/Mercado Pago.

Con el objetivo de centralizar la información en Dexter y tener resultados más completos, los logs pasan por un proceso de **enriquecimiento con distintas APIs internas**. Actualmente contamos con los siguientes enriquecedores de datos:

Enrich IP: Es una API interna que enriquece una IP. En nuestro caso enriquecemos la IP origen del request,con la siguiente información:

- Cantidad de usuarios detrás de la IP.
- País.
- Si la ip pertenece a infraestructura de Mercado Libre.
- Si la ip pertenece a un servicio en la nube.
- Puntaje de abuse interno.
- Si la IP pertenece a un País donde opera Mercado Libre.

Users: Es una API interna que enriquece un usuario a partir del user id, o identificador único de usuario dentro de Mercado Libre. En nuestro caso enriquecemos un usuario que realizó requests hacia una aplicación con la siguiente información:

- Email
- Nombre
- Registro
- Restricciones

4.8.4 Ejecución de querys concurrentes.

Al realizar el análisis forense de una vulnerabilidad, existen distintos casos de uso que requieren analizar logs durante un largo periodo de tiempo para detectar si hubo compromiso de datos de usuarios o de la empresa. Teniendo en cuenta el gran volumen de logs de tráfico para una sola aplicación (véase la sección 4.6), el analista podría demorarse mucho tiempo en concluir con su análisis forense. Para intentar reducir este tiempo de análisis, Dexter cuenta con una característica, que se basa en ejecutar consultas concurrentes cuando se busca obtener logs de varios días completos.

Al ejecutar una Tool con un rango de fechas que incluye varios días, Dexter implementa una división por días consultados, y luego ejecuta concurrentemente todas las consultas posibles. De esta forma se ahorra un tiempo considerable a la hora de realizar un análisis forense extenso, permitiendo concluir antes el reporte forense para tomar las acciones necesarias.

4.9 Casos de Éxito

La herramienta la utilizamos actualmente desde el equipo para analizar los siguientes escenarios:

Alertas de aumento de tráfico: Habitualmente recibimos del equipo de Monitoreo alertas de tráfico anómalo sobre una aplicación, varias de estas alertas necesitan ser analizadas cuando no puede determinarse fácilmente si se necesita aplicar un bloqueo de IP, Usuario,etc. Estas alertas además suelen llegar fuera de horario laboral a la guardia, donde se necesita tener una rápida mitigación del evento.
Es por ello que en estos casos Dexter cumple con su propósito realizando un rápido análisis de lo que está sucediendo para tomar una decisión de contención.

Incidentes de tipo Availability o Information Gathering: Cuando realizamos un análisis forense de incidentes del tipo de DDoS, Scanning o Crawlers que ya ocurrieron, necesitamos realizar además del análisis un informe detallando lo ocurrido.
Con las agregaciones de Dexter y los enriquecimientos de información anteriormente mencionados, podemos tener una perspectiva de lo que ocurrió y tomar acciones de mejora continua para que no vuelva a ocurrir.

Vulnerabilidad Web IDOR: La vulnerabilidad de tipo IDOR es una de las más detectadas por el equipo de Application Security.
Desde el equipo de incidentes realizamos el análisis forense de la vulnerabilidad consultando logs de tráfico de varios meses, dependiendo la criticidad del mismo. En caso de detectar que hubo explotación de la vulnerabilidad, se busca obtener el número de datos filtrados a partir de la explotación y tomar las acciones mitigantes.

Con la tool “IDOR Search Tool” podremos realizar este análisis forense en un tiempo ampliamente reducido al que nos ocupaba hacerlo manualmente. Actualmente obtuvimos los primeros casos de éxito al realizar el análisis forense de las vulnerabilidades reportadas durante el 2021.

4.10 Implementación de Dexter en otra Organización

Para implementar una herramienta de análisis forense como Dexter, se consideran algunas de las siguientes tareas como puntos más importantes:

- Definir los incidentes y escenarios más recurrentes, donde sea necesario automatizar un análisis.
    - Ejemplo: En Dexter partimos desde automatizar el análisis de los eventos e incidentes de tipo Availability e Information Gathering. Luego sumamos vulnerabilidades de tipo IDOR. A futuro se van a implementar escenarios específicos de otros tipos de eventos de seguridad.

- Definir las distintas fuentes de logs, tanto las principales, como las que necesitamos para enriquecer la información.
    - Ejemplo: En el caso de Dexter, los logs principales son los logs de tráfico, y luego los cruzamos para enriquecerlos con detalles de cada IPs, usuario,etc.

- Definir los distintos filtros de búsqueda, necesarios para iniciar un análisis.
    - Ejemplo: En Dexter definimos como primeros casos de uso, tener filtros de búsqueda por IP, host, aplicación interna, usuario y User Agent.

- Buscar en la organización y documentar la existencia de APIs para integrarnos que solucionen o mejoren algunos de los escenarios de análisis planteados.
    - Ejemplo: Como vimos en la sección 4.5.2, Dexter consume de la API de Data Extractor, que se encarga de realizar las consultas contra Athena. A su vez contamos con varias APIs internas para enriquecer la información de logs.

- Definir la lógica y procesamiento que queremos agregar a los logs para que ayuden a tomar una decisión sobre el evento o incidente.
    - Ejemplo: Dependiendo del tipo de incidente o escenario que estemos tratando de automatizar, vamos a necesitar realizar distintas lógicas sobre los logs. En los incidentes de tipo Availability nos basamos en el top de usuarios o IPs que realizaron requests, mientras que en un escenario de Data Breach en comportamientos puntuales.

- Definir roles y equipos que usarán la herramienta. De esta forma podemos crear distintos perfiles con diferentes funcionalidades.
    - Ejemplo: Actualmente en Dexter contamos con 2 roles, uno para realizar cualquier tipo de análisis forenses y otro de Administrador que permitirá la visualización y edición de un panel de configuración.

- Definir un servicio de almacenamiento, para preservar los logs del evento analizado como evidencia.
    - Ejemplo: Una vez que se procesen los logs mediante la herramienta, necesitamos almacenar los logs. En Dexter utilizamos Object Storage, un servicio propio de Fury, el cual utiliza el servicio de Amazon S3 para almacenarlos.

- Definir un diagrama de clases escalable,las tecnologías a utilizar, y arquitectura donde se ejecutará la herramienta.
    - Ejemplo: Como vimos en la sección 4.4, optamos por utilizar Python para el backend, React para el Frontend y base de datos SQL. Dexter se ejecuta en la plataforma Fury.

5. Conclusiones

Como conclusiones al presente trabajo quisiera destacar distintos puntos relacionados tanto al desarrollo de la herramienta Dexter como a la gestión de incidentes.

Tal como fuí expresando a lo largo de la tesina, la herramienta Dexter permite solucionar distintas problemáticas recurrentes al analizar un incidente o evento de seguridad. Entre ellas quisiera destacar las siguientes:

Reducción de tiempo de análisis: Distintas características de Dexter, hacen que se haya reducido notablemente el tiempo de análisis de los eventos de tipo Availability, Information Gathering y Vulnerability. Estas características son:

- Interfaz simple que permite que mediante un solo dato obtengamos un resultado de análisis satisfactorio.
- Las agregaciones que nos facilitan la conclusión del análisis forense, nos permiten reducir no solo el tiempo de análisis, sino también costos de consultas realizadas en Amazon.
- El enriquecimiento de información, nos reduce el tiempo de tener que cambiar de una aplicación a otra para poder consultar los datos. Además de tener toda la información obtenida en una misma aplicación.
- Al analizar varios días de logs, Dexter nos divide la consulta por día, y las ejecuta de forma concurrente. De esta forma reduce notoriamente los tiempos de búsqueda de logs en periodos largos.
- El aviso por Slack, nos permite realizar otras tareas concurrentes sobre el evento, sin estar esperando el resultado, mejorando así el tiempo de respuesta

Preservación de evidencias: Almacenamiento de logs de cada caso analizado,de esta manera nos aseguramos que no se vea afectada la evidencia.

Resultados más precisos: Otra de las mejoras que Dexter nos brinda, es poder tener una conclusión más clara y precisa. Esto se debe a las distintas características nombradas, como las agregaciones, enriquecimiento de información, y el poder tener el análisis centrado en un caso de una forma ordenada.

Validación cruzada: Con Dexter podemos compartir el enlace del caso analizado, permitiendo que cualquier miembro del equipo pueda visualizar tanto las ejecuciones realizadas como los resultados obtenidos, y así poder tener una segunda validación del evento o incidente en curso.

Me gustaría realizar una mención especial, en esta sección de conclusiones,a algunos puntos más relevantes respecto a la Gestión de Incidentes.

En mi trayectoria como parte del equipo de Respuesta ante Incidentes, me encontré con diversas problemáticas y escenarios que implicaron una solución o mejora para el desarrollo y maduración del equipo.

Durante los primeros meses de trabajo, recibimos reportes de eventos e incidentes que eran necesarios registrar para su seguimiento. Esta demanda generó la necesidad de contar con una herramienta de registro de incidentes más robusta y escalable (véase la sección 3.6).

Por otro lado, comenzamos a contar tanto con inventarios de logs para realizar análisis, como con protocolos de contención y forense. Los mismos fueron soluciones de gran apoyo para minimizar el tiempo de respuesta o análisis de incidentes.

La construcción de distintos tipos de métricas de eventos e incidentes de seguridad, fue otra gran solución que nos ayudó a detectar, a dar visibilidad de los principales riesgos y la falta de controles de seguridad tanto en las distintas aplicaciones como en la infraestructura de Mercado Libre.

5.5 Trabajo a futuro

La herramienta de análisis forense Dexter, tiene múltiples desafíos como trabajo a futuro para poder transformarse en una herramienta de análisis forense completa. Los puntos de trabajo a futuro :

- **Nuevas integraciones y enriquecimiento de datos:** Uno de los principales objetivos de Dexter, es poder realizar un análisis forense completo sin tener que salir de la herramienta para consultar un dato. Es por ello que uno de los puntos más importantes de trabajo a futuro es continuar integrando Dexter con nuevas APIs, sumando así más fuentes de datos para poder enriquecer un análisis forense.
- **Más escenarios de análisis:** Es necesario contar con la posibilidad de analizar la

mayor cantidad de escenarios posibles. De esta forma cualquier tipo de evento o incidente podrá ser analizado con Dexter en su totalidad.

- **Integración con la herramienta de registro de incidentes:** Uno de los trabajos a futuro es poder integrar la herramienta presentada en la sección 3.6 , de forma tal de tener vinculado el análisis forense de cada evento o incidente registrado.
- **Filtros Dinámicos:** Otro de los puntos como trabajo a futuro es el poder combinar de forma dinámica todos los filtros para realizar búsquedas más precisas, como por ejemplo poder filtrar por una IP, un Host y un usuario específico.
- **Integración con procedimientos forenses y métricas:** Otro de los trabajos a futuro es poder integrar Dexter con los procedimientos definidos en la sección 3.9 y las métricas definidas en la sección 3.8 ,de esta forma al iniciar un nuevo caso, tendríamos el paso a paso de cómo tendríamos que realizar el análisis forense, y cuanto tiempo debería llevar ese análisis para cumplir con las métricas técnicas propuestas, dependiendo del caso que estamos tratando.
- **Migración a arquitectura más escalable:** Como presentamos en la sección 4.6, una de las problemáticas actuales de Dexter, trata de una decisión de arquitectura de tiempo de procesamiento de logs contra costo económico de consultas. Una de las propuestas de trabajo a futuro es la migración a una arquitectura más escalable para solucionar de raíz la problemática de realizar un análisis forense de varios días, o de aplicaciones con mucho tráfico.
- **Publicación de la herramienta como código open source:** Otro de los puntos importantes que se proponen como trabajo a futuro es poder realizar los cambios necesarios para publicar Dexter como código Open Source. De esta forma se podría iniciar una configuración básica dependiendo las necesidades de cada organización, para integrar Dexter con las distintas herramientas de logs disponibles.

Los distintos desafíos a futuro que tenemos en gestión de incidentes son:

- **Correlación de eventos e incidentes**: Poder encontrar relaciones o similitudes entre un evento actual y uno pasado. De esta manera podríamos adelantarnos a un incidente y responder antes de tiempo.
- **Mejora continua de Cálculo de Impacto:** El resultado del cálculo de impacto depende de cientos de variables que van cambiando constantemente, lo cual es un proceso que permanece en mejora continua.
- **Nuevas Métricas**: Necesitamos continuar definiendo métricas nuevas, tanto técnicas como de gestión, de esta forma podremos medir mejor no solo el rendimiento del equipo, sino donde necesitamos mejorar los controles.
- **Nuevos protocolos de respuesta:** Seguir sumando procedimientos y protocolos de respuesta, que no solo ayuda a la madurez del equipo para futuros colaboradores, sino también poder comenzar a automatizar dichos protocolos.

Referencias bibliográficas

- *[1] FIRST.ORG, (2020), "4. CSIRT Services Framework v2.1", First Services Framework (Accedido en Agosto de 2021).*
- *[2] Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). Computer security incident handling guide. NIST (Accedido en Marzo de 2021).*
- *[3] Amazon Web Services, (2021), "Amazon Athena", Consultas de datos al instante | Análisis de datos SQL | Amazon Athena (Accedido en Marzo de 2021).*
- *[4] Amazon Web Services, (2021), "Almacenamiento de datos en la nube S3", Almacenamiento de datos seguro en la nube (S3) (Accedido en Abril de 2021).*
- *[5]Python Software Foundation, (2021), "Python", https://www.python.org/ (Accedido en Abril de 2021).*
- *[6] Facebook Open Source, (2021), "React: Una biblioteca de JavaScript para construir interfaces de usuario", React – Una biblioteca de JavaScript para construir interfaces de usuario (Accedido en Abril de 2021).*
- *[7] Amazon Web Services, (2021), "Amazon Virtual Private Cloud,Cree en una red virtual aislada de forma lógica en la nube de AWS.", Red virtual privada en la nube (VPC) (Accedido en Mayo de 2021).*
- *[8] Amazon Web Services, (2021), "Elastic Load Balancing, Distribuir el tráfico de la red para mejorar la escalabilidad de las aplicaciones", Elastic load balancing para la gestión de tráfico en aplicaciones (Accedido en Mayo de 2021).*
- *[9] Amazon Web Services, (2021), "Auto Scaling groups", https://docs.aws.amazon.com/autoscaling/ec2/ (Accedido en Mayo de 2021).*
- *[10] Amazon Web Services, (2021), "Amazon EC2", https://aws.amazon.com/es/ec2/ (Accedido en Mayo de 2021).*
- *[11] Federico Alliani, Pablo Garbossa (2020), "Mercado Libre: How to Block Malicious Traffic in a Dynamic Environment", https://aws.amazon.com/es/blogs/architecture/mercado-libre-how-to-block-malicious-traffic-in-a-dynamic-environment/ (Accedido en Mayo de 2021).*
- *[12] Mario Pinho, (2020), "AWS Shield Threat Landscape report is now available",AWS Shield Threat Landscape report is now available | Amazon Web Services (Accedido en Mayo de 2021).*
- *[13] OWASP, (2021), "Application Threat Modeling"* Application Threat Modeling | OWASP *(Accedido en Mayo de 2021).*
- *[14] Venkatesh Jagannathan, "Threat Modeling Architecting & Designing with Security in Mind", Advanced Threat Modelling Knowledge Session (Accedido en Mayo de 2021).*

- *[16] Amazon Web Services, (2021), "Precios de Amazon Athena", Precios de Amazon Athena (Accedido en Mayo de 2021).*
- *[17] Mert Hocanin (2017), "Top 10 Performance Tuning Tips for Amazon Athena AWS" (Accedido en Junio de 2021).*
- *[18] Shostack, A. (2014). "Threat modeling: Designing for security" (Accedido en Mayo de 2021)".*
- *[19] ENISA (2018). "Reference Incident Classification Taxonomy" (Accedido en Agosto de 2021).*
- *[20] CISA (2017). "Incident Notification Guideliness" (Accedido en Agosto de 2021).*

Índice de Figuras