UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Evolución del sistema de gestión de incidentes de seguridad orientado a CSIRT de la UNLP - Ngen

AUTORES: Damián Rubio

DIRECTOR/A: Lic. Einar Lanfranco

CODIRECTOR/A: Mtr. Paula Venosa

CARRERA: Licenciatura en Sistemas

Resumen

Esta tesina detalla el desarrollo de un sistema de software de infraestructura programable y configurable llamado Ngen, capaz de brindar soporte a la gestión de incidentes de seguridad en el ámbito de trabajo de un Computer Security Incident Response Team (CSIRT) .

Palabras Clave

Ngen, Gestión de incidentes, CERT, CSIRT, ciberseguridad

Conclusiones

Ngen surge como un reflejo de las necesidades de nuestro contexto laboral y profesional en CERT UNLP.
No solo soluciona muchas de las tareas diarias sino que permite tener una visión amplia de los eventos de nuestra comunidad que lo diferencia de otros sistemas actuales.
Además, mejora el bienestar laboral gracias a la automatización de varias tareas cotidianas y repetitivas, que nos permiten realizar otras tareas más complejas en paralelo.

Trabajos Realizados

Se desarrolló un sistema web de gestión de incidentes capaz de abarcar la mayoría de los requisitos de un CSIRT. Aplicando, no solo experiencias y requerimientos propios, sino también, requisitos generados a partir de experiencias externas con entidades gubernamentales. Simplificando el trabajo diario de los operadores, automatizando el ciclo de vida de los incidentes y permitiendo una comunicación óptima con la comunidad a la que pertenece.

Trabajos Futuros

Aplicar cifrado de emails para cualquier comunicación sensible.
Generar más modularidad en el código permitiendo configurar a Ngen para activar y desactivar módulos bajo demanda.
Integración con sistemas CMBD muy utilizados por grandes organizaciones para mantener actualizada su estructura de activos y sus responsables .
Conexión entre instancias de Ngen que puedan compartir información.

Fecha de la presentación: Abril 2023

Evolución del sistema de gestión de incidentes de seguridad orientado a CSIRT de la UNLP - Ngen.

Por

Damián Rubio

Facultad de Informática
Universidad Nacional de La Plata

Directores: Einar Lanfranco. Paula Venosa

Tesina de Licenciatura en Sistemas

2023

Dedicación

Dedicado a mi madre que, con su amor, guía mis sueños.
Gracias.

Agradecimientos

A mis padres por ser mi inspiración y apoyarme en toda mi carrera.

A los integrantes y creadores del CERT UNLP, Paula Venosa, Einar Lanfranco y Nicolás Macia, que fueron parte fundamental de mi formación tanto profesional como académica. Y por haber brindado sus conocimientos y, sobretodo, su amistad.

A Lía Molinari, mi primera tutora y mi pie inicial en la seguridad informática.

Y por último, a la Universidad y la facultad por permitirme realizarme profesionalmente.

Tabla de Contenidos

Lista de Figuras

Lista de Tablas

CAPÍTULO 1

INTRODUCCIÓN

1.1 Objetivo

El objetivo principal de esta tesina es el desarrollo de un sistema de software de infraestructura programable y configurable, capaz de brindar soporte a la gestión de incidentes de seguridad, en el ámbito de trabajo de un Equipo de Respuesta a Incidentes de Seguridad (CERT[1] o CSIRT[2]).

Será un sistema web que represente una evolución a partir del sistema actual mediante no sólo la utilización de nuevas tecnologías sino también aplicando las lecciones aprendidas después de varios años de uso.

Este desarrollo deberá cumplir con las características de ser integrable con otros componentes de software utilizados en la comunidad de CSIRTs. Con ese objetivo en vista se deben poder integrar fuentes de información que aportan datos de posibles incidentes, incorporar los enriquecedores de información que sirven para mejorar la base de conocimiento de un incidente y facilitar el intercambio de información con otros grupos.

Para ello recibirá incidentes tanto por vía manual, mediante la interfaz web, como mediante la presentación de una API para ser utilizada por otros sistemas, y se encargará del reporte automático y personalizado a los involucrados en la investigación, permitiendo el seguimiento del incidente y dando lugar a la producción de estadísticas utilizables en

[1] CERT: Computer Emergency Response Team
[2] CSIRT: Computer Security Incident Response Team

un análisis posterior para identificar, definir e implementar mecanismos de mejora.

1.2 Motivación

La gestión de incidentes es la tarea primordial de un CSIRT [3], siendo fundamental contar con mecanismos de clasificación, documentación y seguimiento de los mismos.

Resulta que estas tareas son, en su mayoría, ejecutadas de manera manual o a partir de programas independientes que los mismos analistas desarrollan para solucionar las distintas etapas de la atención de incidentes. Si bien cada CSIRT tiene su manera particular de gestión, la falta de automatización termina entorpeciendo, o impidiendo, el proceso de gestión.

Existen varios sistemas que intentan realizar esta tarea, entre ellos podemos mencionar algunos que son históricos en la comunidad, generalmente basados en sistemas de tickets [4] como RTIR [1].

Si bien la mayoría de ellos cumplen con una parte de la gestión de incidentes, como son poder documentar y dar seguimiento a los incidentes, no cumplen con muchos de otros requisitos funcionales asociados a tareas que deben ser realizadas diariamente de forma manual por los usuarios de estos sistemas, tales como:

- Unificar incidentes (agregación).
- Reconocer eventos repetidos actuales y/o histéricos.
- Incluir metadatos (geolocalización, DNS[5], RDAP [6], etc.).
- Reportar incidentes al responsable automáticamente.
- Generar evidencias y adjuntarlas a los reportes.
- Manejar niveles de confidencialidad.

[3] CSIRT: Computer Security Incident Response Team o Equipo de Respuesta ante Incidencias de Seguridad Informáticas

[4] Un sistema de tickets es una plataforma de software diseñada para administrar y rastrear las solicitudes de atención al cliente. Agiliza el proceso de resolución de problemas de los clientes, lo que facilita que las empresas brinden un soporte rápido y efectivo. Con un sistema de tickets, los clientes pueden enviar solicitudes. Luego, las solicitudes se organizan y priorizan, lo que permite que los equipos de soporte respondan de manera rápida y eficiente.

[5] DNS: Domain Name System o sistema de nombres de dominio, un servicio de Internet que traduce los nombres de dominio a direcciones IP.

[6] RDAP: Registration Data Access Protocol

- Manejar Taxonomías según estándares.
- Manejar Playbooks [7] para la resolución de incidentes.
- Integrar desarrollos ampliamente adoptados por los CSIRTs a nivel mundial como por ejemplo MISP [2] o IntelMQ [3].

A diferencia de los sistemas de tickets tradicionales, un sistema de gestión automatizado de incidentes debe poder integrarse varios canales de información y permitir la administración de grandes volúmenes de datos, facilitando tanto el enriquecimiento como el análisis estadístico.

Otros sistemas más nuevos, como TheHive Project[4], que es uno de los más utilizados actualmente. Cumple con varios de los requisitos nombrados pero no completamente. Una de las grandes falencias de este, y otros sistemas más actuales, es la falta de comunicación y documentación de la comunidad a la que pertenecen. Carecen de este tipo de visión que es fundamental para la clara y rápida comunicación con su comunidad.

La falta de un sistema que realmente pudiese cumplir todos estos requisitos fue la motivación principal para desarrollar el sistema Ngen. En ese momento mi actividad profesional se centraba en el ámbito del CERTUNLP [5], el CSIRT de la Universidad Nacional de La Plata, y por ello conozco en detalle la problemática de la gestión de incidentes y las limitaciones del producto que utilizábamos.

Tomé como motivación adicional que en mi ámbito laboral tenía la posibilidad no sólo de desarrollar el producto, sino también de ponerlo en funcionamiento en el ambiente de producción, llegando a que sea superador a el que se utilizaba anteriormente.

1.3 Estructura de la Tesina

A continuación se detalla de forma concisa la estructura y los contenidos de cada capítulo.

1.3.1 Capítulo 1: Introducción

En este primer capítulo se detalla el objetivo de esta tesis junto con la motivación por la cual se realiza.

[7] Playbook: procedimientos escritos que sirven como guías paso a paso a los analistas de los CSIRT en la gestión de incidentes.

1.3.2 Capítulo 2: Estado del arte

En este capítulo se desarrollan los conceptos de CSIRT y sus servicios, además, conceptos de "constituency", eventos e incidentes.

Dentro de los servicios de un CSIRT se detalla, en particular, la gestión de incidentes. Luego se analizan los sistemas actuales de gestión de incidentes según 6 criterios: Clasificación, gestión, anuncio, retroalimentación, interacciones y automatización de tareas.

1.3.3 Capítulo 3: Solución propuesta

Este capítulo está dedicado al desarrollo de los módulos que componen el sistema de gestión Ngen. Se concluye con una comparación entre los sistemas anteriormente explicados y Ngen.

1.3.4 Capítulo 4: Implementación y resultados

Se detalla la puesta en funcionamiento de Ngen junto a sus casos de éxito.

1.3.5 Capítulo 5: Conclusiones y Trabajo futuro

En este último capítulo se presentan las conclusiones del trabajo y las proyecciones futuras.

CAPÍTULO 2

ESTADO DEL ARTE

2.1 CSIRT

Según "State of the Practice of CSIRTs" [6], un Equipo de Respuesta a Incidentes de Seguridad (CERT, del inglés Computer Emergency Response Team) es una organización responsable de recibir, revisar, responder a informes y actividad sobre incidentes de seguridad. Sus servicios son generalmente prestados para un área de cobertura definida que podría ser una entidad relacionada u organización de la cual dependen, una corporación, una organización de gobierno o educativa; una región o país, una red de investigación; o un servicio pago para un cliente.

El término CSIRT (Computer Security Incident Response Team) o Equipo de Respuesta ante Incidencias de Seguridad Informáticas, sirve también para referirse al mismo concepto. De hecho el término CSIRT es el que se suele usar en Europa en lugar del término protegido CERT, que está registrado en EE. UU. por CERT Coordination Center (CERT/CC).

Un CSIRT suele funcionar de manera reactiva recibiendo informes de amenazas, ataques, escaneos, uso indebido de recursos, acceso no autorizado de datos o activos de información. Analizando los informes se determina que está sucediendo y el curso de acción a seguir para mitigar la situación y resolver el problema. También puede desempeñar un papel proactivo, esto puede incluir proporcionar concientización, capacitación en seguridad, consultoría de seguridad, mantenimiento de configuración, producción de documentos técnicos y avisos [6].

Aunque comúnmente vemos referidos a estos equipos como "equipo de respuesta ante incidentes", el término "respuesta ante incidentes" o "incident response" solo se relaciona con uno de los servicios que se prestan en un CSIRT. En cambio, el término "gestión de incidentes" describe mucho más ampliamente las actividades que muchos CSIRT realizan en sus operaciones diarias [6].

2.1.1 Eventos e Incidentes

Quizás una manera más genérica de definir el concepto de incidente puede ser definiendo evento y evento adverso.

Un evento es cualquier ocurrencia observable en un sistema o red. Los eventos incluyen un usuario que se conecta a un recurso compartido de archivos, un servidor que recibe una solicitud de una página web, un usuario que envía un correo electrónico y un firewall que bloquea un intento de conexión.

Un evento adverso es un evento con una consecuencia negativa, como fallas del sistema, inundaciones de paquetes, uso no autorizado de privilegios del sistema, acceso no autorizado a datos confidenciales y ejecución de malware que destruye datos.

Podríamos entonces definir incidente como un evento adverso que afecta los activos de una organización y que puede comprometer la confidencialidad, integridad y/o disponibilidad de la información.

En particular, la definición real del término "incidente" varía de un equipo a otro. Por ejemplo [6]:

- Cualquier evento adverso real o presunto en relación con la seguridad de los sistemas informáticos o redes informáticas.
- El acto de violar una política de seguridad explícita o implícita.
- Una colección de datos que representan uno o más ataques relacionados.
- Cualquier evento irregular o adverso, que puede ser electrónico, físico o social que ocurre en cualquier parte de la infraestructura del estado.
- Un ataque contra una computadora o red, ya sea real o percibido.
- Un evento adverso en un sistema de información y/o red, o la amenaza de que ocurra tal evento.
- Una situación en la que la información de una entidad está en riesgo.

- Eventos que interrumpen el procedimiento operativo normal y precipitan algún nivel de crisis.
- Un grupo de ataques que se pueden distinguir de otros incidentes debido al carácter distintivo de los atacantes y al grado de similitud de sitios, técnicas y tiempos.
- Eventos adversos que amenazan la seguridad en los sistemas y redes informáticos.
- Cualquier evento adverso por el cual algún aspecto de la seguridad informática pudiera verse amenazado.
- Evento del sistema relevante para la seguridad en el que se desobedece o se infringe la política de seguridad del sistema.

Esta variedad de definiciones también complica la comparación de estadísticas entre organizaciones.

Aunque existen muchas definiciones del término "incidente" también existen algunas similitudes. La definición de "incidente" se relaciona con algún tipo de actividad no autorizada contra una computadora o red que resulta en una violación de una política de seguridad. Ya sea que se trate de una acción, un evento, una situación o la recopilación de datos relacionados con un ataque, todos generalmente están de acuerdo en que el CSIRT debe identificar la amenaza y luego tomar la acción adecuada, según la orientación definida en las políticas y procedimientos del equipo. Esto generalmente se conoce como "respuesta a incidentes".

Uno de los beneficios de tener una capacidad de respuesta a incidentes es que permite responder de manera sistemática, siguiendo una metodología consistente, para que se tomen las acciones apropiadas. Ayudando a minimizar la pérdida, o el robo de información, y la interrupción de los servicios.

Otro beneficio es la capacidad de utilizar la información obtenida durante la gestión de incidentes para prepararse mejor ante problemas futuros y brindar una protección más sólida [7].

Aun así, el concepto de incidente puede seguir siendo ambiguo. Algunos CSIRTs consideran algunas vulnerabilidades como incidentes. Por ejemplo, un puerto o servicio abierto, como un DNS [1] o NTP [2], se considera como incidente cuando se detecta de manera proactiva.

[1]DNS: Domain Name System o sistema de nombres de dominio, un servicio de Internet que traduce los nombres de dominio a direcciones IP.

[2]NTP: Network Time Protocol, protocolo de Internet para sincronizar los relojes de los sistemas informáticos.

2.1.2 Constituency

Aunque la traducción literal es "distrito electoral" o "circunscripción electoral" en el contexto de CSIRTs, "contituency" se refiere a "Comunidad de la que el CERT/CSIRT es responsable y a la que ofrece sus servicios".

La comunidad generalmente se refiere a las personas u organizaciones que son atendidas por el CSIRT. Estos miembros constituyentes comparten algún tipo de características específicas (red, sector, ubicación, agencia, etc.) y se identifican como empleados, clientes, suscriptores o incluso consumidores de información.

La comunidad en sí puede ser una serie de entidades diferentes, como departamentos individuales dentro de una organización, una universidad, una empresa, agencia gubernamental o militar, corporaciones nacionales o internacionales, proveedores de servicios o estados nacionales. Independientemente de a quien sirva, el CSIRT debe identificar claramente la comunidad para asegurarse de que brinda servicios a las personas adecuadas [6].

Por ejemplo, la comunidad del CSIRT de la Universidad Nacional de La Plata puede ser definida como: *la comunidad de la "Universidad Nacional de La Plata", el dominio y subdominios de "unlp.edu.ar" y el bloque IP "163.10.0.0/16"*.

Según "Handbook for Computer Security Incident Response Teams" [8], la gama de servicios ofrecidos por un CSIRT y la naturaleza de esos servicios, pueden tener la necesidad de definir más de una comunidad. Estas múltiples comunidades pueden cruzarse, ser subconjuntos o estar totalmente separadas de otras atendidas por ese CSIRT. Es por ello que puede haber complicaciones al momento de definir la comunidad en casos donde la misma es muy grande o dinámica (que cambie cuando los clientes vienen y van).

Incluso en el caso de un CSIRT con una comunidad muy delimitada, lo más probable es que tenga que tratar con información asociada, o proveniente, de partes que no pertenecen a su comunidad. Por ejemplo, un CSIRT que proporcione un servicio de gestión de incidentes a una comunidad delimitada sin duda deseará aceptar informes de incidentes que la afecten directamente desde fuera de sus límites, manejar adecuadamente esa información, asegurarse de que llegue a los puntos de contacto apropiados y sea coordinado dentro de la misma.

Muchos CSIRTs actúan como un punto de coordinación entre su comunidad y otras partes externas (como otros CSIRT, administradores de sistemas, proveedores, fuerzas del orden, asesores legales o medios de comunicación). Estas interacciones pueden variar desde la simple transmisión de solicitudes hasta el intercambio completo de datos y la

cooperación total.

2.1.3 Servicios

Cada CSIRT es diferente y brinda servicios basados en la misión, el propósito y la comunidad. Algunos de los servicios ofrecidos se relacionan directamente con la gestión de incidentes, la cual es un servicio central de un CSIRT. Otros servicios, como la capacitación en seguridad o las auditorías, solo se relacionan indirectamente con la gestión de incidentes, mientras atienden necesidades de seguridad organizacionales más amplias. Algunos servicios pueden ser proporcionados por otras partes de la organización, como un departamento de Seguridad de la Información o algún departamento encargado de la capacitación, en lugar del CSIRT, o incluso pueden ser subcontratados. La asignación real de tareas y responsabilidades depende de la estructura de la organización a la cual pertenece el CSIRT [6].

Los servicios de un CSIRT pueden ser categorizados de la siguiente forma [6]:

- **Servicios reactivos:** estos servicios pueden iniciarse mediante notificación de terceros o mediante la visualización de registros y alertas del sistema de detección de intrusiones o monitoreo (IDS[3]). Los servicios reactivos son el componente central del trabajo del CSIRT.

- **Servicios proactivos:** estos servicios brindan asistencia e información para ayudar a preparar, proteger y asegurar los sistemas. Están diseñados para mejorar la infraestructura y los procesos de seguridad de la comunidad antes de que ocurra o se detecte cualquier incidente o evento. Los principales objetivos son evitar incidentes y reducir su impacto y alcance cuando ocurren.

- **Servicios de gestión de la calidad de la seguridad:** los servicios que entran en esta categoría no son exclusivos de la gestión de incidentes o del CSIRT en particular. Son servicios bien conocidos y establecidos diseñados para mejorar la seguridad general de una organización. Al aprovechar las experiencias adquiridas en la prestación de los servicios reactivos y proactivos descritos anteriormente, un CSIRT puede aportar perspectivas únicas a estos servicios de gestión de la calidad que de otro modo no estarían disponibles. Estos servicios están diseñados para incorporar comentarios y

[3]IDS: Intrusion Detection System

lecciones aprendidas en función del conocimiento adquirido al responder a incidentes, vulnerabilidades y ataques.

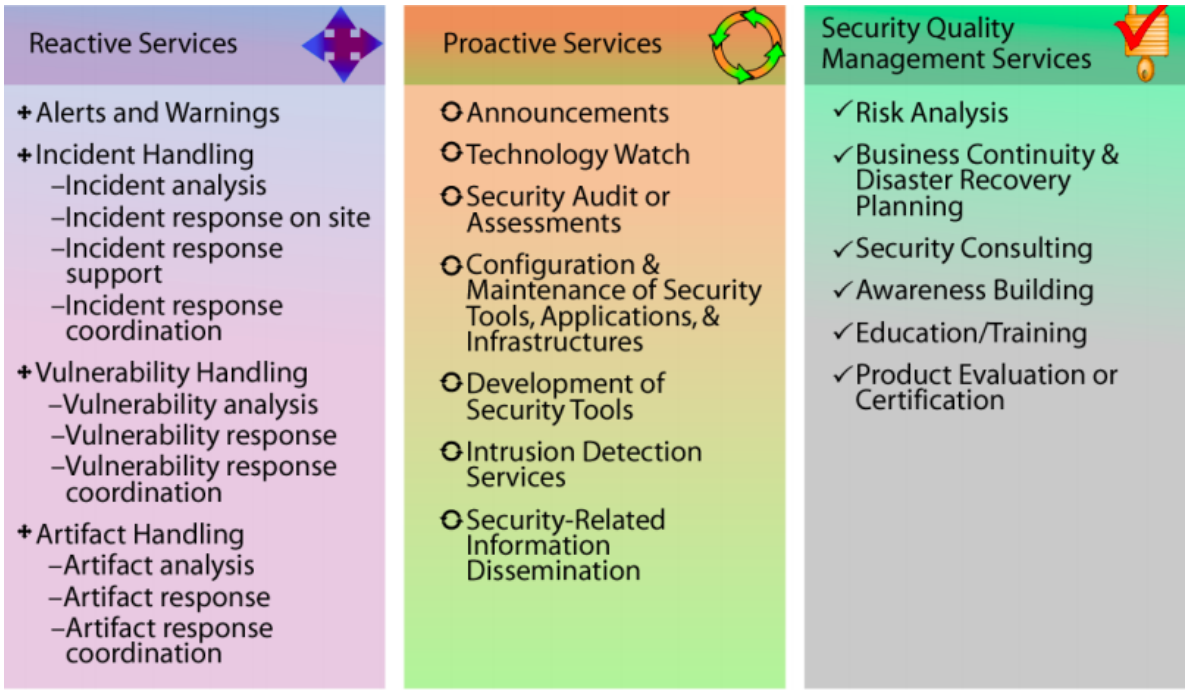

Figura 2.1: Servicios de un CSIRT por categoría [6]

Este tipo de categorización, aunque útil, es anticuada. Una categorización mas detallada, y actual, es la propuesta por el FIRST [4] [9].

En su documento "FIRST CSIRT Services Framework v2.1" [10] el FIRST propone una estructura de servicios englobados en "Ámbitos de servicios" que los agrupan por aspectos en común.

Además, cada servicio se compone de funciones y estas de subfunciones.

ÁMBITOS DE SERVICIO– > SERVICIOS– > FUNCIONES– > SUBFUNCIONES

Al igual que en los servicios anteriores, no se espera que se presten todos y cada uno de ellos. Cada equipo tendrá que elegir los servicios necesarios para su contexto.

[4] FIRST: Forum of Incident Response and Security Teams

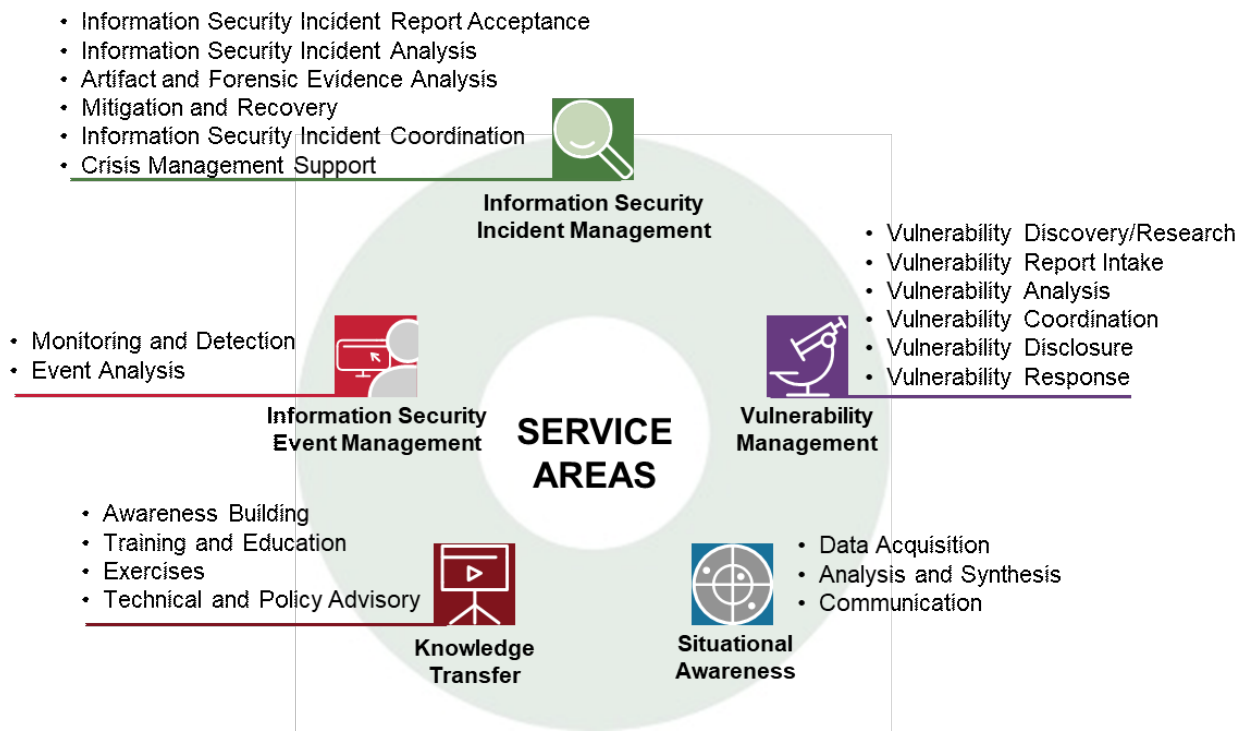

Figura 2.2: Servicios y ámbitos de servicios FIRST [10]

A continuación se detallan los ámbitos anteriormente nombrados [10], en especial "Gestión de eventos de seguridad" y "Gestión de incidentes de seguridad". Estos ámbitos están directamente relacionados con las operaciones realizadas por un CSIRT y, por ende, a la funcionalidad esperada de los sistemas de gestión de incidentes.

2.1.3.1 Gestión de eventos de seguridad

La gestión de eventos de seguridad tiene por objeto identificar los incidentes de seguridad a partir de la correlación y el análisis de los eventos de seguridad provenientes de diversas fuentes de datos.

Supervisión y detección: poner en marcha un procesamiento automatizado y continuo de muy diversas fuentes de información y datos contextuales a fin de identificar posibles incidentes de seguridad, como ataques, intrusiones, filtración de datos o infracciones de la política de seguridad.

Análisis de eventos: seleccionar y clasificar posibles incidentes de seguridad detectados para identificar, clasificar y priorizar verdaderos positivos. Además de

identificar eventos directamente relacionados con incidentes de seguridad posibles o en curso.

2.1.3.2 Gestión de incidentes de seguridad

Este ámbito de servicio constituye el núcleo del equipo de seguridad y consiste en servicios que son esenciales para ayudar a la comunidad durante un ataque o incidente. Los equipos de seguridad deben estar preparados para ayudar y apoyar. Gracias a esta posición y experiencia únicas, son capaces no sólo de recopilar y evaluar informes de incidentes de seguridad, sino también de analizar los datos relevantes y realizar un análisis técnico detallado del propio incidente y de cualquier dispositivo utilizado. A partir de este análisis, se pueden recomendar medidas de mitigación y de recuperación del incidente, y se ayudará a los responsables a aplicar las recomendaciones.

Aceptación del informe de incidentes de seguridad: recibir y procesar informes de posibles incidentes de seguridad remitidos por la comunidad, los servicios de gestión de eventos de seguridad o por terceros.

Análisis de incidentes de seguridad: analizar y comprender mejor los incidentes de seguridad confirmados.

- Clasificación de incidentes de seguridad (prioridades y clasificación): clasificar, priorizar y crear una evaluación inicial del incidente de seguridad.

- Recopilación de información: admitir, catalogar, almacenar y rastrear información relativa al incidente de seguridad y a todos los eventos de seguridad que se consideren parte del mismo.

- Coordinación de análisis detallado: iniciar y rastrear cualquier otro análisis técnico sobre el incidente de seguridad.

Análisis de los artefactos y de pruebas forenses: analizar y comprender los artefactos[5] relacionados con un incidente de seguridad confirmado, tomando en cuenta de la necesidad de preservar las pruebas forenses.

[5]En informática se suelen llamar artefactos a los objetos creados a partir de tecnologías digitales

- Correlación entre incidentes: permitir el uso de toda la información disponible para comprender mejor el contexto y detectar interrelaciones que de otro modo no se habrían reconocido ni habrían permitido actuar.

- Análisis de medios o de superficie: comparar la información recabada a partir de los artefactos con otros actores públicos y privados y/o repositorios autorizados.

- Análisis comparativo: realizar un análisis centrado en la identificación de la funcionalidad o intención común, en particular el análisis de la semejanza con los artefactos catalogados.

Mitigación y recuperación: contener el incidente de seguridad en la medida de lo posible para limitar el número de víctimas, reducir las pérdidas y recuperarse de los daños, evitar nuevos ataques y nuevas pérdidas mediante la eliminación de las vulnerabilidades o puntos débiles explotados y mejorar la seguridad cibernética en general.

Coordinación de incidentes de seguridad: garantizar notificación oportuna y la distribución de información exacta; mantener el flujo de información y rastrear la situación de las actividades de las entidades encargadas o a las que se les encomiende participar en la respuesta al incidente de seguridad; y asegurarse de que el plan de respuesta se ejecuta y que las divergencias causadas tanto por las demoras como por la nueva información se gestionan en consecuencia.

- Comunicación: colaborar eficazmente con los interesados y establecer múltiples canales de comunicación adecuados con la confidencialidad necesaria.

- Distribución de notificaciones: alertar a las entidades afectadas por el incidente de seguridad o a las que puedan contribuir a la respuesta al mismo y proporcionarles la información necesaria para que comprendan su función y las posibles expectativas de su cooperación y apoyo.

- Distribución de información pertinente: mantener la comunicación con las entidades identificadas y proporcionar un adecuado flujo disponible a fin de que esas entidades puedan beneficiarse de los conocimientos disponibles y de las lecciones extraídas, aplicar respuestas mejoradas o adoptar nuevas medidas ad hoc [6].

[6] ad hoc: Que está hecho especialmente para un fin determinado o pensado para una situación concreta.

- Coordinación de actividades: hacer un seguimiento de la situación de todas las comunicaciones y actividades.
- Notificación: garantizar que todas las entidades implicadas en la empresa dispongan de información sobre el estado de las actividades en curso, de modo que al decidir sobre la forma de proceder conozcan lo mejor posible la situación.
- Comunicación con los medios: colaborar con los medios de comunicación (públicos) para poder proporcionar información empírica precisa y fácil de entender sobre los eventos en curso, a fin de evitar la propagación de rumores e información engañosa.

Ayuda en la gestión de la crisis: proporcionar conocimientos y contactos a otros expertos en seguridad, equipos de seguridad y comunidades para ayudar a mitigar la crisis.

2.1.4 Gestión de vulnerabilidades

El ámbito de servicios de gestión de vulnerabilidades comprende servicios relacionados con el descubrimiento, el análisis y el tratamiento de vulnerabilidades de seguridad nuevas o notificadas en los sistemas de información. Este ámbito también incluye servicios relacionados con la detección de vulnerabilidades conocidas y la respuesta a las mismas a fin de evitar que sean explotadas. Por consiguiente, este ámbito de servicios abarca los servicios relacionados con las vulnerabilidades nuevas y conocidas.

2.1.5 Conciencia coyuntural

Por conciencia coyuntural se entiende la capacidad de identificar, procesar, comprender y comunicar los aspectos esenciales de lo que está sucediendo en el ámbito de responsabilidad del equipo de seguridad y que pueda afectar a las actividades o a la función de sus responsables. Este ámbito de servicio incluye determinar la forma de recabar información pertinente de diferentes fuentes, cómo integrar esa información y cómo difundirla de manera oportuna para que los responsables puedan adoptar decisiones adecuadas.

2.1.6 Transferencia de conocimientos

Por la naturaleza de sus servicios, los equipos de seguridad están en una posición única para recopilar datos pertinentes, realizar análisis detallados e identificar amenazas,

tendencias y riesgos, así como para crear las prácticas idóneas operativas actuales que ayuden a las organizaciones a detectar, prevenir y responder a los incidentes de seguridad. La transferencia de estos conocimientos a sus responsables es fundamental para mejorar la seguridad cibernética en general.

2.2 Servicio de gestión de incidentes

En esta sección busca profundizar los conceptos planteados por el ámbito de "Gestión de incidentes de seguridad" y sus funciones. Detallando cuatro funciones generales definidas en "Handbook for Computer Security Incident Response Teams" [8]: **clasificación**, **gestión**, **anuncio** y **retroalimentación**.

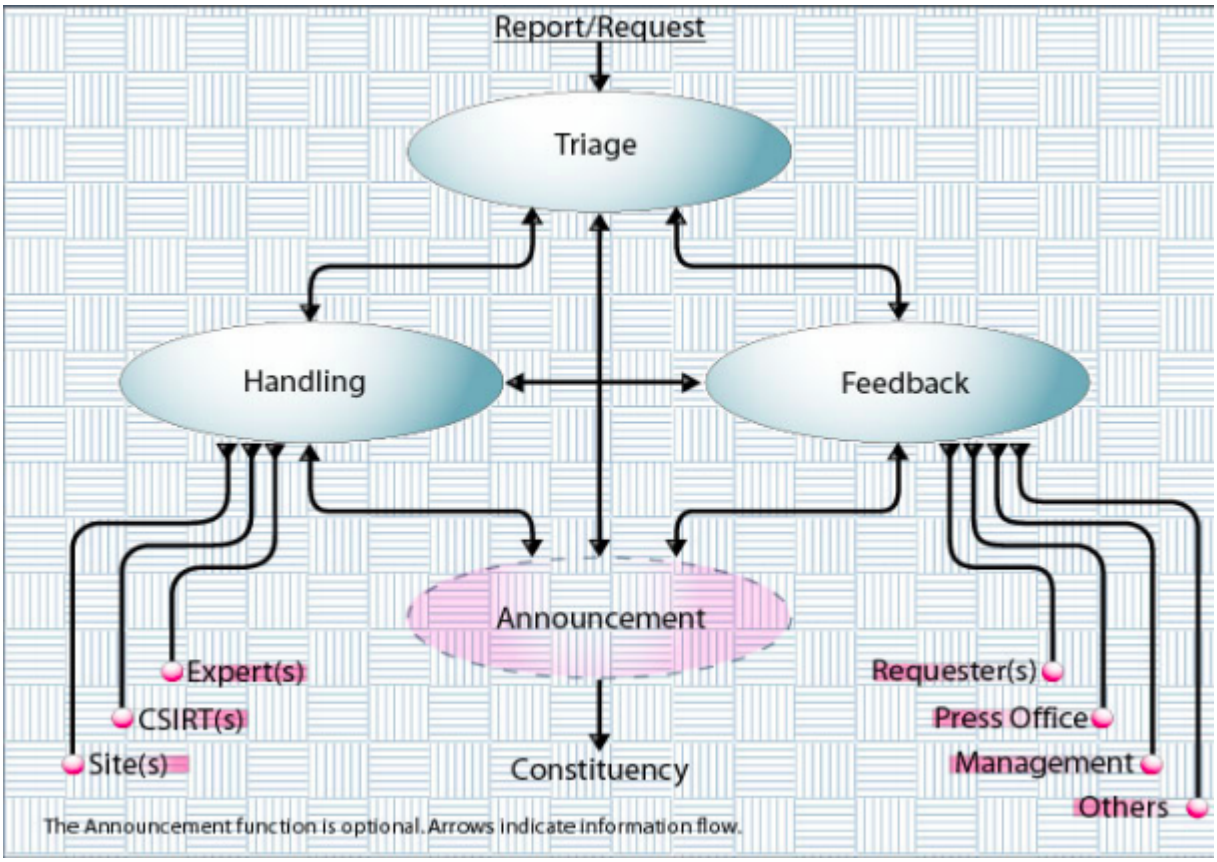

Figura 2.3: Funciones del servicio de gestión de incidentes [8]

2.2.1 Clasificación

La función de clasificación proporciona un único punto de contacto y el punto focal para aceptar, recopilar, clasificar, ordenar y transmitir la información entrante al servicio.

El objetivo de esta función es garantizar que toda la información destinada al servicio de gestión de incidentes se canalice a través de un único punto focal, independientemente del método por el que llegue (por ejemplo, por correo electrónico o teléfono) para su redistribución y gestión dentro del mismo. Este objetivo se logra comúnmente anunciando la función de clasificación como el único punto de contacto para todo el servicio de gestión de incidentes.

Dado que este es un requisito común en muchos servicios de CSIRT, los equipos suelen anunciar un único punto de contacto para todo el CSIRT; independientemente del servicio requerido, se proporciona una única función de clasificación para todos los servicios que ofrece el CSIRT. Este medio de contacto suele ser el email de "abuse" o "report", por ejemplo "abuse@cert.unlp.edu.ar". La manera en que cada CSIRT publica su punto de contacto suele variar, pero la más eficiente es documentar la comunidad con RDAP [7][11] y WHOIS [12].

Una vez que la información es recibida por la función de clasificación, se envía un aviso de recibimiento, luego la información se ordenará, priorizará, rastreará y pasará a otras funciones dentro del servicio.

2.2.1.1 Relación de eventos

Basándose en la información y los datos con respecto a los eventos de servicio existentes, se llevará a cabo una clasificación inicial para identificar qué función del servicio de gestión de incidentes debe manejar la información. El siguiente paso es determinar si la información está directamente relacionada con algún evento actual o pasado. Si está directamente relacionado con algún evento existente o previamente rastreado, se etiquetará como parte de ese evento. De lo contrario, se registrará como un evento nuevo de un tipo determinado y se etiquetará adecuadamente. Además de ser clasificada y etiquetada, la función de clasificación asigna comúnmente una prioridad inicial a la información de acuerdo con el esquema de prioridad en uso por las funciones dentro del servicio.

Las herramientas para ingresar, acceder, rastrear información y eventos pueden facilitar y semi-automatizar la manipulación de datos y búsquedas. Estas herramientas

[7]RDAP: Registration Data Access Protocol

pueden ayudar al personal responsable de la clasificación al ayudar a establecer la identificación de:

- nuevos eventos (incidentes, solicitudes, informes de vulnerabilidad, otros avisos de información).
- información directamente relacionada con los eventos actualmente rastreados.
- información directamente relacionada con un evento previamente cerrado.
- eventos que se rastrean por separado, pero que pueden tener una relación directa.
- información que se considera fuera del alcance del servicio de gestión de incidentes.

Si la información contiene detalles insuficientes o está incompleta, es probable que la función de clasificación se vuelva lenta, imprecisa o incapaz de cumplir su función. En tales casos, puede ser necesario buscar información más detallada del remitente antes de que la información pueda ser clasificada adecuadamente, lo que retrasa el proceso.

2.2.1.2 Uso de números de seguimiento

Si un equipo utiliza un esquema de número de seguimiento y puede alentar o exigir a otros que utilicen los números asignados en toda la correspondencia, esto facilitará enormemente el proceso de clasificación.

Para facilitar el soporte automatizado, el esquema de numeración debe proporcionar identificadores simples para el reconocimiento de personas y herramientas. En un sistema de seguimiento sólido, los números de seguimiento son las "etiquetas" que utiliza el sistema para clasificar automáticamente la información entrante, almacenarla y correlacionar esta misma con otra actividad, preferentemente sin intervención humana. Esto agiliza el proceso y permite que la función de clasificación se concentre más intensamente en la correlación correcta de la información sin etiquetar.

Los números de seguimiento se pueden utilizar fácilmente en el asunto de los mensajes de correo electrónico, documentarlos en las portadas de documentos y especificarlos en los mensajes.

Debido a que los números de seguimiento se utilizan en las comunicaciones externas, deben considerarse como información pública y, por lo tanto, no deben revelar información confidencial, como los nombres de los hosts o dominios involucrados. Otra información sensible que se debe evitar en un esquema de seguimiento incluye información que indicaría el número, la naturaleza o el alcance de los eventos reportados. Por estas

razones, el uso de algún esquema de generación de números aleatorios es una mejor práctica.

2.2.2 Gestión

La función de gestión proporciona apoyo y orientación relacionados con los incidentes. Se realiza una revisión del informe para determinar qué ocurrió y/o el tipo específico de actividad involucrada. El análisis del informe puede incluir la revisión de pruebas y materiales de respaldo para identificar quién está involucrado, quien necesita ser contactado y la asistencia que se solicita o que se brinda. El equipo deberá identificar las respuestas apropiadas y la notificación o seguimiento a realizar.

2.2.2.1 Ciclo de vida de un incidente

Cualquiera sea la definición de incidente de un equipo, probablemente se ajustará al ciclo de vida descrito en esta sección.

Como se menciona en la función de clasificación, parte del ciclo de vida de un incidente tiene lugar dentro de esta función, donde un incidente puede categorizarse inicialmente, identificarse como un nuevo evento o como parte de algún incidente existente. Un nuevo incidente también se puede identificar durante la función de gestión como resultado de información clasificada incorrectamente o información nueva que se descubre como resultado de un análisis técnico más profundo. La siguiente figura proporciona una ilustración del proceso del ciclo de un incidente.

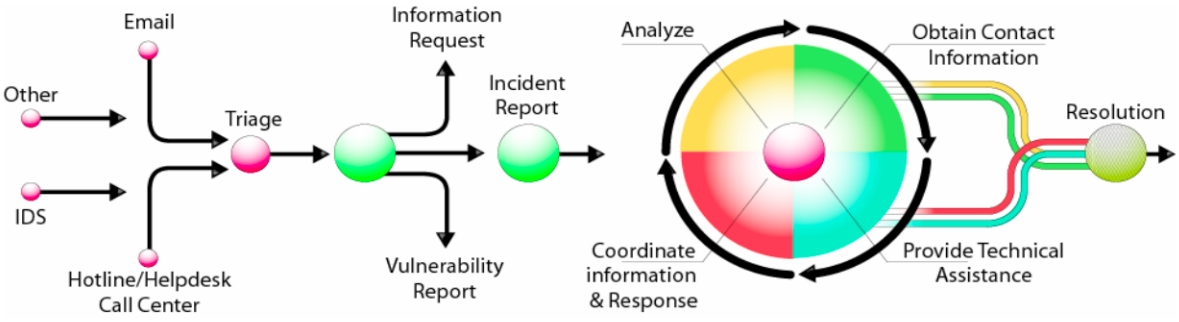

Figura 2.4: Ciclo de vida de incidentes CERT/CC [8]

Una vez que un incidente es creado, puede pasar a través de muchos estados diferentes, con toda su información relacionada, hasta que no se requiera más acciones

desde la perspectiva del equipo y finalmente se cierra. También es importante tener en cuenta que un incidente o evento puede recorrer la parte de análisis varias veces durante el ciclo de vida.

El cierre de un incidente normalmente ocurre cuando ninguna de las partes involucradas en el incidente está identificando o reportando nueva información y el CSIRT ha emprendido sus acciones para responder adecuadamente a todas las partes afectadas por la actividad.

Cuando un CSIRT decide cerrar un incidente, debe asegurarse de que todas las partes afectadas estén o hayan sido informadas del cierre.

Esto ayudará a establecer las expectativas adecuadas y evitará confusiones en los casos en que alguien piense que el incidente aún está abierto y se pregunte por qué no escuchan nada más del CSIRT. El equipo puede informar por separado a todas las partes involucradas cuando cierren el incidente o informar a las partes durante la correspondencia del incidente en curso.

Al igual que la definición de incidente, el criterio para abrir o cerrar un incidente puede variar de un equipo a otro. Esto significa que si un equipo cierra un incidente relacionado con otro CSIRT, este último puede no considerar que el incidente ha terminado y seguir analizando el problema.

Es posible que sea necesario reabrir los incidentes cerrados si se pone a disposición del equipo nueva información, como la reactivación de la actividad maliciosa en uno de los sitios involucrados. Cuando surge la necesidad de reabrir un incidente, el número de seguimiento original debe reutilizarse si es posible. Sin embargo, si la actividad no se considera una continuación del incidente original, es apropiado generar un nuevo incidente para la actividad y emitir un nuevo número de seguimiento.

De manera similar, puede estar disponible nueva información que vincule directamente dos o más incidentes que anteriormente parecían no estar relacionados. En tales casos, un equipo debe decidir si los incidentes deben fusionarse en uno solo. De ser así, hay que identificar el número de seguimiento que debe usarse y quién debe ser informado, o si deben permanecer como incidentes separados y marcados como relacionados.

2.2.2.2 Auditoría

Durante el ciclo de vida de cada incidente, es muy importante realizar un seguimiento de la información relacionada con el mismo en distintos niveles de detalle. Esto facilita la organización de la información y la respuesta eficaz. Esta información en una manera

lógica y organizada también proporcionará un registro histórico de la actividad y todas las acciones tomadas para ayudar en la distribución y asignación de trabajo del incidente.

Este registro también puede proporcionar información estadística y de tendencias que se puede utilizar dentro de la función de gestión o por otras funciones o servicios.

El nivel de detalle registrado puede variar de un equipo a otro según sus requisitos específicos, el nivel de servicio de gestión de incidentes proporcionado y la profundidad del análisis realizado.

2.2.3 Anuncio

La función de anuncio genera información adaptada a la comunidad en varios formatos para revelar detalles de amenazas en curso, pasos que se pueden tomar para protegerse contra esas amenazas o información sobre el alcance y la naturaleza de los ataques recientes informados al equipo.

Los principales objetivos de los anuncios son divulgar información a la comunidad, ayudarlos a proteger sus sistemas o buscar posibles signos de ataque al proporcionar notificaciones de amenazas potenciales, actuales o recientes; y además sugerir métodos para detectar, recuperarse o prevenir amenazas.

Al divulgar información relacionada con un tipo de ataque específico, se debe tener cuidado para garantizar que el nivel de divulgación sea suficiente para permitir que los destinatarios comprendan la amenaza y la verifiquen, pero no lo suficientemente detallada como para permitir que la información se utilice para implementar el ataque. Ésta es la tarea más desafiante de la función de anuncio.

El criterio de cuando debe ser anunciado algún tipo de amenaza depende de cada CSIRT. Así mismo, la importancia o prioridad que se le da a cada anuncio dependerá del contexto de la comunidad y la importancia de esa amenaza en ella.

La mayoría de los equipos generan una plantilla estándar para cada tipo de anuncio que indica el diseño y el contenido adecuados. Esta "repetición" puede facilitar enormemente el desarrollo del material y también ayuda a mantener una apariencia coherente en el formulario de anuncio y, por lo tanto, establece las expectativas de la comunidad sobre lo que cubrirá el contenido.

De acuerdo con las políticas y procedimientos del CSIRT que rigen la divulgación de información, la información destinada a ser utilizada en el anuncio debe estar autorizada para su divulgación al nivel apropiado, ya sea para distribución pública o restringida. Algunas reglas generales de autorización deben establecerse de antemano para ayudar

a que este proceso se desarrolle sin problemas en la práctica. Al proporcionar esta información a otras partes, debe hacerse evidente cualquier restricción de uso.

Dependiendo de la prioridad y el nivel de divulgación del anuncio, se debe considerar los canales de distribución apropiados para el mismo. Considerando si estos son lo suficientemente seguros, rápidos o adecuados en general.

2.2.4 Retroalimentación

Por otro lado, el tipo de solicitudes que recibe el CSIRT puede proporcionar una idea de las necesidades actuales de la comunidad y otros intereses del equipo. Como resultado, brindar retroalimentación a tales solicitudes puede ayudar a brindar un mejor servicio y al mismo tiempo aclarar las expectativas de la comunidad en lugar de ignorar problemas obvios y conceptos erróneos. El CSIRT debe esforzarse por responder a las solicitudes, sin importar cuál sea la solicitud, incluso si la respuesta es "No podemos responder a esta solicitud, está más allá de los servicios que brindamos", o apunta a otros recursos de información.

Además, luego de realizar un anuncio, es de esperar que los afectados respondan tales inquietudes y realicen solicitudes adicionales con respecto a lo comunicado. En este caso, la comunicación es fundamental para poder resolver incidentes a tiempo.

2.2.5 Interacciones

A lo largo del ciclo de vida del incidente, la mayoría de las actividades de un CSIRT implican interacciones con otras partes. Debido a la importancia y las implicaciones de tales interacciones, se debe tener mucho cuidado para establecer contactos con las partes adecuadas.

Establecer los contactos correctos o apropiados, es un arte en sí mismo. Es importante transmitir información, pero lo más importante es encontrar a la persona más adecuada para manejar y recibir la información y/o la persona autorizada para tomar las acciones y/o decisiones necesarias. Por lo tanto, establecer y mantener buenos contactos debe ser un esfuerzo continuo del CSIRT, con la intención de construir una red de confianza para satisfacer las necesidades del proceso de gestión de incidentes.

2.2.5.1 Contactos

Los contactos se pueden agrupar ampliamente en dos categorías: contactos relacionados con incidentes y contactos no relacionados con incidentes.

Contactos relacionados con incidentes: estos son los contactos que necesitará un CSIRT para manejar un incidente específico. Pueden incluir contactos dentro y fuera de una organización que experimenta un incidente.

En organizaciones grandes, puede haber un punto de contacto inicial predeterminado o "abuse" al que el CSIRT notifica sobre un informe de incidente que ocurre en ese sitio en particular. Sin embargo, puede ser esencial que el CSIRT se ponga en contacto con un departamento específico o con las personas adecuadas que puedan responder a la actividad. Sin contacto directo con el personal técnico o de gestión apropiado, el CSIRT puede perder tiempo y recursos valiosos.

Contactos no relacionados con incidentes: los contactos no relacionados con incidentes se pueden usar para proporcionar información para el equipo, ayudarlo a cumplir con su servicio, respaldar alguna operación o usarse para obtener información de expertos.

2.2.5.2 Búsqueda de contactos

Encontrar los contactos adecuados para las organizaciones no siempre es una tarea sencilla. Para los contactos que no son críticos, se pueden utilizar los recursos disponibles públicamente, como directorios telefónicos o servicios similares disponibles o mediante una búsqueda en Internet.

Siempre que una decisión crítica deba basarse en un contacto, el uso del contacto incorrecto puede resultar en la filtración o divulgación de información sensible a partes inapropiadas o a personas externas. También demuestra una falta de control y atención a los detalles dentro del CSIRT, lo que es malo para su reputación.

Para mantener la confianza de la comunidad, se debe tener mucho cuidado en identificar, examinar y utilizar los contactos correctos. Si la información de contacto disponible públicamente se puede falsificar, manipular o corromper, debe verificarse antes de su uso. Siempre es preferible obtener la información de contacto directamente de la fuente, de los propios contactos o de su gerencia.

Mantener la información de contacto puede ser un desafío más abrumador que encontrarla. La información de contacto se vuelve obsoleta, o parcialmente obsoleta, cuando las personas abandonan una organización, ascienden, se reasignan a otros tipos de trabajo o simplemente se trasladan a otro escritorio. Se puede pedir a los contactos que transmitan información relacionada con este tipo de cambios. Sin embargo, la realidad

es que esto rara vez ocurre ya sea con contactos externos o inclusive internos a la comunidad.

Para los contactos que no son críticos, es mejor aceptar alguna posibilidad de información desactualizada o incorrecta en la base de datos y corregir la información cuando se conozca. Para los contactos críticos, esto es menos apropiado, y las revisiones periódicas periódicas pueden ayudar a abordar este problema, además de pedirles a los contactos que proporcionen cambios y información actualizada.

2.3 Incident Management Systems

En esta sección se evalúa y compara varios de los sistemas de gestión de incidentes actuales según criterios planteados en la sección anterior 2.2:

Clasificación: evalúa si el sistema proporciona un punto único de contacto para aceptar, clasificar, priorizar y ordenar eventos debidamente. Si relaciona eventos nuevos con eventos existentes y utiliza números de seguimiento para los mismos.

Gestión: evalúa si el sistema proporciona funciones correctas para la gestión de incidentes en todo su ciclo de vida.

Anuncio: evalúa si el sistema se comunica correctamente con su comunidad objetivo.

Retroalimentación: evalúa si el sistema permite una retroalimentación en la comunicación con su comunidad cuando esta responde a anuncios.

Interacciones: evalúa si el sistema permite mantener actualizada la información de contacto al comunicarse con entidades internas o externas a la comunidad.

Automatización de tareas: evalúa si el sistema permite automatizar tareas diarias.

Los sistemas que veremos a continuación son:

- TheHive [4]
- RTIR [1]
- LUCIA [13]

- FIR [14]

2.3.1 TheHive Project

TheHive Project es una plataforma de gestión de incidentes gratuita, de código abierto diseñada para facilitar tareas a CSIRTs. Está estrechamente integrada con otras dos herramientas MISP [2] y CORTEX [15] que se explicarán en breve. Nace en el 2014 y es desarrollado por investigadores de CERT Banque de France [16].

TheHive Project se centra en tres pilares [4]:

- **Colaborar**: varios analistas pueden colaborar a un caso simultáneamente en tiempo real.
- **Elaborar**: se pueden crear casos y tareas relacionadas.
- **Actuar**: los observables de cada caso pueden ser analizados por Cortex enriqueciendo información y también pudiendo tomar acciones sobre los mismos con sus "Responders".

2.3.1.1 TheHive

Por sí solo, sin Cortex ni MISP, TheHive se centra en la gestión de casos, alertas, tareas y la auditoría de los mismos.

Las alertas son el objeto elemental de este sistema. Son creadas solo por feeds externos por medio de su API y su librería TheHive4py [18] y luego pueden ser convertidas o combinadas a casos, según corresponda; o ser descartadas como falsos positivos.

Los Casos son el objeto principal del sistema que agrupa todos los datos respectivos a una investigación: tareas a realizar, comentarios y artefactos. Este es el objeto que sigue el ciclo de vida del incidente, puede abrirse, cerrarse, reabrirse y combinarse. Pueden crearse "Cases Templates" o Plantillas que contienen observables y tareas precargadas pudiendo así crear distintos tipos de casos.

Un caso puede tener uno o varios artefactos, llamados Observables, como "IP, "dominio", "ASN"[8], etc. Los mismos son atributos que pueden ser enriquecidos luego con el sistema Cortex.

Las Tareas se utilizan para guiar el flujo de trabajo sobre un tipo de caso. Describen acciones que debe tomar el operador asignado al caso y en ellas puede agregar registros

[8]ASN: Autonomous System Number

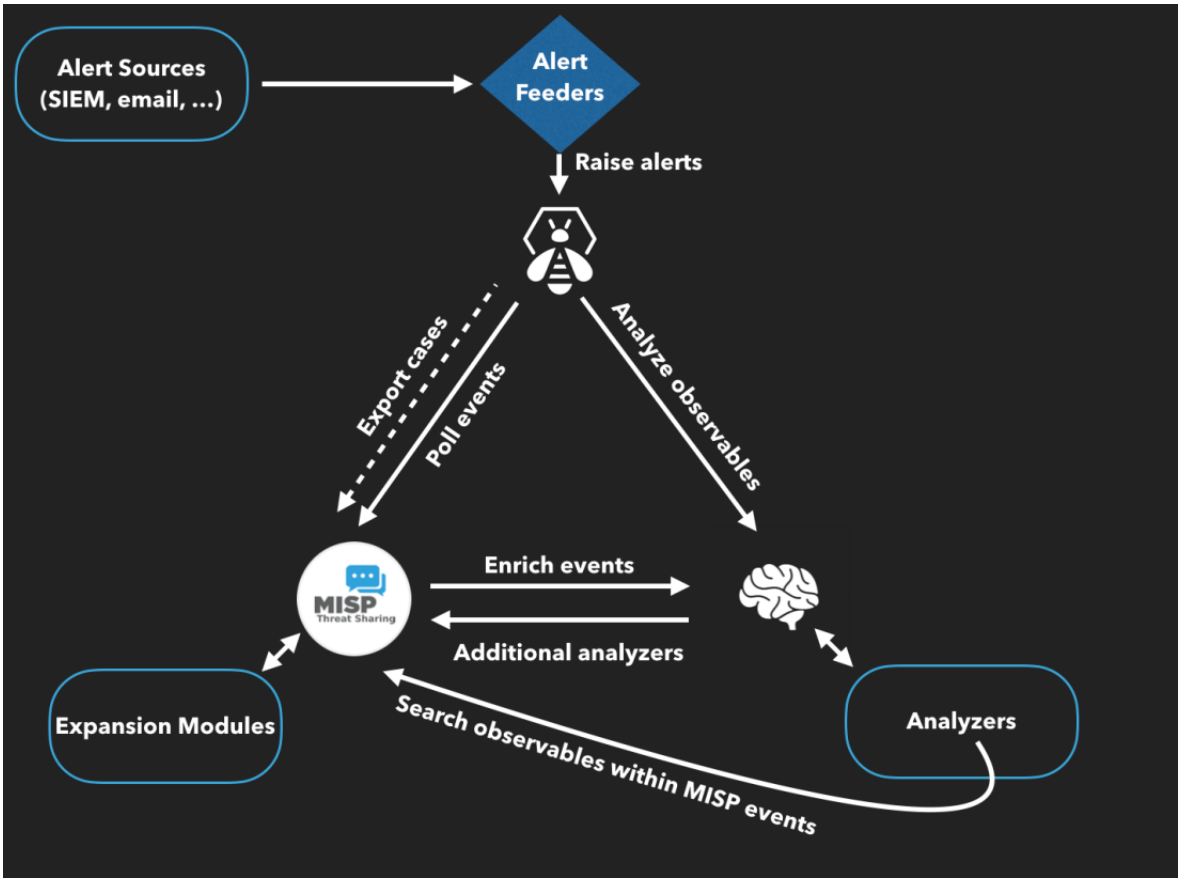

Figura 2.5: Esquema de relación entre TheHive, Cortex y MISP [17]

de texto, archivos u otros observables. Pueden ser utilizadas para realizar Playbooks, documentos que detallan los pasos a seguir en caso de que ocurra un incidente.

2.3.1.2 Cortex

Cortex [15] es un motor de análisis independiente, permite a los usuarios enviar observables e IOC [9] a herramientas populares de OSINT [10], a través de una serie de analizadores basados en Python [11].

[9] IOC: "Indicators of compromise" o "Indicadores de compromiso". Son toda aquella información relevante que describe cualquier incidente de ciberseguridad, actividad y/o artefacto malicioso, mediante el análisis de sus patrones de comportamiento

[10] "Open Source Intelligence" o "Inteligencia de fuentes abiertas" hace referencia al conocimiento recopilado a partir de fuentes de acceso público.El proceso incluye la búsqueda, selección y adquisición de la información, así como un posterior procesado y análisis de la misma con el fin de obtener conocimiento útil y aplicable en distintos ámbitos.

[11] Python: lenguaje de programación.

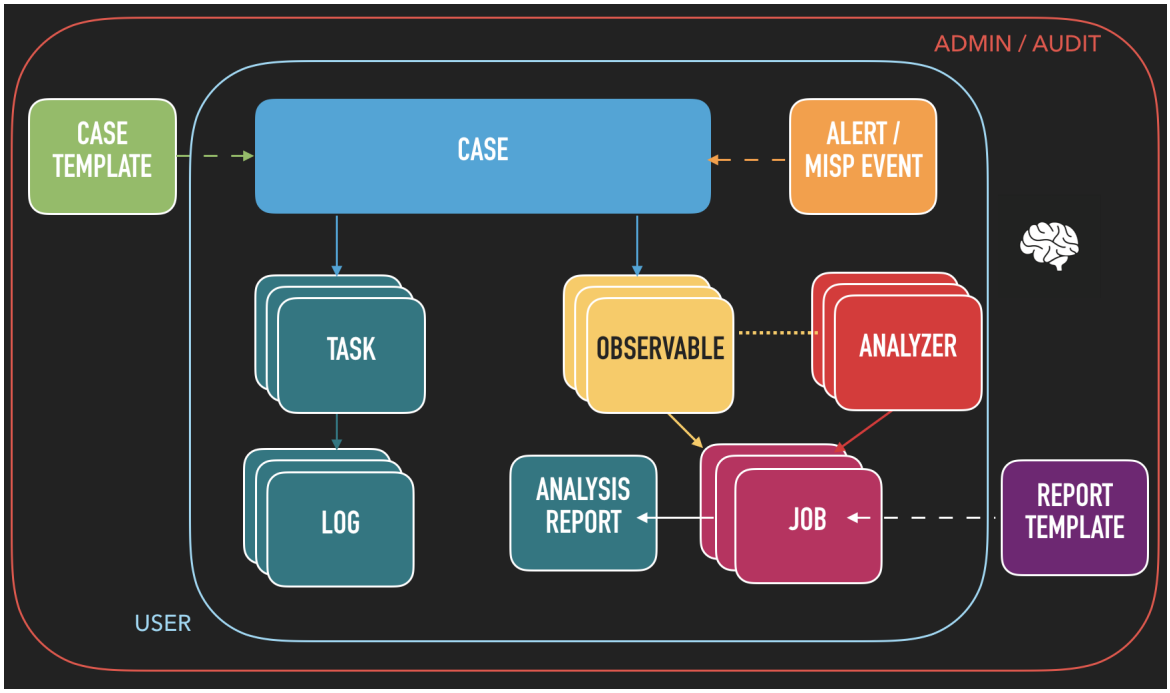

Figura 2.6: Arquitectura interna de TheHive [17]

Los analizadores y "Responders" son aplicaciones autónomas administradas y ejecutadas a través del motor central Cortex.

Analyzers: los "Analyzers" o "Analizadores" son programas basados en Python que enriquecen a los observables recolectando información de varios servicios externos de OSINT.

Responders: los "Responders" de Cortex realizan acciones específicas sobre alertas, casos, tareas y observables recopilados en el curso de la investigación. Por ejemplo, enviar un correo electrónico a los constituyentes de la comunidad, bloquear una dirección IP a nivel de proxy, notificar a los miembros del equipo que una alerta debe ser atendida con urgencia,etc.

2.3.1.3 MISP

TheHive también puede integrarse con MISP [2], una plataforma de inteligencia de amenazas o "threat intelligence ", para recopilar, compartir, almacenar y correlacionar IOCs de ataques dirigidos, inteligencia de amenazas, información de fraude financiero, información de vulnerabilidades o incluso información de lucha contra el terrorismo.

TheHive puede exportar e importar observables desde y hacia varias instancias de MISP, aprovechando así la capacidad de correlación de datos de MISP y retroalimentandose con nuevos observables y/o alertas.

2.3.1.4 Conclusión

TheHive es el sistema más similar a Ngen en el campo hoy en día. Abarca varios de los puntos de evaluación planteados anteriormente casi por completo.

Clasificación: las alertas brindan un único punto de entrada clasificando, ordenando y priorizando las a las mismas, permitiendo también rechazar falsos positivos.

Tanto alertas como casos poseen número de seguimiento y pueden ser combinados. Aunque, esto último, no se realiza automáticamente sin importar que los datos de los elementos sean idénticos.

Gestión: la gestión de incidentes es el punto más fuerte de TheHive, además es el único punto que puede abarcar solo por su cuenta sin necesidad de MISP y Cortex. Los casos permiten seguir y documentar todos los posibles estados de un incidente a lo largo de su ciclo de vida y las tareas ayudan al desarrollo de la investigación.

Anuncio: el anuncio debe ser creado manualmente. Este proyecto no contempla la necesidad de crear reportes a la comunidad. En caso de que exista algún tipo de anuncio, debe ser creado y enviado manualmente por un operador. Se podría configurar un "Responder" de Cortex para que envíe un email con la información de un caso, pero eso dependería de cada organización que utilice esta herramienta.

Retroalimentación: depende del tipo de anuncio, si existiera alguno. Como se explica en el punto anterior no parece haber una comunicación clara con la comunidad lo que genera una retroalimentación nula.

Interacciones: la integración con MISP ofrece gran capacidad para compartir información con entidades externas aunque no parece contemplar la necesidad de comunicarse con la comunidad de manera directa. Deja de lado la relación con la comunidad a la que pertenece. No existe manera de registrar los contactos de los constituyentes o guardar cualquier información relacionada a ellos. Como ya se explicó en 2.2.5.1 es muy importante contactar a los contactos adecuados ya que puede afectar

a la imagen del CSIRT. Puede retrasar la atención de un incidente en particular al no poder contactar a los responsables o tardar en encontrarlos cada vez que algún evento sucede.

Automatización de tareas : la automatización es ampliamente solucionada gracias a la combinación de TheHive con Cortex. Los "Responders" permiten crear distintos tipos de automatizaciones elementales y los analizadores ahorran tiempo a los operadores en búsqueda de información.

2.3.2 RT

"Request tracker" o "RT" [19] es un sistema de ticketing y flujo de trabajo de código abierto bajo la licencia GPL v3 [20], desarrollada y respaldada por Best Practical Solutions. Permite realizar un seguimiento de lo que se necesita hacer, quién está trabajando en qué tareas, qué se ha hecho y cuando se completaron o no las tareas.

2.3.2.1 Ticket

El Ticket es el principal objeto en RT, representa la tarea que se debe realizar. Los tickets contienen varios datos importantes como:

- **Prioridad**: un número del 1 al 99 que representa la importancia de la tarea.
- **Estado**: representa el estado del ticket dentro del ciclo de vida del mismo.
- **Dueño**: operador responsable de esta tarea.
- **Solicitante**: persona que está esperando que esta tarea se realice.
- **Cola**: cola a la que pertenece el ticket, puede ser considerada la categoría del mismo.
- **Referencia**: un ticket puede ser padre, hijo de otra tarea. O hacer referencia o depender de otra tarea.

Los tickets se organizan en "Queues" o colas que son la principal herramienta de organización de RT. Las colas son un conjunto de tickets esperando a ser realizados y a su vez son una especie de categoría para los mismos ya que pueden crearse varias colas para distintos tipos de tareas a realizar. Existen cuatro colas, que son creadas por defecto, para rastrear incidentes: informes de incidentes, incidentes, investigaciones y contramedidas.

2.3.2.2 Automatización a través de emails

Una de las principales características de RT es su automatización a través de emails. Puede ser configurado para crear tickets automáticamente cuando se recibe un email a una cuenta configurada específicamente. A su vez, puede enviarse un email para avisar que una tarea fue recibida vía email y correctamente creada. Y también, contestar emails enviados por RT son convertidos en comentarios en los tickets. Esta característica también permite a herramientas externas crear tickets mediante emails.

2.3.2.3 REST API

RT también provee una API REST para poder interactuar con la base de datos pudiendo realizar acciones sobre los tickets, usuarios y colas.

2.3.2.4 Scrips

Además pueden configurarse acciones automáticas mediante "Scrips". Estos realizan una tarea cuando se cumple alguna condición. Permiten personalizar el comportamiento de RT implementando nuevos flujos de trabajo y extendiendo la lógica.

2.3.3 RTIR

"RT for Incident Response" o "RTIR" [1] es una extensión de RT que proporciona colas y flujos de trabajo pre configurados diseñados para equipos de respuesta a incidentes. Está diseñada para proporcionar un flujo de trabajo simple y efectivo para los miembros de los equipos de un CSIRT.

2.3.3.1 Informes de incidentes

Los informes de incidentes es donde aparecen los nuevos eventos. Cuando un usuario envía un correo electrónico a la dirección que se configuró en RT, RTIR crea automáticamente un Informe de incidente o "IR" y establece su fecha de vencimiento de acuerdo con las reglas configuradas por el administrador.

2.3.3.2 Incidentes

Una vez que haya verificado que un nuevo informe de incidente es válido, puede crear un nuevo incidente a partir de él o vincular el IR a un incidente existente. Si recibe

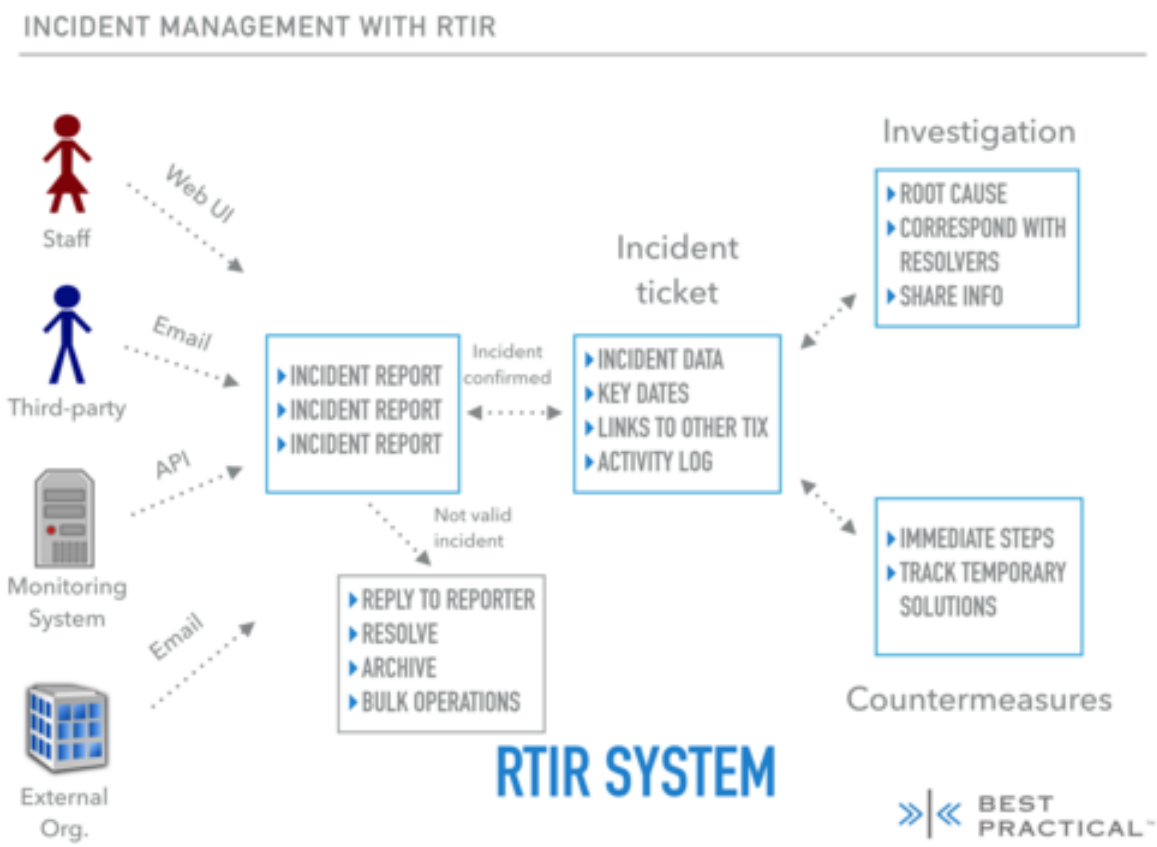

Figura 2.7: RTIR gestión de incidentes [1]

varios informes sobre el mismo problema, puede vincular todos estos informes al mismo incidente principal para mantenerlos juntos.

2.3.3.3 Investigación

Las investigaciones se utilizan para rastrear cualquier acción tomada externamente. Se puede lanzar una Investigación a una parte externa, a partir de un ticket de incidente, pidiéndoles que investiguen y/o solucionen un problema.

2.3.3.4 Contramedidas

Las contramedidas se utilizan para rastrear cualquier acción temprana tomada internamente, como por ejemplo comunicarse con el soporte para que bloquee una IP o bloque CIDR en el borde de la red. Puede crearlos a partir de un incidente existente. Las

contramedidas tienen varios estados: activación pendiente, activa, eliminación pendiente, eliminada.

2.3.3.5 Combinación de incidentes

Si resulta que dos Incidentes son realmente iguales, pueden fusionarse. La operación de combinación crea efectivamente un ticket a partir de dos, que contiene los datos y la correspondencia de ambos. Solo se deben fusionar los tickets dentro de las colas del mismo tipo.

2.3.3.6 División de incidentes

Permite crear un nuevo ticket a partir de uno existente con la información del ticket original.

2.3.3.7 Rechazar un IR

Rechazar significa "no vamos a trabajar en este informe". Hay varias razones para hacer esto: el ticket es spam, el problema no se resolvería, está fuera de alcance, etc. Los tickets rechazados se pueden volver a abrir.

2.3.3.8 Abandonar un IR

Esta operación es similar a rechazar un IR, pero cuando se abandona un incidente también se rechazan sus IRs, se resuelven las investigaciones y se remueven las contramedidas asociadas.

2.3.3.9 Bloqueo

Los bloqueos son informativos: si un usuario bloquea un ticket, a otros usuarios se les notificará (en rojo) que otro usuario ha bloqueado el ticket, con el nombre del usuario que lo ha bloqueado y cuánto tiempo lo ha tenido bloqueado, pero lo harán aún podrá editar y enviar cambios en el ticket.

2.3.3.10 Relacionar incidentes

Al crear un nuevo IR, puede ingresar el número de identificación de un incidente en el campo "Incidente". Esto creará un enlace desde el IR al incidente. Puede vincular

incidentes con informes de incidentes, investigaciones, contramedidas y artículos existentes.

2.3.3.11 Resolver un incidente

Esta acción sería equivalente a cerrar un incidente. Cuando se resuelve un incidente se resuelven también sus IRs hijos. Se le puede agregar un comentario de cierre.

2.3.3.12 Comunidad objetivo RTIR

RTIR permite configurar múltiples comunidades [21] permitiendo una división lógica de las colas de tickets y permisos de operaciones en los mismos. Se define por nombre, email, colas asociadas y permisos ACL.

Un ticket es asociado a la cola de una comunidad de diferentes maneras:

- Se asigna automáticamente según la cuenta de email correspondiente en el IR entrante.
- Cualquier incidente, investigación o contramedida creada a partir de un ticket hereda la comunidad del ticket original.
- Se puede seleccionar manualmente en la interfaz web.

2.3.3.13 Conclusión

Los sistemas de tickets fueron unas de las primeras herramientas que se utilizaron en la gestión de incidentes. RTIR mejora de gran manera el uso de tickets en este campo pero aun así este tipo de sistemas ya no son tan útiles. Los tickets son mucho mas útiles para organizar tareas entre equipos pero la gestión de incidentes conlleva otras mecánicas que exceden a la gestión de los mismos solo con tickets.

Clasificación: los puntos de entradas de RTIR son sus colas y cuentas de email que automáticamente agregan informes a las mismas. Los reportes pueden ser descartados en casos de falsos positivos y combinarse para crear incidentes haciendo que la clasificación sea muy ordenada.

Gestión: el ciclo de vida del incidente está cubierto en las distintas operaciones que se puede realizar con los tickets permiten abarcar los estados del incidente a lo largo de su

vida. No solo pueden combinarse y dividirse, sino que pueden ser referenciados por otros incidentes sin la necesidad de realizar una combinación.

Anuncio: tanto las investigaciones como las contramedidas pueden utilizarse para contactarse interna o externamente más allá de su función original, ya que RTIR no diferencia entre un correo interno o externo. No existen los tipos de anuncios, siempre se envía el mismo formato de email.

Retroalimentación: las investigaciones y las contramedidas permiten enviar emails y registrar las respuestas en el mismo ticket, lo cual permite un feedback óptimo.

Interacciones: el contacto solo se ubica a través de una dirección de email que se ingresa manualmente en el ticket, por lo que no hay un registro de los responsables de las distintas áreas de la comunidad. En otras palabras, pueden crearse investigaciones diferentes que contengan las mismas direcciones IPs pero estén reportadas a distintos responsables. Esto puede generar problemas como comunicar indebidamente un incidente.

Automatización de tareas: la automatización de varias tareas puede realizarse con los "Scrips" de RT. Hay varias maneras de aplicarlos ya que son muy genéricos y realizan tareas a través de alguna decisión. Un buen ejemplo del uso de estos objetos puede verse en el Proyecto LUCIA de CCN-CSIRT de España que se detalla a continuación.

2.3.4 Lucia

LUCIA (Listado Unificado de Coordinación de Incidentes y Amenazas) es una herramienta desarrollada por el CCN-CERT [12][22] para la gestión de incidentes en las entidades del ámbito de aplicación del Esquema Nacional de Seguridad de España. Está basada en el sistema, nombrado anteriormente, Request Tracker (RT) y en su extensión Request Tracker for Incident Response (RTIR).

Como se ha mencionado, el sistema LUCIA agrega funcionalidad al sistema RTIR a partir de varias modificaciones y desarrollos de módulos propios.

[12]CERT del Centro Criptológico Nacional de España

2.3.4.1 Modulo de sincronización

Una de las mejoras más importantes, que se pueden observar en este sistema, es la capacidad de interacción entre otras instancias de LUCIA. Este sistema puede generar una federación de sistemas en la que CCN-CERT establece una comunicación con cada uno de los sistemas independientes a través de un canal seguro, permitiendo el intercambio de incidentes entre las distintas instancias.

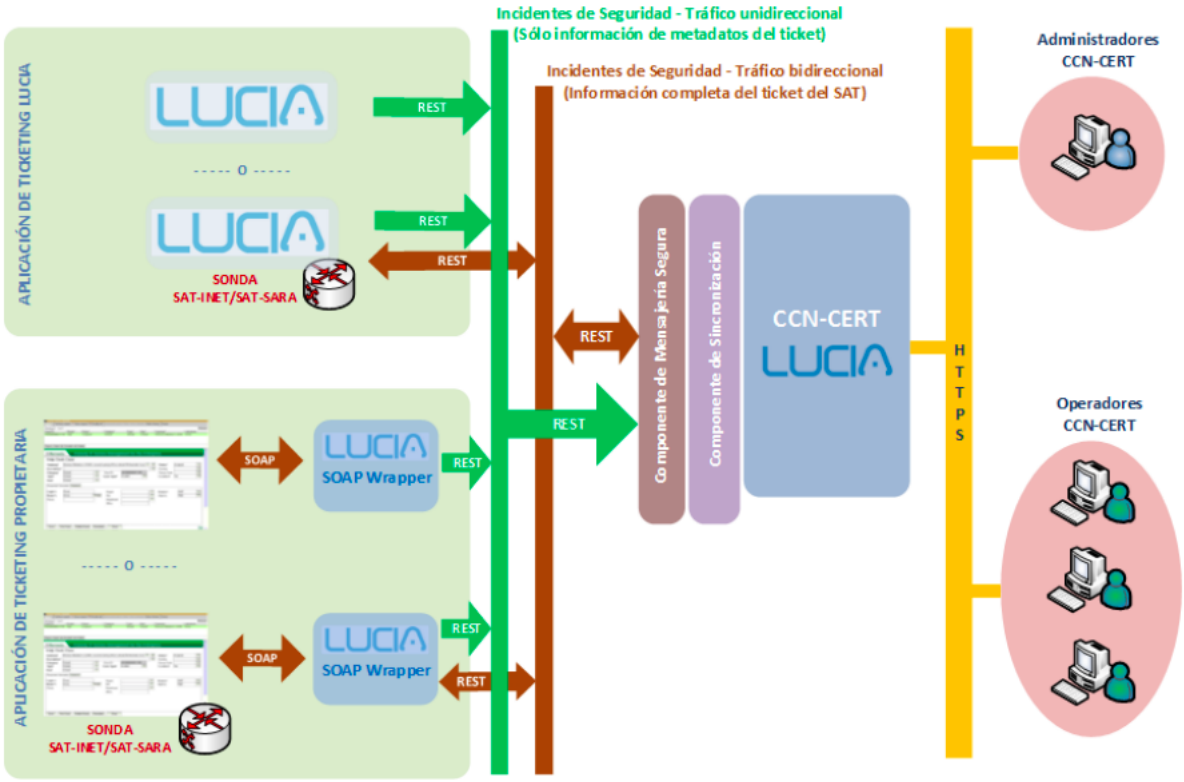

Figura 2.8: Arquitectura de LUCIA [23]

Por defecto las instancias de LUCIA comparten incidentes con el CCN-CERT solo unidireccionalmente. En el caso de que el organismo se encuentre adscrito a alguno de los proyectos de "Sistema de Alerta Temprana" del CCN-CERT. La sincronización de los incidentes puede ser bidireccional y completa o solo de metadatos.

2.3.4.2 Notificaciones de correo

Todas las acciones realizadas en LUCIA generan un correo de notificación a la cuenta del organismo con una breve descripción y un link al ticket relacionado.

2.3.4.3 Recordatorios

En LUCIA se encuentran programados una serie de recordatorios automáticos de incidentes vía email. El envío de recordatorios se realiza por las noches para cada incidente dependiendo de la criticidad del mismo y su actividad reciente.

2.3.4.4 Cierres automáticos

LUCIA resuelve incidentes automáticamente a medida que la "fecha límite" de los incidentes se va cumpliendo o el campo "resolución" sea "sin respuesta".

2.3.4.5 Conclusión

LUCIA es un sistema que se utiliza por gran parte de los organismos federados del gobierno de España. Ellos aseguran que han mejorado RTIR en varios aspectos como la API, campos adicionales y los módulos de automatización.

Cabe destacar que la función de sincronización instancias de LUCIA es, a mi criterio, algo innovador que agrega muchísimo valor al sistema.

Mas allá de las mejoras, LUCIA sigue siendo un sistema de ticketing y, por ende, sigue acarreando las desventajas de este tipo de sistemas.

En cuanto a los puntos de la conclusión, no hay muchas diferencias con RTIR pero existen claras mejoras que se realizaron a través de las mecánicas del mismo RTIR, como son los "scripts". Como por ejemplo, los módulos de automatizaciones que están realizadas a través de este mecanismo.

2.3.5 FIR

FIR (Fast Incident Response) [14] es una plataforma de gestión de incidentes desarrollada por el CSIRT Société Générale [24] de Francia, que es una de las principales empresas europeas de servicios financieros, actividad que también se extiende a otras partes del mundo. Es uno de los bancos más antiguos de Francia.

El sistema FIR si bien está basado en los requisitos del equipo de seguridad que lo desarrolla está pensado para que cualquier CSIRT pueda utilizarlo.

2.3.5.1 Eventos e incidentes

Los elementos básicos de FIR son los eventos e incidentes, son en realidad el mismo objeto diferenciados por un booleano, aunque el incidente tiene algunos campos adicionales.

Se pueden crear "Incident templates" que básicamente son plantillas de incidentes con estos campos precargados. Permitiendo acelerar el anuncio de incidentes creando reportes a partir de datos por defecto.

Luego de creados los eventos o incidentes se pueden realizar algunas acciones adicionales como agregar comentarios, editar la información de los mismos, abrir, cerrar o bloquear un incidente.

Además se pueden agregar atributos que están destinados a asociar valores numéricos con incidentes. Los atributos pueden utilizarse para generar campos adicionales para luego generar estadísticas a partir de los mismos.

2.3.5.2 Plugins

Se puede agregar funcionalidad a FIR a partir de módulos que pueden activarse y desactivarse a medida. Esto lo hace muy dinámico a la hora de abarcar distintos contextos. Mucha de la funcionalidad de FIR está en estos plugins [25].

fir_plugins: encargado de la funcionalidad de integración de todos los plugins internamente. Define una estructura y nomenclatura de módulos para una integración automática. Además permite definir requerimientos y dependencias para cada plugin.

fir_artifacts: permite definir artefactos que FIR puede buscar y correlacionar automáticamente. Los artefactos son objetos de interés que pueden ser enriquecidos luego con información extra. Además, FIR puede relacionar incidentes a partir de artefactos que tengan en común. Cada tipo de artefacto se generará automáticamente al examinar la descripción del incidente, los comentarios o nuggets.

Los artefactos por defecto son:

- Emails
- Hashes
- Hostnames
- IP addresses
- URLs

fir_nuggets: un nugget es un artefacto o cualquier cosa digna de mención que se pueda encontrar durante una investigación. Este plugin permite crear una línea de tiempo de investigación forense dentro de un incidente. Sirve para tomar notas cuando se lleva a cabo una investigación forense. Se pueden agregar nuggets en el orden que se desee y la línea de tiempo de investigación se creará automáticamente. Cada nugget tiene fecha de inicio y fecha de fin, un campo de nota y otro para poner información en texto plano.

fir_alerting: este plugin permite enviar emails de alerta o de contramedidas a los responsables del incidente. También pueden crearse plantillas de emails en la sección de administración. Existen dos templates posibles, de recipientes y de categoría:

- **Recipientes**: define el from, to, cc y bcc.
- **Categoría**: define el subject y el body del email a partir de la categoría del incidente.

fir_abuse: permite buscar el contacto de abuse de un artefacto en particular. La búsqueda se realiza automáticamente a través de una tarea de enriquecimiento.

Para enviar un email con esta dirección se debe hacer click derecho en el artefacto que despliega un menú "Send Abuse" y que luego abrirá una ventana para realizar el envío con la dirección de contacto de abuse en el campo "to".

fir_todos: permite crear lista de tareas asociada a cada incidente. Las mismas pueden marcarse como completadas o eliminarlas. Además, se pueden crear plantillas de tareas en la sección de administración que se agregaran automáticamente a incidentes según su categoría.

fir_api: agrega la funcionalidad de API REST a la aplicación.

fir_relations: permite asociar eventos e incidentes según las referencias de los mismos en las descripciones o comentarios de los mismos.

2.3.5.3 Conclusión

FIR es un sistema que parece estar maduro en muchos aspectos. El hecho de que haya sido creado en el contexto de un CSIRT, al igual que Ngen, hace que los requisitos de un equipo de seguridad se vean ampliamente contemplados.

Clasificación: la clasificación de los incidentes es buena. La categorización de los eventos e incidentes es personalizable y se pueden relacionar incidentes explícitamente o a través de sus artefactos.

Los eventos y los incidentes no tienen diferencia entre sí, salvo por unos pocos campos adicionales en el incidente, con lo que los incidentes no tienen una diferencia en funcionalidades o comportamiento que quizás si deberían tener en particular por su ciclo de vida.

Por otra parte, la identificación de los incidentes no es clara. No hay un ID específico al cual hacer referencia.

Gestión: el ciclo de vida de los incidentes es básico, solo se pueden abrir o cerrar. Aunque las funciones que agregan información adicional, como los comentarios, nuggets y artefactos, complementan de alguna manera esta falta de información en el ciclo de vida.

Anuncio: el anuncio se realiza a través del plugin fir_alerting el cual permite generar emails a través de plantillas y comunicarse con la comunidad objetivo. Pueden generarse plantillas para cada tipo de incidente y guardar en las mismas información de contacto. Este mecanismo es totalmente manual y depende del operador para que el reporte sea enviado en tiempo y forma.

Retroalimentación: el feedback se realiza cuando la comunidad responde a los emails que envía el sistema y estos pueden agregarse manualmente como comentarios en los incidentes.

Interacciones: aunque la información de contacto puede guardarse en las plantillas de alertas o utilizando el email de abuse desde el plugin fir_abuse, no hay manera de mantener información de contacto interna o externa.

Automatización de tareas: aunque muchas de las tareas cotidianas de un operador de un CSIRT están solucionadas con las funciones de FIR, muchas de las tareas son disparadas por la acción manual del operador. Igualmente, se puede utilizar la infraestructura, muy madura, de plugins para poder generar tareas automáticas.

2.3.6 Conclusión general

Los sistemas que han sido analizados en esta sección, dejan en evidencia la falta de funciones claves que un sistema de gestión debe tener para poder aplicarse a todos los contextos posibles. Cada uno está adaptado a un contexto distinto y soluciona algunos de los puntos evaluados pero todos fallan en la interacción con la comunidad a la que pertenecen.

Más allá de la retroalimentación, que muchas veces no se da o no se puede dar, la interacción con la comunidad es de vital importancia para la imagen de un CSIRT.

Ninguno de los sistemas nombrados permite tener una visión detallada de la comunidad como, por ejemplo, la topología de la red o los distintos servicios, redes o hosts que pueden componerla.

Es claro que lo que tienen en común estos sistemas es que dejan de lado alguno de los puntos planteados, creando la necesidad de un sistema lo suficientemente genérico que pueda abarcar todos estos puntos claves de gestión de incidentes en la mayoría de los contextos.

En la siguiente tabla se detallan los puntos de evaluación por cada uno de los sistemas de gestión analizados:

Nombre	Triaje	Gestión	Anuncio	Feedback	Interacciones	Automatización
TheHive	Completo	Completo	Incompleto	Incompleto	Incompleto	Completo
RTIR	Intermedio	Intermedio	Incompleto	Incompleto	Incompleto	Intermedio
LUCIA	Intermedio	Intermedio	Incompleto	Incompleto	Incompleto	Completo
FIR	Completo	Intermedio	Intermedio	Incompleto	Intermedio	Completo
Ngen	Completo	Completo	Completo	Intermedio	Completo	Completo

Table 2.1: Tabla de comparación de los sistemas de gestión de incidentes

CAPÍTULO 3

SOLUCIÓN PROPUESTA

El sistema Ngen comenzó a desarrollarse en el 2014, siendo una evolución de un sistema de gestión de incidentes más simple, creado anteriormente en el CERTUNLP.

Antes de comenzar con el desarrollo de Ngen, se realizó una basta investigación en búsqueda de sistemas o herramientas que pudieran ayudar a la actualización de la gestión de incidentes, pero las únicas soluciones que se encontraron fueron sistemas de ticketing que, como ya se ha explicado en capítulos anteriores, no cumplen con las necesidades operacionales de un CSIRT. Otros sistemas, como TheHive, no fueron considerados en su momento ya que recién comenzaban su desarrollo y fueron popularmente conocidos años más tarde.

Es entonces que Ngen nace de la necesidad de poder crear un sistema que pueda dar soporte a las tareas de gestión de incidentes que se realizan día a día en el CERTUNLP.

Adoptando como idea central la simplificación del trabajo diario de los operadores automatizando muchas de las tareas cotidianas. Como por ejemplo, la automatización del ciclo de vida del incidente. Esto permite a los casos abrirse y cerrarse automáticamente siguiendo ventanas de tiempo definidas en la configuración del sistema. Gracias a esta configuración, los casos cumplirán su ciclo de vida en tiempo y forma, según las necesidades del contexto.

Ngen hace especial hincapié en la comunicación con la comunidad, algo que ningún otro sistema estudiado dió importancia. El sistema permite tener una visión de la topología de la organización y relacionar varios tipos de contacto a cada uno de los activos.

Estos aportes diferencian a Ngen del resto de los sistemas actuales, gracias a su capacidad de automatización que permite a través de sus configuraciones y la versatilidad para poder realizar tareas cotidianas de manera simple.

Siguiendo el esquema planteado en la sección anterior del Servicio de gestión de incidentes (2.2) me referiré primero a los objetos elementales, caso, evento y artefacto. Siguiendo luego con la clasificación de estos objetos, su gestión, anuncio, feedback e interacciones.

3.1 Casos, Eventos y Artefactos

Los ataques no suelen ser solo un evento en particular, por lo general suelen ser la correlación de más de un evento a lo largo de un período de tiempo. La representación de incidentes como objetos individuales es, entonces, limitante. No se logra abstraer el real comportamiento de un ataque complejo a través de eventos aislados. Esta complejidad suele ser una combinación de eventos más elementales que pueden ser considerados en conjunto.

Los atacantes suelen primero observar a las víctimas para obtener información. Luego de esta investigación, suelen infiltrarse en los sistemas del objetivo intentando no ser detectados. Si tienen éxito, intentarán permanecer en los dispositivos de la víctima obteniendo más información y así realizar un ataque preciso. Esta metodología se denomina Kill Chain [26] y detalla los pasos que un atacante realiza para alcanzar su objetivo.

Estos distintos pasos que los atacantes realizan, pueden interpretarse como varios eventos individuales que luego serán investigados como parte de un ataque.

Para poder representar este tipo actividades en Ngen, se crearon los casos y eventos que son, en sí, la abstracción de los elementos detallados en "Eventos e incidentes" 2.1.1.

3.1.1 Evento

Los eventos son la representación de cualquier evento adverso, desde una vulnerabilidad en un servidor, hasta un ataque concreto en el mismo. Son los objetos elementales de Ngen y están compuestos por una taxonomía, una dirección IP o dominio y un "feed" o fuente de información. Pueden ser creados automáticamente a partir de distintas fuentes de información o manualmente por un operador. Cuando un evento es analizado y se comprueba que es verídico, se crea un caso que abre la investigación del mismo.

3.1.2 Caso

Los casos son un conjunto de eventos contextualizados por una investigación. Son los objetos que seguirán el ciclo de vida del incidente y pueden estar compuestos de uno o varios eventos. Mantienen la información del incidente a lo largo de toda la investigación hasta que se cierran.

Cada caso puede ser asignado a un operador para que realice la investigación pertinente. A lo largo de su ciclo de vida, los casos pueden ser atendidos y resueltos. La atención de un caso comienza cuando el asignado comienza la investigación de los eventos dentro del caso. Esta acción se suele denominar "abrir un caso". La resolución de un caso se considera cuando la investigación termina y el caso se considera "cerrado"

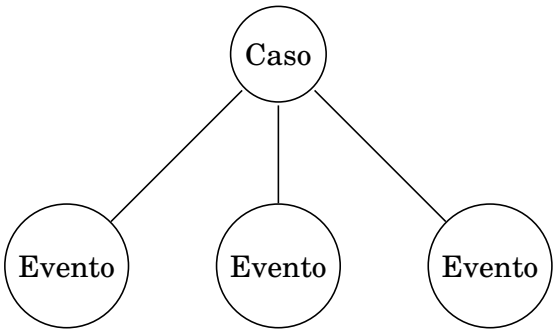

Figura 3.1: Arquitectura de un caso

3.1.3 Evidencia

Tanto los eventos como los casos permiten adjuntar archivos de evidencia que también pueden ser enviados como adjuntos en las comunicaciones por email.

3.1.3.1 Almacenamiento

Suele suceder que la evidencia adjuntada en distintos eventos o casos sea idéntica pero con nombre de archivo distinto. Esto lleva muchas veces a repetir evidencia, que se traduce a menos espacio en disco y más cómputo de análisis para su investigación.

Para evitar que existan archivos de evidencia repetidos se realiza un hashing sha1 sobre el archivo al momento de su creación. El hash resultante es utilizado como nombre en el sistema de archivos de Ngen y así evitar subir el mismo archivo varias veces.

Un ejemplo de la nomenclatura utilizada para la validación:

$$/media/evidence/(Case|Event)/<id>/<evidence\_sha1>.ext$$

Este control solo es realizable en el espacio de cada objeto. Lo que significa que es posible subir el mismo archivo para dos casos distintos.

3.1.4 Artefactos

El término artefacto se usa ampliamente en informática, aunque no existe una definición oficial de este término. La definición más cercana al significado es la de la palabra artefacto dentro de la arqueología que define un artefacto como algo hecho o dado forma por el hombre.

En informática se suelen llamar artefactos a los objetos creados a partir de tecnologías digitales.

Existen varios tipos de artefactos informáticos que podemos considerar:

- ip
- domain
- fqdn [1]
- url
- email
- hash
- file
- user-agent
- autonomous-system
- other

Como se puede observar en estos ejmplos, los datos que se consideran artefactos no son nada nuevo. La utilidad que trajo el uso de artefactos es en la separación del mismo con la entidad que lo contiene.

Cuando independizamos un artefacto, nos permite realizar algunas tareas interesantes, como por ejemplo:

[1] FQDN: fully qualified domain name

- Relación entre objetos: podemos relacionar varios objetos que tengan artefactos en común.
- Enriquecimiento: a varios de los datos considerados artefactos se les pueden agregar información a partir de la información que ya contienen, lo que comúnmente se llama enriquecimiento.

3.1.4.1 Relación de eventos a través de artefactos

Pueden existir eventos que no son idénticos pero están relacionados a partir de algún dato en particular. Estos eventos podrían estar relacionados parcialmente y haber sido agregados por separado.

Cuando un evento o un caso es creado, varios artefactos son creados con él. Si el artefacto ya existe, el objeto se agrega a la colección de relaciones de ese artefacto. Lo que significa que ese objeto ahora puede ver todos los objetos que comparten ese artefacto.

Por ejemplo, un evento que contiene la IP 163.10.0.135 podría estar relacionado con otro evento que contiene la misma IP pero fueron agregados por separado. Lo mismo para un evento que contiene el dominio unlp.edu.ar y podría estar relacionado con otro evento.

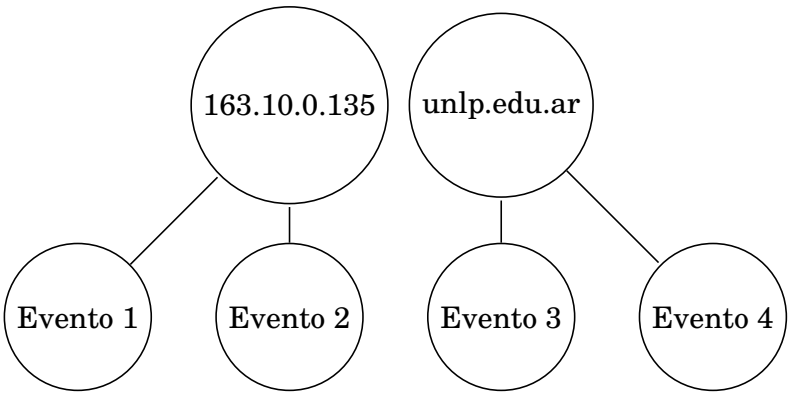

Figura 3.2: Relación de eventos y artefactos

Que compartan artefactos no quiere decir que necesariamente sean el mismo evento repetido o que pertenezcan al mismo curso de ataque. Pero a través de los artefactos se puede visualizar este tipo de relaciones mucho más fácilmente. Si un evento está repetido o pertenece a un caso ya en investigación, puede ser fácilmente detectado con esta mecánica.

El uso de artefactos puede ser de gran utilidad al momento de relacionar eventos nuevos, actuales o del pasado, facilitando el servicio de clasificación del CSIRT.

3.1.4.2 Enriquecimiento

El enriquecimiento es, básicamente, la generación de información a partir de la información que ya se tiene. Cuando hablamos de enriquecimiento de artefactos, nos referimos a poder generar más información a partir de los mismos y hasta poder crear otros artefactos.

Un enriquecimiento muy común, es el enriquecimiento de IPs. A partir de una dirección de red podemos obtener:

- Información de geolocalización.
- DNS que resuelvan esa IP.
- Información de contacto ya sea técnico o de abuse.
- Dominios relacionados con esa IP.

Podemos enriquecer nuevamente estos datos nuevos generando aún más información y artefactos.

Siguiendo con el ejemplo de arriba, si enriquecemos el dominio unlp.edu.ar podemos obtener la IP que resuelve el DNS, que es la IP 163.10.0.135. Ahora los eventos que antes parecían no tener relación, generan una nueva a partir de un enriquecimiento.

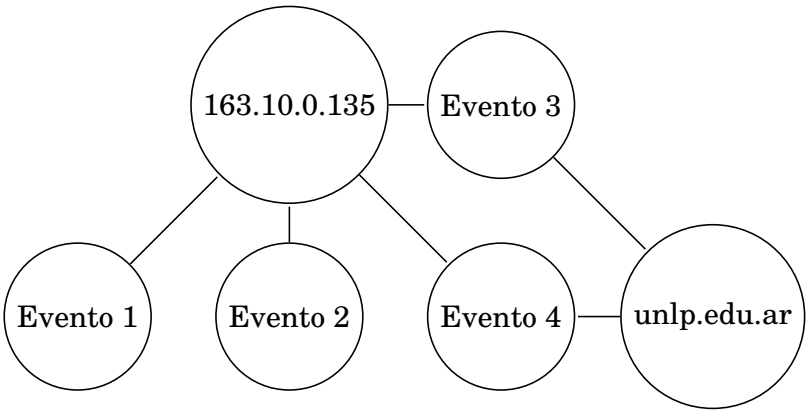

Figura 3.3: Enriquecimiento de artefactos y relaciones

3.2 Clasificación

Como se explicó en la sección 2.2.1, la función de clasificación proporciona un único punto de contacto, independientemente del método por el que lleguen los eventos, para su redistribución y manejo dentro del servicio de clasificación.

Esta función lleva a cabo la clasificación y priorización inicial de los eventos y determina si la información de los mismos está relacionada, o no, con algún evento histórico o actual.

También, es la función encargada de combinar estos objetos automáticamente según corresponda. Existen dos maneras de relacionar eventos:

- Combinación o "merging": se realiza cuando uno o más eventos idénticos, o se consideran como él mismo, se combinan en uno solo.

- Relación a través de artefactos: relación parcial entre eventos que comparten información en común, recién explicada.

Estas dos formas de relaciones cumplen con todos los puntos definidos en la relación de eventos de la función clasificación 2.2.1.1 definidas en el capítulo anterior.

3.2.1 Combinación de eventos y casos

Existen varios escenarios donde, por ejemplo, dos eventos son creados por fuentes de información distintas pero son, en sí, el mismo evento. Cuando esto sucede, se debe tomar acción para evitar eventos repetidos que pueden entorpecer la investigación de los mismos.

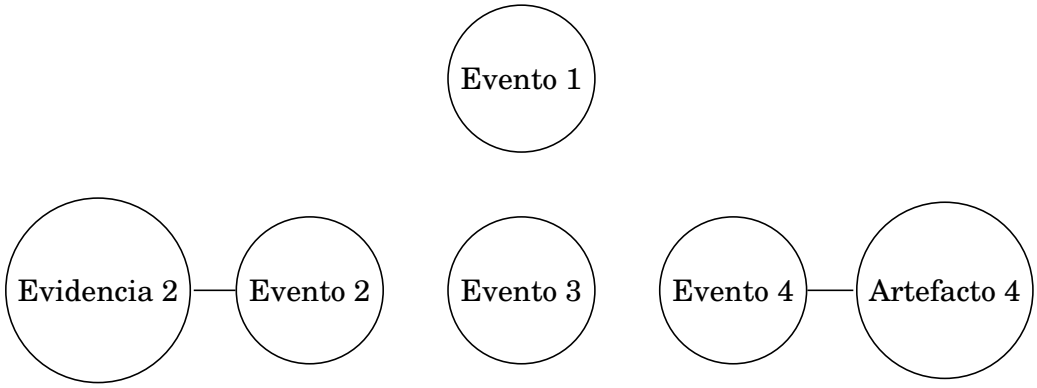

Figura 3.4: Combinación de eventos

Tanto los eventos como los casos pueden ser combinados entre sí. La combinación o "merging" se realiza a través de una estructura de árbol n-ario de profundidad 1. Cuando estos objetos se combinan, uno de ellos será elegido como padre del resto. Por ejemplo, si tomamos 4 eventos idénticos para su combinación, podemos tomar el primer evento como padre de los demás.

Los eventos hijos ahora quedarán bloqueados, lo que significa que no pueden ser alterados de ninguna manera, ya que esto rompería la relación que tienen entre ellos.

Cuando se realiza la acción de combinación, los objetos hijos transfieren varias de sus relaciones a su objeto padre, como:

- Evidencias
- Comentarios
- Artefactos

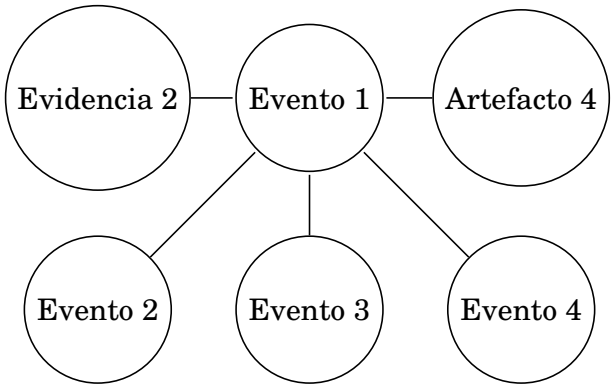

Figura 3.5: Combinación de eventos

Además, cuando los casos son combinados con otros que contienen eventos, estos también son transferidos al caso padre.

3.2.1.1 Reglas de combinación

Existen varias reglas que sirven para evitar bucles infinitos o incongruencias entre los objetos combinados:

- Un objeto no puede ser padre de sí mismo.
- Los objetos bloqueados no pueden combinarse.

- Los objetos hijos no pueden combinarse.
- Los casos estarán bloqueados cuando sean resueltos.
- Los eventos estarán bloqueados cuando el caso al que pertenecen está bloqueado.

3.2.1.2 Combinación automática y manual

La combinación de eventos puede ser tanto automática como manual. La combinación automática se realiza solo en eventos idénticos al momento de su creación.

Se puede combinar automáticamente un evento nuevo con otro ya existente, si sus taxonomías, fuentes de información y su IP o dominios son idénticos; y si el evento existente no pertenece a un caso ya resuelto.

Si el evento existente perteneciera a un caso ya resuelto, significa que este evento ya fue analizado anteriormente y su caso terminó. Por ende, se debería abrir un caso nuevo o reabrir el caso anterior. Reabrir automáticamente un caso podría parecer útil, pero realmente no se puede saber si un evento idéntico pertenece a un mismo caso que sucedió anteriormente. Los casos contextualizan a los eventos en investigaciones que pueden abarcar varios eventos distintos que pueden estar relacionados manualmente por los investigadores del CSIRT. Entonces, la reiteración de un evento que ya existió no necesariamente significa que su investigación anterior continua.

La combinación de casos es solo de manera manual. Gracias a su naturaleza heterogénea no es posible realizar esta tarea automáticamente, pero si existe un caso idéntico a otro sería fácilmente identificable a través de la relación de sus artefactos y eventos.

3.2.2 Uso de números de seguimiento

Punto también definido por la función de clasificación, los números de seguimiento permiten crear un esquema que el equipo puede utilizar para dar seguimiento a cada caso o evento en particular, facilitando la comunicación de los mismos con el resto del equipo o la comunidad.

Esos números de seguimiento deben ser considerados como información pública por lo que no deberían contener información confidencial.

Es muy común ver en sistemas utilizar los IDs de la base de datos como números de seguimiento.

Una solución simple para este problema es el uso de otro tipo de identificador. Uno que sea paralelo al de la base de datos y pueda ser publicado sin presentar un problema de seguridad.

Para realizar esta solución se utilizan los UUID [27]. UUID son las siglas de Universally Unique IDentifier, literalmente, 'identificador único universal'.

Es un número de 16 bytes (128 bits) se expresa mediante 32 dígitos hexadecimales divididos en cinco grupos separados por guiones de la forma 8-4-4-4-12. Por ejemplo: 550e8400-e29b-41d4-a716-446655440000

Existen 5 maneras distintas de generar los UUIDs. En está caso se utilizará la versión 4 que se define como UUID aleatorio.

La gran aleatoriedad de estos identificadores asegura que no existirán colisiones en su generación.

Estos identificadores se crearán para cada caso y evento en el momento de su creación. La función de estos UUIDs es ser la representación de estos objetos cuando son compartidos públicamente, ya sea con la comunidad o fuera de ella.

3.2.3 Prioridades

Es de suma importancia saber la gravedad relativa de un incidente para poder priorizar y resolver adecuadamente los problemas de manera rápida y eficiente.

Usualmente en la gestión de incidentes se utilizan prioridades para discernir cuál incidente será atendido primero.

La ITIL (Information Technology Infrastructure Library)[28][2] en su módulo de servicio de operaciones, intenta aclarar la identificación de la gravedad de los incidentes. Aquí se define a una **prioridad** como la suma de **urgencia** e **impacto** [29].

3.2.3.1 Impacto

Se define el impacto como una medida del efecto de un incidente en los procesos de negocio o imagen de la organización.

[2]La Biblioteca de infraestructura de tecnología de la información (ITIL) define la estructura organizacional y los requisitos de habilidades de una organización de tecnología de la información y un conjunto de procedimientos y prácticas de gestión operativa estándar para permitir que la organización administre una operación de TI y la infraestructura asociada. Los procedimientos y prácticas operativos son independientes del proveedor y se aplican a todos los aspectos dentro de la infraestructura de TI.

ITIL fue creado originalmente por la CCTA bajo los auspicios del gobierno británico, e ITIL es una marca registrada de la Oficina de Comercio Gubernamental del Gobierno del Reino Unido (normalmente conocida como OGC) [28].

El impacto no debería expresarse en términos absolutos, sino en un rango o grado que está sujeto a la interpretación del contexto. El mismo puede establecerse en función de distintos factores y es algo que cada organización define. Esos factores pueden ser:

- Número de personas afectadas.
- Pérdidas financieras potenciales.
- Gravedad de las responsabilidades legales.
- Número de sistemas afectados.

3.2.3.2 Urgencia

La urgencia de un incidente se mide por la rapidez con la que se debe resolver. Es decir, la urgencia está relacionada al tiempo. Depende de la velocidad a la que la organización espera algo. Este algo puede referirse a restablecer el funcionamiento normal del servicio o desarrollar, implementar y entregar un servicio o producto nuevo o actualizado. Cuanto más tiempo se esté dispuesto a esperar o retrasarse, menor será su urgencia.

ITIL define 3 niveles de urgencia y 3 de impacto: Alto, Medio y Bajo. O High, Medium y Low.

Podemos combinar estos 3 valores y crear una matriz de prioridades.

urgency/impact	High	Medium	Low
High	1	2	3
Medium	2	3	4
Low	3	4	5

Table 3.1: Tabla de impacto y urgencia [29]

Esta matriz nos da 5 valores llamados códigos de prioridad o "Priority code" y describir cada uno desde crítico a menos crítico.

3.2.3.3 Tiempos de respuesta y resolución

ITIL, además, nos acerca la definición de **tiempo de respuesta** y **tiempo de resolución**.

Tiempo de respuesta es el tiempo estimado para atender un incidente. El hecho de "atender" puede variar del contexto pero puede ser desde crear un ticket, iniciar una investigación o enviar un email a los responsables.

Priority Code	Description	Response Time	Resolution Time
1	Critical	Immediate	1 Hour
2	High	10 Minutes	4 Hours
3	Medium	1 Hour	8 Hours
4	Low	4 Hours	24 Hours
5	Very low	1 Day	1 Week

Table 3.2: Tabla de prioridades [29]

Tiempo de resolución es el tiempo estimado para resolver un incidente. Un incidente puede resolverse si el origen del mismo es mitigado. Pero esta definición también depende del contexto. Un CSIRT puede entender que un incidente está resuelto cuando no se repite en el tiempo o no hay evidencias de que persiste, más allá que los responsables confirmen o no la mitigación.

3.2.3.4 Prioridad por defecto

Esta función también asigna la prioridad por defecto correspondiente al caso o evento en el momento de su creación. Esta primera prioridad es asignada automáticamente si ningún valor fue asignado manualmente, tomando el valor asignado en la configuración del sistema.

Para los casos, la asignación automática de prioridad y otros campos se realiza a través de los "Templates", detallados más adelante.

3.2.4 Fuentes de información

Las fuentes de información o "feeds" pueden verse como la entrada de la función de clasificación 2.2.1 vista anteriormente.

Los CSIRTs reciben información de varios lugares, cada uno de estos con su propia estructura y datos. Estos deben ser analizados, normalizados y luego agregados a Ngen como nuevos eventos.

3.2.4.1 Normalización de fuentes de información

Uno de los problemas más comunes a la hora de tomar nuevas fuentes de información es el formato y la normalización de los datos. A lo largo de este documento se describe lo difícil que es tener una estandarización de muchos de los datos que un CSIRT trabaja.

En un principio el análisis de cada fuente debía hacerse adhoc. Para cada fuente se requería una librería específica que pudiese normalizar los datos a la estructura de Ngen.

Pueden encontrarse gran cantidad de datos similares expresados de distintas maneras. Aún cuando se trabaja con estándares internacionales, muchos de los creadores de estos datos pueden formatearlos de manera errónea o con subjetividad.

Además, cuando las fuentes de información crecen en cantidad y complejidad el mantenimiento de cada una crece.

Las fuentes de información pueden servirse de muchas maneras como, servidores de archivos o servidores web, o través de emails con adjuntos, etc. Lo que también dificulta su consumición, requiriendo aún más tiempo de análisis para cada fuente.

Esta problemática fue tomando cada vez más importancia y complejidad, lo que requirió que se desacople completamente de Ngen y se utilice un sistema que se encargará de todas estas tareas. El sistema encargado de hacer estas tareas es IntelMQ, que es explicado en detalle más adelante.

3.2.5 Taxonomías

El objetivo de las taxonomías es la representación de los posibles tipos de eventos que Ngen puede recibir, enviar y comunicar.

Como ya se describió en capítulos anteriores, la utilización de taxonomías es un tema delicado en el contexto de un CSIRT. Muchos CSIRTs definen sus propias maneras de referirse a vulnerabilidades y/o incidentes. Haciendo que compartir este tipo de información sea difícil.

El uso de estándares de taxonomías debería resolver esta temática pero tampoco hay una decisión de soportar algún estándar en particular y existen varios de ellos, aquí algunos de los más utilizados:

- ENISA [30]
- MITRE ATT&CK [31]
- MISP taxonomies [32]

Aunque estos estándares tengan cosas en común, como puede apreciarse en el mapeo de taxonomías de MISP [32], no suelen utilizarse con facilidad a la hora de compartir eventos. Por otra parte, si son útiles como complemento a las taxonomías internas y

propias de cada CSIRT. Pueden nombrarse a la par de una taxonomía en particular. Por ejemplo, si tenemos la taxonomía "denegación de servicio" podemos nombrar varias taxonomías externas y dar contexto a la primera:

- ecsirt:availability="ddos".
- europol-incident:availability="dos-ddos".
- circl:incident-classification="denial-of-service".
- enisa:nefarious-activity-abuse="denial-of-service".

Este tipo de solución es una de las más utilizadas ya que permite tener una propia interpretación de las taxonomías y poder relacionarlas a los estándares más conocidos.

3.2.5.1 Árbol de taxonomías

Los CSIRT también pueden crear sus propias taxonomías internas, que pueden no pertenecer a alguna de las nombradas en los estándares anteriormente, pero son similares en su estructura.

Algo que se puede observar en los distintos estándares, es el uso de 2 o más niveles de taxonomías. Un nivel más genérico y otro más específico.

Siguiendo con el ejemplo anterior de DoS, la taxonomía de ecsirt:availability="ddos" está compuesta por la categoría "availability" y la subcategoría "ddos".

Siguiendo la descripción de Enisa en la categoría "availability" también podemos encontrar:

- DoS
- DDoS
- Sabotage
- Outage (no malice)

Esto da a entender la necesidad de agrupar taxonomías en otras más genéricas que den una distribución más estructurada.

La propuesta de Ngen a esta necesidad es la estructuración de las taxonomías en forma de árbol donde las más genéricas se encuentran en los primeros niveles y las más específicas en las hojas.

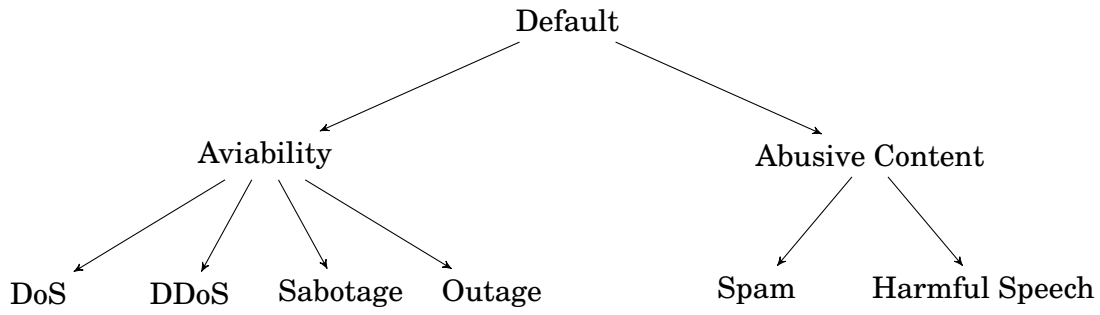

Figura 3.6: Árbol de taxonomías

De esta manera pueden definirse cualquier tipo de taxonomías y simplifica la comunicación de los casos a través de reportes, que serán explicados en el capítulo de "Anuncio" más adelante.

3.2.5.2 Playbooks

Cuando un operador trabaja con la gestión de incidentes suelen tener responsabilidades y realizar tareas que pueden estar pautadas según protocolos de seguridad.

Los playbooks son procedimientos escritos que sirven como guías paso a paso a los analistas de los CSIRT. En ellas se basan para poder gestionar un incidente o vulnerabilidad de manera ordenada, evitando así tomar decisiones apresuradas o ineficientes. Y fundamentalmente minimiza la inversión de tiempo al no reinventar las soluciones para los problemas recurrentes. Suelen estar asociadas a los tipos de incidentes y ser actualizadas constantemente.

En la práctica suelen ser una serie de tareas ordenadas que los operadores van marcando como completadas, pudiendo agregar comentarios y tomando nota del tiempo de realización de las mismas.

En este esquema una taxonomía puede tener varias playbooks, y una playbook estar compuesta de varias tareas a realizar.

Las tareas se irán completando por los operadores según avancen con la investigación de un caso. Estas tareas deben completarse para cada uno de sus eventos, en donde se registrarán la concretación de las mismas, comentarios y tiempos de resolución.

3.2.6 Templates

Los templates se utilizan para generar casos a partir de una agrupación de eventos automática.

Cuando una fuente de información ha demostrado su veracidad, por ejemplo, por la falta de falsos positivos y la velocidad con la que se renuevan sus eventos. Se puede confiar en que esa fuente de información generará, en su mayoría, eventos que terminarán siendo casos a investigar. Por esta razón la creación automática de casos suele tener un papel importante en la automatización de tareas dentro del CSIRT.

Los templates se conforman por campos de eventos y datos de casos

- Para eventos:
  - taxonomía
  - feed
  - CIDR
  - dominio
- Para casos:
  - TLP
  - state
  - lifecycle
  - priority

Los campos de eventos se utilizan para buscar los templates. Cuando un evento es creado, se consulta en la lista de templates si alguno coincide con los datos del evento. De ser así, se toma ese template y se crea un caso con los datos del mismo.

Pueden existir varios templates con taxonomías y feeds idénticos diferenciados por sus CIDR o dominios. Esto permite generar casos específicos para cada red sin importar la similitud de los eventos.

3.3 Gestión

Luego de pasar por la función de clasificación, los eventos son puestos a disposición de la función de Gestión.

Los objetos principales de esta función no son los eventos, sino los casos. Estos últimos son los que respetarán el ciclo de vida de un incidente y su investigación.

Para iniciar la investigación de un evento, este debe ser asignado a un caso. Esto puede realizarse de forma manual o automática a través de los ya nombrados Templates.

3.3.1 Ciclo de vida de un caso

Una vez creado el caso se inicia su ciclo de vida. Los casos pueden cumplir dos tipos de ciclos de vida, manual o automático. Esta distinción puede configurarse en cada caso.

En el ciclo de vida manual, los operadores deciden cuando el caso se abre o se cierra. Es decir, cuando cambia de estado.

Existe una problemática con respecto al ciclo de vida manual y es que cuando la cantidad de eventos y casos comienza a crecer, ya sea por la cantidad de fuentes de información conectadas o la real actividad de la comunidad, los operadores no logran abrir o cerrar los casos a tiempo, extendiendo los tiempos de respuesta. Lo que lleva a que el CSIRT atienda o resuelva casos siempre de forma atrasada.

Una vez que el CSIRT toma cierta agilidad y madurez puede saber si una fuente de información es confiable o no.

Que una fuente de información sea confiable o segura, significa que los eventos generados por esta fuente son de confianza y que generan pocos falsos positivos. Cuando esto sucede se puede suponer que los eventos siempre deben ser considerados como casos para reportarse en tiempo y forma.

3.3.2 Ciclo de vida automático

Para realizar este tipo de tareas se creó el ciclo de vida automático. Este ciclo utiliza ventanas de tiempo, definidas por las prioridades, para realizar cambios de estados automáticos sobre los casos.

Estas ventanas, llamadas "ventanas de atención", permiten saber cuando un caso es atendido o resuelto a tiempo y cuando está atrasado. El ciclo de vida automático toma estas ventanas y las utiliza para realizar los cambios de estados, abriendo y cerrando los casos.

El ciclo de vida automático está compuesto por dos acciones:

- **auto open**, donde el caso solo se abre automáticamente pero requiere cerrarse manualmente.

- **auto close**, donde el caso no se abre por sí mismo pero si se cierra automáticamente luego de ser abierto de manera manual.

Los operadores pueden optar por elegir el ciclo automático completo o alguna de sus partes.

3.3.3 Estados de un caso

Como se mencionó en 2.2.2.1 los incidentes pasan por varios estados a lo largo de su ciclo de vida, desde que son creados hasta que se cierran.

Los estados se utilizan para saber en qué parte del ciclo de vida se encuentra un caso en particular. Estos estados pueden variar dependiendo del CSIRT, su contexto y sus taxonomías.

Como ya se dijo en capítulos anteriores, las definiciones taxonómicas son muy distintas entre CSIRTs y a veces complican la comunicación entre los mismos. Igualmente, casi siempre existen estados que se consideran como un caso "abierto" o un caso "cerrado". Estos estados suelen siempre estar representados en las taxonomías de los CSIRTs de alguna manera.

Los estados pueden describir situaciones como, por ejemplo, la espera de confirmación o permiso para proseguir, hasta el hecho de tener que realizar una llamada a un superior. Por estas razones algunos estados pueden ser "waitng for call" o "waiting for approvement", según la necesidad específica de cada CSIRT y que puede no tener sentido en otro contexto.

Algo que se puede apreciar en el ejemplo anterior, es la necesidad de nombrar de manera diferente el mismo estado, algo que es muy común de ver. Estados como, "cerrado por inactividad", "cerrado", "descartado" comúnmente se refieren a diferentes maneras de llamar al estado final de cierre.

Esto puede ser representado como un "comportamiento" común entre los estados. Podemos entonces nombrar algunos comportamientos comunes:

- New: el estado que representa al caso recientemente creado pero que no está ni abierto ni cerrado. Es la primera instancia del ciclo de vida.
- Open: representa al estado de un caso cuando comienza a su análisis e investigación.
- Closed: cuando el caso es considerado cerrado.

Esta abstracción permite tener varios estados distintos que representan lo mismo, como por ej: "abierto", "esperando revisión", serían ambos dos estados de "Open".

Podemos realizar una segunda abstracción de estos "comportamientos" y representar estos estados como otros dos atributos, si un caso está "atendido" o "resuelto".

Como se muestra en la siguiente tabla, estos "comportamientos" pueden ser representados por el uso binario de estos dos booleanos "attended" y "solved". Estos

N	attended	solved	comportamientos
1	false	false	new
2	false	true	none
3	true	false	open
4	true	true	closed

Table 3.3: Relación de atributos y comportamientos

booleanos se agregan a los diferentes estados y se asignan con "true" o "false" dependiendo su uso.

Esta tabla también nos brinda información sobre otra necesidad en el uso de estados. La fila número 2 representa a un caso que no fue atendido pero fue cerrado. Esto no debería suceder en un caso de uso normal de un CSIRT ya que para cerrarse , los casos, deben ser primero abiertos o al menos atendidos de alguna forma.

Esto nos lleva a pensar que los estados deberían tener un orden y no pueden ser aplicados sin criterio. Sin ciertas restricciones, un operador podría aplicar el estado "esperando por aprobación" a un estado cerrado u otras combinaciones incongruentes de estados.

3.3.3.1 Grafo de estados

La solución a esta necesidad es un grafo direccionado de estados. Esto permite definir los pasos requeridos para llegar a un estado determinado.

Como se ejemplifico anteriormente, un caso puede tener el estado cerrado si antes haber pasado por algún estado de "open". Entonces, se debería definir un grafo de estados donde el estado "abierto" tenga una arista al estado "cerrado".

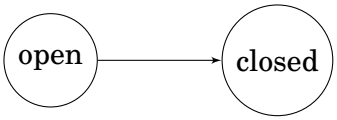

Figura 3.7: Grafo de estados de abierto a cerrado

Siguiendo con el ejemplo, el estado "open" debería también tener un estado padre, ya que un caso tampoco debería ser abierto sin pasar por un estado de "new" o creación.

Entonces a partir de este ejemplo ya podemos presentar interacciones de estados más complejas, como las ejemplificadas antes.

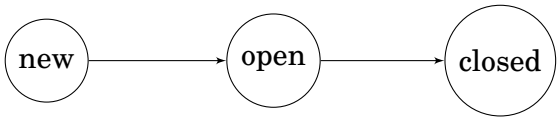

Figura 3.8: Grafo de estados de nuevo a cerrado

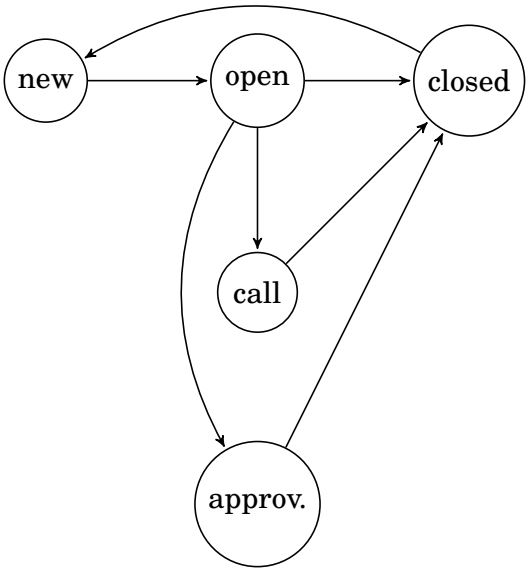

Figura 3.9: Grafo de estados complejo

En este último grafo podemos ver dos estados "open" adicionales "call" y "approvement" que pueden asignarse a casos abiertos. También una nueva arista de "closed" a "new" para poder reabrir casos cerrados ubicándolos en el inicio del grafo.

3.3.4 Ventanas de atención

Cuando los casos pasan por varios estados a lo largo de su ciclo de vida, pueden ser modificados, haciendo que sean atendidos o resueltos.

Estos cambios de estado se ven reflejados en varios campos fecha del caso:

- **created**: contiene el timestamp del momento de creación del caso.
- **attend_date**: contiene el timestamp del momento en el que el caso fue atendido.
- **solve_date**: contiene el timestamp del momento en el que el caso fue resuelto.

Combinando estos estados con los tiempos de respuesta de las prioridades 3.2.3. Se puede observar que estos tiempos no solo sirven para marcar los límites de tiempo

necesarios, sino que también pueden servir para definir ventanas de tiempo, en el ciclo de vida del incidente, que permitan saber cuándo un caso es atendido o resuelto a tiempo.

Entonces podemos definir a las ventanas de atención como:

- attend window: case:created + priority:attend_time.
- solve window: case:attend_date + priority:solve_time.

Podemos saber si un caso es atendido a tiempo si luego de ser creado cambia a un estado atendido dentro de la "attend window". Y podemos saber si un caso fue resuelto a tiempo si luego de ser atendido cambia a un estado resuelto dentro de la "solve window". En caso contrario, si cambia de estado fuera de las ventanas, podemos decir que esa acción está retrasada.

Siguiendo la tabla de prioridades 3.2, si un caso tiene prioridad "Medium" tendrá un tiempo de respuesta de 1 hora y un tiempo de resolución de 8 horas.

Entonces podemos decir que, según los tiempos de la prioridad, un caso será atendido a tiempo si lo hace antes de 1 hora después haber sido creado y estará resuelto a tiempo si lo hace antes de 8 horas de ser atendido.

En la siguiente línea de tiempo muestra la relación de las ventanas de atención y los cambios de estado de un caso a través del tiempo. Se ilustra un caso que cambia de estados en sobre los límites de las ventanas que no estaría retrasado ni adelantado en su atención.

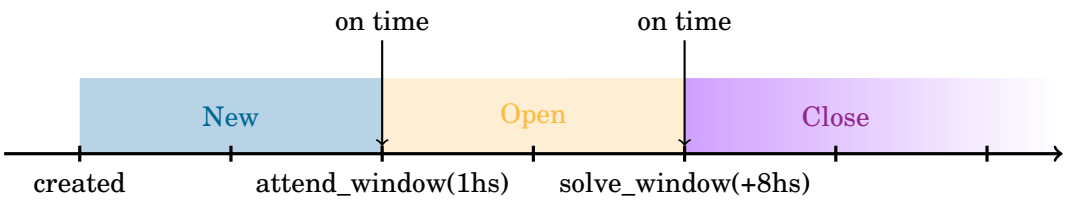

Figura 3.10: Línea de tiempo de cambio de estados a tiempo

En esta segunda figura podemos ver un caso que cambia de estado antes de los límites de las ventanas dándonos tiempos de atención adelantados

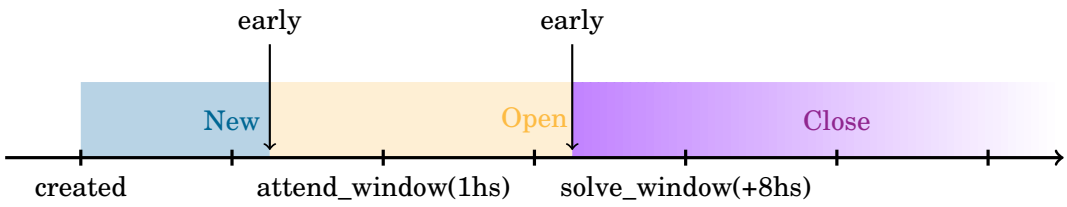

Figura 3.11: Línea de tiempo de cambio de estados adelantados

En esta tercera figura podemos ver un caso que cambia de estado después de los límites de las ventanas dándonos tiempos de atención retrasados

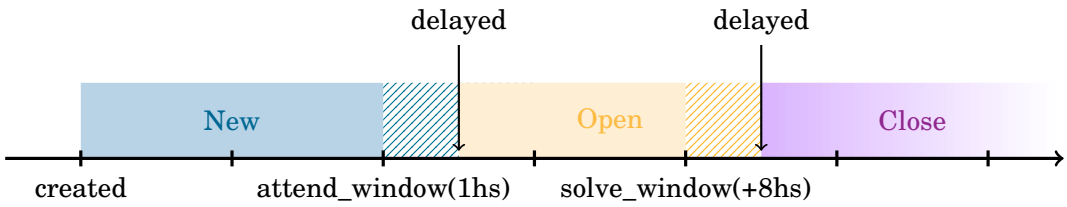

Figura 3.12: Línea de tiempo de cambio de estados atrasados

Cuando un caso se retrasa en sus ventanas se dice que está "sin atender" o "sin resolver".

Entre la fecha límite de la ventana de atención y la fecha en la que el caso fue atendido o resuelto se forma otra ventana, representada con área de líneas diagonales, que representa el tiempo en el caso se encuentra retrasado.

Una particularidad de esta ventana de retraso es que casi siempre existe no importa el contexto, ya que es casi imposible atender o resolver manualmente todos los casos de una comunidad en tiempo y forma. En la mayoría de los casos siempre existirá algún retraso.

Este retraso se identifica de distinta manera según la ventana de atención. Para los no atendidos es "unattend" y para los no resueltos "unsolved".

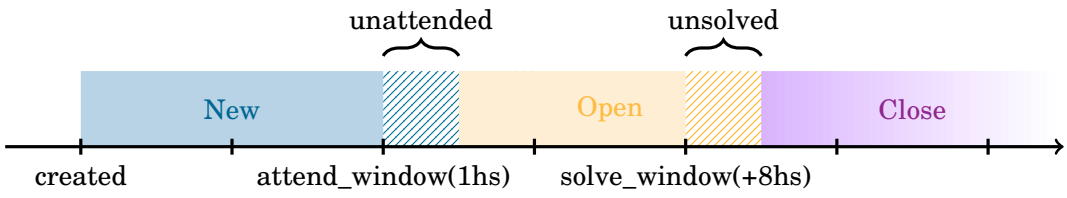

Figura 3.13: Línea de tiempo de cambio de estados con ventanas de retraso

Al igual que las demás ventanas, las ventanas de retraso están representada por dos campos del tipo fecha. Se utilizan los ya nombrados "attend_date" y "solve_date" que son guardados cuando el caso es atendido o resuelto respectivamente.

- unattend window: (case:created + attend_date) - attend window.
- unsolve window: (case:attend_date + solve_date) - solve window.

3.4 Anuncio

Una parte fundamental de la gestión de incidentes es la manera en que el CSIRT se comunica con su comunidad.

Cada vez que un caso es abierto para su investigación, o se genera alguna acción importante sobre el mismo, se debe informar a los involucrados. Estos deben recibir la información del caso de una manera simple y concisa, que no sea difícil entender.

Cuando una investigación es comunicada, debe contener algún tipo de ayuda para que los involucrados puedan solucionar las vulnerabilidades y comprender el impacto de las mismas.

3.4.1 Reportes

La solución de Ngen a este problema es la utilización de reportes generados en distintas partes del ciclo de vida de los casos.

Los reportes pueden configurarse para generar contenido HTML que será enviado a los involucrados luego de ciertas acciones:

- Crear
- Asignar un evento
- Abrir
- Cambiar de estado
- Cerrar

Estos reportes serán enviados a los involucrados de los eventos dentro del caso y opcionalmente al equipo del CSIRT y al operador asignado del caso. Estos últimos pueden

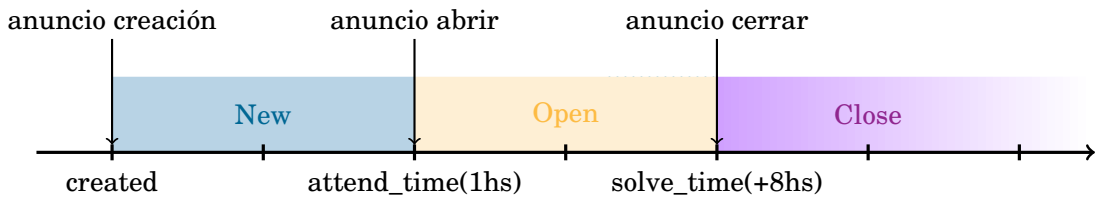

Figura 3.14: Anuncios en el ciclo de vida del caso

ser configurados para recibir anuncios solo de cierta prioridad. Por ejemplo, el equipo puede querer solo recibir emails de criticidad Alta o mayor.

El asunto de cada email en las comunicaciones tendrá el UUID del caso en cuestión, para poder realizar un seguimiento de las solicitudes de los involucrados que pueden surgir luego del anuncio.

3.4.1.1 Lenguaje de los reportes

Cuando se envían reportes a los involucrados en un caso, es importante saber en que lenguaje se debe enviar. El lenguaje de la comunidad por lo general será siempre el mismo, pero al comunicarse con comunidades externas los lenguajes pueden variar.

Desde la configuración general de Ngen se puede configurar un lenguaje de comunicación externo. Esta configuración permite enviar reportes fuera de la comunidad en un lenguaje específico. Este lenguaje suele ser inglés ya que es un idioma que se considera universal.

3.4.1.2 Reportes y evidencias

Las evidencias de los casos y los eventos son adjuntadas en las distintas comunicaciones para que los involucrados puedan evaluarlas.

Cuando se adjuntan varios archivos a un email suele ser difícil distinguir qué archivo corresponde a un evento o caso solo teniendo como factor el nombre del archivo. Más allá que en el detalle del email se listen los adjuntos de cada evento, si los involucrados descargan varios adjuntos para su análisis, no podrán saber, a simple vista, a cuál evento pertenece.

Para facilitar la distinción de los adjuntos y sus respectivos eventos o casos, se genera un nombre de archivo con un formato detallado.

$$Class(UUID) : created\_date : evidence.filename$$

Un ejemplo concreto:

$$Event(550e8400-e29b-41d4-a716-446655440000):2022-06-22:screenshot.png$$

De esta manera los involucrados pueden relacionar cada adjunto con su respectivo caso o evento.

3.4.1.3 Reportes y taxonomías

Los reportes del ciclo de vida de los casos comunican eventos que son de una taxonomía en particular.

Cuando se genera el mensaje con toda la información necesaria para la comunicación, las taxonomías brindan las recomendaciones sobre cómo mitigar el problema que representan.

Los reportes de las taxonomías se componen de 6 campos:

- Problema: este campo detalla las características principales de la vulnerabilidad o incidente.
- Problemas derivados: se detallan los problemas que puedan derivar del problema.
- Verificación: instrucciones para verificar que el problema reportado existe.
- Recomendaciones: recomendaciones sobre la mitigación del problema.
- Más información: información adicional.
- Lenguaje: este campo permite guardar el mismo reporte en distinto lenguajes para una comunicación óptima dentro y fuera de la comunidad.

Si un evento tiene una taxonomía que no tiene un reporte asignado, este no podría generar el mensaje correspondiente para realizar la comunicación. Utilizando la estructura de árbol de las taxonomías, se puede igualmente comunicar ese evento buscando hacia arriba en el árbol las taxonomías padre hasta la taxonomía por defecto. La cual siempre sabe cómo reportarse.

3.4.1.4 Renotificaciones

Cuando un caso es atendido se envía un reporte automáticamente a los involucrados, en el mejor de los casos este reporte es contestado dentro de la ventana de atención y el caso es cerrado a tiempo.

Muchas veces, los involucrados no contestan los reportes por distintas razones. Una de las maneras que se tiene para poder completar esta comunicación es reiterar el reporte con un aviso de renotificación. Con esto se intenta dar a entender la urgencia y así obtener una respuesta.

Para ésto se crearon 2 campos nuevos:

- En los casos se creó **notification_count** que contiene la cantidad de veces que el caso ya fue notificado.

- En las prioridades se creó **notification_amount** que contiene la cantidad de veces que se quiere que los casos sean notificados.

A partir de estos campos podemos configurar las prioridades para permitir notificar automáticamente los casos la cantidad de veces necesaria.

Entonces podemos calcular las distintas renotificaciones dividiendo la ventana de resolución por la cantidad de renotificaciones configurada y luego sumarlas a la fecha de atención.

$$attend\_date \quad + \quad solve\_window \quad * \quad \frac{notification\_count}{notification\_amount + 1}$$

Se le suma 1 a notification_amount para que la última notificación no caiga sobre el límite de la ventana de atención. Cuando la división de notification_count y notification_amount da 1 se ignora.

Siguiendo con el ejemplo de prioridad Medium con un notification_amount de 3 + 1 podemos calcular las divisiones:

$$8\,hs * (1\,/\,4) = 2hs$$
$$8\,hs * (2\,/\,4) = 4hs$$
$$8\,hs * (3\,/\,4) = 6hs$$
$$8\,hs * (4\,/\,4) = 8hs$$

Y luego sumar estas renotificaciones a la fecha de atención:

De esta manera podemos configurar las ventanas y la cantidad de renotificaciones de manera que se acople a los tiempos del contexto de cada CSIRT.

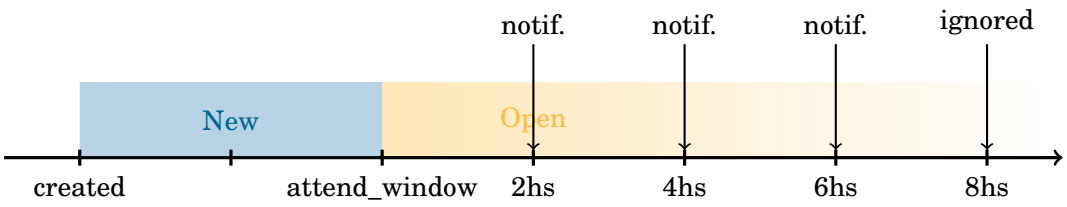

Figura 3.15: Línea de tiempo de renotificaciones

3.4.2 Nivel de divulgación

Los casos pueden contener información sensible que debe ser compartida solo con una comunidad específica o una audiencia limitada. Tanto a los casos como a los eventos, se puede asignar un "Traffic Light Protocol" o TLP [33], que facilita el intercambio de información sensible pero no clasificada. A través de este esquema se puede asegurar que esta información sensible esta siendo compartida con la audiencia apropiada.

El TLP se agrega a cualquier tipo de comunicación de un caso, para especificar las restricciones de divulgación a los receptores del mensaje.

3.4.2.1 Traffic Light Protocol (TLP)

Se utilizan cuatro colores para indicar los límites esperados más allá del receptor: rojo, ámbar, verde y blanco

A continuación se detalla cada uno de ellos [34]:

- TLP:RED
  - **Cuándo utilizar:** Se debe utilizar TLP:RED cuando la información está limitada a personas concretas, y podría tener impacto en la privacidad, reputación u operaciones si es mal utilizada.
  - **Cómo compartir:** Los receptores no deben compartir información designada como TLP:RED con ningún tercero fuera del ámbito donde fue expuesta originalmente.
  - **Color:** #ff0033
- TLP:AMBER
  - **Cuándo utilizar:** Se debe utilizar TLP:AMBER cuando la información requiere ser distribuida de forma limitada, pero supone un riesgo para la privacidad, reputación u operaciones si es compartida fuera de la organización.

  - **Cómo compartir:** Los receptores pueden compartir información indicada como TLP:AMBER únicamente con miembros de su propia organización que necesitan conocerla, y con clientes, proveedores o asociados que deban estar al tanto para protegerse a sí mismos o evitar daños. El emisor puede especificar restricciones adicionales para compartir esta información. Nota: se debe especificar TLP:AMBER+STRICT , si la fuente desea restringir la compartición sólo a la propia organización.
  - **Color:** #ffc000

- TLP:GREEN

  - **Cuándo utilizar:** Se debe utilizar TLP:GREEN cuando la información es útil para todas las organizaciones que participan, así como con terceros de la comunidad o el sector.
  - **Cómo compartir:** Los receptores pueden compartir la información indicada como TLP:GREEN con organizaciones afiliadas o miembros del mismo sector, pero nunca a través de canales públicos.
  - **Color:** #33ff00

- TLP:WHITE

  - **Cuándo utilizar:** Se debe utilizar TLP:WHITE cuando la información no supone ningún riesgo de mal uso, dentro de las reglas y procedimientos establecidos para su difusión pública.
  - **Cómo compartir:** La información TLP:WHITE puede ser distribuida sin restricciones, sujeta a controles de Copyright.
  - **Color:** #ffffff

Código	Resumen
TLP:RED	De no divulgación, restringido solo a participantes.
TLP:AMBER	Divulgación limitada, restringida a las organizaciones de los participantes.
TLP:GREEN	Divulgación limitada, restringida a la comunidad.
TLP:WHITE	La divulgación no está limitada.

Table 3.4: Tabla resumen de TPL

3.4.3 Retroalimentación

Una vez que la comunicación fue realizada, es de esperar algún tipo de respuesta por parte de los responsables.

Si el anuncio se realizó a través de un email, el responsable puede responder ese email manteniendo el asunto del mismo. Los operadores podrán entonces relacionar esa respuesta con el caso.

Estas interacciones con los responsables suelen quedar fuera de la documentación de los casos. Esto, a futuro, puede perjudicar la investigación debido a que la información quedó solo en el intercambio de emails y no dentro del sistema.

3.4.3.1 Comentarios

Tanto en los eventos como en los casos se pueden realizar comentarios que pueden ayudar a documentar información adicional de los mismos. Como el intercambio de emails con los responsables, información adicional de la investigación o intercambio de opiniones entre los operadores.

También puede utilizarse un pequeño script en el servidor de email del CSIRT para automatizar la documentación de la respuesta de los responsables en los comentarios.

Ayudándose del asunto del email que contiene el UUID del caso, puede realizarse una petición a la API de Ngen y crear un comentario a ese caso en particular con el contenido del email de respuesta.

3.5 Interacciones

Como se describió en la sección 2.2.5, las interacciones con la comunidad o con otros CSIRTs, son actividades recurrentes en el ciclo de vida de un incidente. Establecer un contacto rápido y seguro con estas partes puede cambiar rotundamente la resolución de una investigación.

La búsqueda de contactos y su mantenimiento es una de las funciones más sensibles que un CSIRT puede realizar. Estos no solo son difíciles de conseguir, en casos de contactos externos, sino que también son sensibles al tiempo y cambian, muchas veces, con periodicidad.

Poder mantener una base de datos actualizada de dichos contactos es una tarea que difícilmente puede realizarse de manera manual. Aunque, en la mayoría de los casos, se realiza de esa manera.

3.5.1 Redes

Según lo visto en 2.1.2, la comunidad puede definirse como un conjunto de IPs o dominios. Un CSIRT puede tener la necesidad de definir una o varias comunidades que también pueden ser dinámicas, en el sentido de que puede haber comunidades que se agreguen temporalmente al alcance del CSIRT.

Uno de los puntos en los que falla la mayoría de los sistemas de gestión de incidentes actuales es en la representación e interacción con la comunidad a la que sirven. La comunicación con la comunidad es un punto fundamental en el ciclo de vida de un incidente. Puede afectar directamente a la imagen del CSIRT con su comunidad o con entidades externas, si se comparte información sensible con contactos no pertinentes o no se contacta a la entidad en absoluto por no encontrar, por ejemplo, una dirección de email donde enviar la información.

Es importante tener una representación de la comunidad y un sistema que permita buscar el contacto de cada una de sus entidades. Y también, permitir la actualización de estos contactos y tener una visión actualizada de la comunidad, su topología y contactos.

El objetivo de las redes es representar la comunidad objetivo a través de direcciones IP o dominios. Cada red está compuesta principalmente por una IP CIDR(v4 o v6) o un dominio DNS, y uno o varios métodos de contacto. Estos métodos de contacto pueden ser mail, telegram, teléfono, etc. Los mismos, serán luego los que se utilizarán al momento de comunicar un caso.

Existe 2 tipos de redes:

- **Internas:** Las redes internas son aquellas que pertenecen a la comunidad del CSIRT y representan su topología.
- **Externas:** Las redes externas son aquellas que no pertenecen a la comunidad del CSIRT pero son contactos directos. Por ejemplo, cuando se requiere reportar un incidente a otro CSIRT conocido, se puede agregar las redes o dominios del mismo y así reportar eventos automáticamente.

3.5.1.1 Árbol de redes

Las redes se distribuyen en forma de un árbol N-ario, desde la más genérica a la más específica, y permite al sistema Ngen agregar, modificar y eliminar redes sin perder su orden o estructura.

Esta estructura simplifica varias problemáticas que surgen a la hora de trabajar con datos como direcciones IP y dominios. Ya que estos datos son, por naturaleza, partes de

un árbol. Las direcciones IP se distribuyen en forma de árbol gracias a la segmentación de redes CIDR. Y asimismo, los dominios son parte del árbol DNS.

Existirá un subárbol para cada tipo de red IPv4, IPv6 y Dominio. Además, cualquier IPv4 /32 o IPv6 /128, serán siempre hojas del árbol.

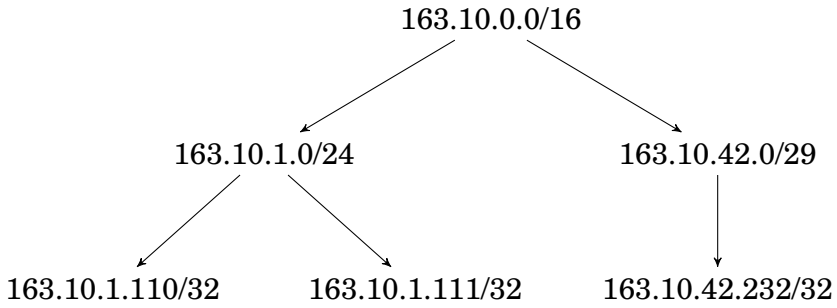

Figura 3.16: Árbol N-ario de redes IPv4

Si no existiera una estructura que represente de la mejor manera su topología, las redes serían objetos con relaciones simples y no permitirían realizar consultas complejas sobre las mismas.

3.5.1.2 Organización automática del árbol

Cuando una red es creada se busca su lugar dentro del árbol. En un caso común de árboles ordenados, la inserción se realizaría recorriendo el árbol recursivamente hasta encontrar el lugar correspondiente. En este caso en particular, las IPs y dominios ya son en sí un árbol y la inserción puede realizarse buscando la red que será padre y luego las redes serán sus hijas. En otras palabras, las redes que contienen a la red creada y qué redes están contenidas en ella.

En el caso de las redes del tipo IP, esta acción se realiza a través de la base de datos PostgreSQL[35] y sus funciones de para redes [36]. Utilizando la librería netfields [37] que abstrae varias de las funciones a alto nivel. Luego de encontrar a la red padre, se buscan las redes hijas de la red padre que estén contenidas en la red creada utilizando la librería netfields.

En el caso de los dominios la búsqueda de padres o hijos es diferente. No existen librerías que solucionen cosas de este tipo ya que el campo Domain no es, en sí, muy complejo. Es solo una cadena de caracteres dividida por puntos y la búsqueda de padres o hijos se da manipulando la cadena de caracteres del dominio.

La búsqueda del padre se realiza dividiendo el dominio en los distintos subdominios que lo componen. Si, por ejemplo, tomamos la red de dominio '**info.unlp.edu.ar**' los distintos subdominios que la componen, serian: '**unlp.edu.ar**', '**edu.ar**', '**ar**'. Entonces se realiza una búsqueda de cada una de estas redes, y se toma la red existente más específica.

La búsqueda de las redes hijas se realiza buscando las redes cuyos dominios terminan en '**info.unlp.edu.ar**' y sean hijas de la red padre. Ya que todos los dominios que terminen con '**info.unlp.edu.ar**', serán subdominios de esta red.

3.5.1.3 Redes huérfanas

La búsqueda de las redes hijas se ve afectada por las redes huérfanas, estas son aquellas redes que no tienen padre y suelen aparecer cuando se empieza a agregar redes individuales en el sistema. La existencia de estas redes obliga a realizar una búsqueda distinta, que difiere de las búsqueda padre/hijo en árboles ya que son redes que no tienen esa estructura.

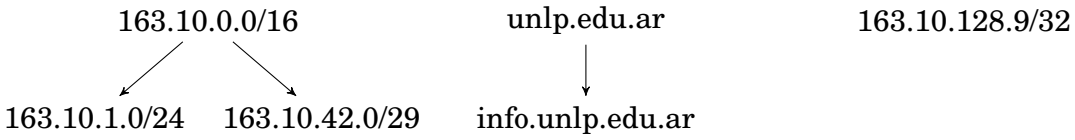

Figura 3.17: Redes huérfanas en árbol de redes

Para evitar la existencia de redes huérfanas, se crea una red Default. Esta red es siempre la raíz del árbol de redes y el padre de todas las redes huérfanas. Esto permite realizar búsquedas homogéneas.

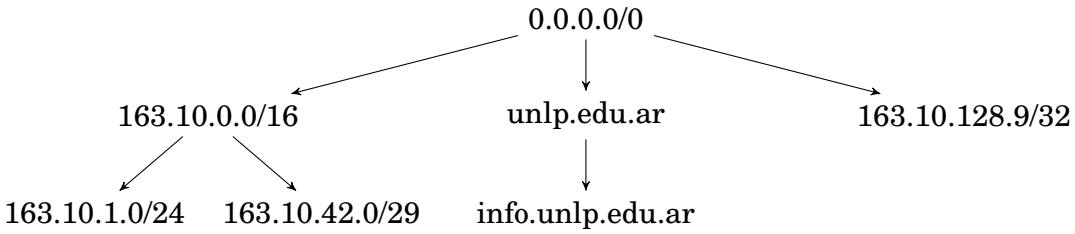

Figura 3.18: Árbol de redes de la red Default

Esta red debe ser lo suficientemente genérica como para ser siempre la resultante de una búsqueda por redes padre. Para cumplir con esto sus campos se componen en CIDR '**0.0.0.0/0**' y dominio **''** (carácter vacío).

3.5.1.4 Entidades de red

Es común encontrar situaciones donde bloques IPs totalmente separados, bloques v4 y v6, o dominios se asignen a un mismo departamento en una organización. Las entidades sirven para agrupar redes contextualmente sin importar su tipo. Esta agrupación también ayuda a generar datos estadísticos. Siguiendo con el ejemplo anterior, podemos contextualizar a las redes de la siguiente manera:

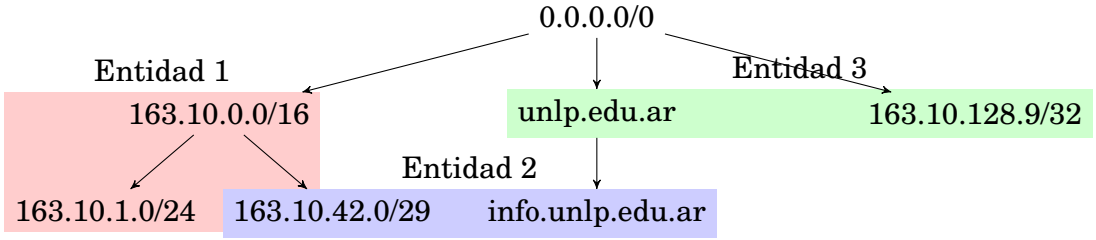

Figura 3.19: Contextualización de redes con entidades de red

3.5.2 Contactos

La búsqueda de contactos para la comunicación de un caso puede realizarse de dos maneras, dependiendo si la comunicación es interna o externa.

Para comunicaciones internas se utilizan las redes. Cuando un caso es reportado, cada uno de sus eventos obtiene su email de contacto a partir de la red a la que pertenece. Si la red no tiene un contacto, se dará el contacto de la red padre o la red por defecto.

Para comunicaciones externas se utilizan las redes y el posible enriquecimiento de los artefactos de un evento. En este caso las direcciones IP o de dominio pueden aportar un contacto de abuse a través de la búsqueda del mismo por el sistema Cortex.

Luego de obtener los contactos, el caso es comunicado a cada uno de ellos. Listando en cada comunicación los eventos específicos de cada contacto para evitar la divulgación de información sensible a las distintas partes de un incidente.

Por ejemplo, puede existir casos donde se comunique tanto a la víctima como a el atacante. Enviar información cruzada en este caso podría terminar en una divulgación de información grave.

3.5.2.1 Tipos de contactos

- **Email:** Una dirección de correo electrónico donde llegarán los distintos tipos de reportes. Además, se puede agregar un archivo de clave pública PGP para que los

reportes sean cifrados.

- **Telegram:** Alias de la cuenta de Telegram y se enviarán mensajes a través de un robot que se configurará desde Ngen.
- **Phone:** Número de teléfono para contactar a los responsables.
- **URI:** dirección web de un formulario de contacto a los responsables.

3.5.2.2 Roles

Los roles definen la relación del contacto con respecto a la red. Es aquí donde los contactos pueden ser diferenciados entre "Contactos relacionados con incidentes" y "Contactos no relacionados con incidentes" explicados en 2.2.5.1.

El RFC 7483 "JSON Responses for the Registration Data Access Protocol" [38] define los distintos roles de las entidades de RDAP:

- **Registrant/Registrante:** Es el registrante del registro. En algunos registros, esto se conoce como mantenedor.
- **technical/Técnico:** Contacto técnico para el registro.
- **administrative/administrativo:** Contacto administrativo para el registro.
- **abuse/abuso:** Contacto que maneja problemas de abuso de red en nombre del registrante del registro. Es el principal tipo de contacto de una entidad para reportar incidentes.
- **billing/facturación:** Contacto que maneja los problemas de pago y facturación en nombre del registrante del registro.
- **registrar/registrador:** Contacto que representa la autoridad responsable del registro en el registro.
- **reseller/revendedor:** Contacto que representa a un tercero a través del cual se realizó el registro,es decir, no el registro o el registrador.
- **sponsor/patrocinador:** Contacto que representa un patrocinador de política de dominio, como un patrocinador aprobado por ICANN.
- **proxy:** Contacto que representa un proxy para otro objeto de entidad, como un registrante.

- **notifications/notificaciones:** Contacto designado para recibir notificaciones sobre instancias de objetos de asociación.
- **noc:** Contacto que maneja las comunicaciones relacionadas con un centro de operaciones de red (NOC).

3.5.2.3 Prioridades

La prioridad, definida en la sección 3.2.3, se utiliza en los contactos para definir la sensibilidad con la cual puede enviarse información al mismo. El contacto recibirá reportes de casos con prioridad mayor a la prioridad asignada. Entonces, un contacto con la prioridad más baja recibirá reportes de todas las prioridades, pero si se le asigna una prioridad alta solo recibiría reportes de prioridades altas o críticas.

CAPÍTULO 4

IMPLEMENTACIÓN Y RESULTADOS

La realización y puesta en marcha de NGEN, la solución desarrollada para esta tesis y descripta en el capítulo anterior, requirió desarrollar la siguiente infraestructura.

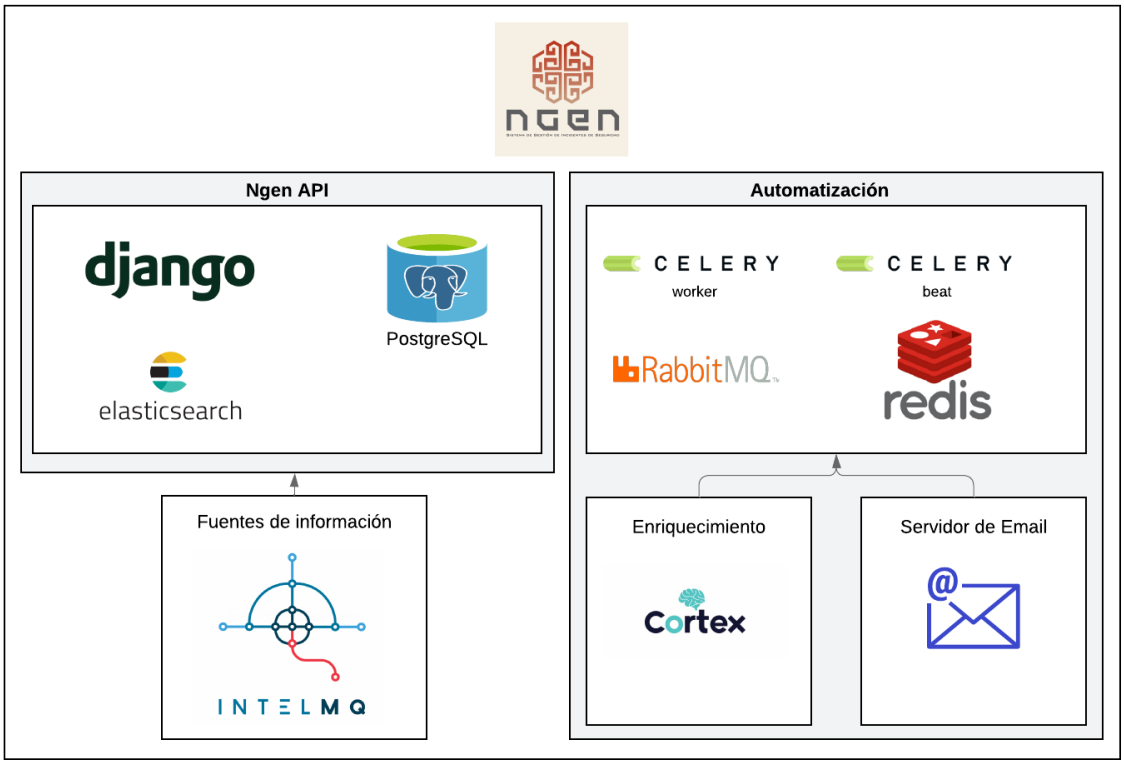

Figura 4.1: Diagrama de infraestructura de Ngen

Como parte principal se encuentra la API REST, que concentra la mayoría de la funcionalidad descripta en el capítulo anterior. Esta interfaz permite que Ngen sea utilizado por otros sistemas de manera simple, mejorando notablemente la automatización de tareas. Uno de estos sistemas es IntelMQ que se comunica con la API para generar eventos a partir de fuentes de información.

Elasticsearch es utilizado como motor de búsqueda. Almacenando información del sistema en paralelo junto a la DB, permite realizar búsquedas dinámicas y eficientes sobre la gran cantidad de información que generan los casos y eventos.

La sección de "Automatización" está conformada por el sistema Celery que es el encargado de realizar todas las tareas automáticas y asincrónicas, como el enriquecimiento a través de Cortex y el envío de reportes por email.

A continuación se detallarán cada una de estas secciones y sus características.

4.1 Ngen API

La API está desarrollada en el framework web Django[39], y su librería Django-rest[40] que permite desarrollar una API REST a través de Django. Junto a la base de datos PostgreSQL[35], que facilita varias funciones de manera nativa como la gestión de direcciones de red o IPs, conforman la base de Ngen.

A continuación se explicaran la implementación de cada una de las soluciones propuestas en el capítulo anterior.

4.1.1 Auditoría

La clase abstracta NgenModel centraliza el comportamiento en común de todas las clases del sistema. Parte de este comportamiento en común es que todas las clases de Ngen son auditadas con el paquete django-auditlog [41] que permite realizar esta acción solo agregando un campo a la clase a auditar.

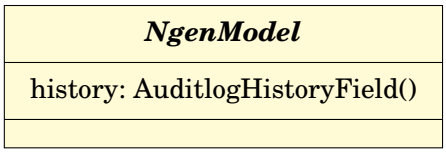

Figura 4.2: Implementación de auditorías

Esta librería guarda automáticamente cualquier cambio en las instancias de la clase en forma de clave : valor viejo / valor nuevo. Además de almacenar el actor de la acción, la fecha y demás detalles.

4.1.2 Evidencia

La gestión de evidencias 3.1.3 es solucionada a través de la clase abstracta NgenEvidenceMixin, encargada crear cada instancia de Evidence a partir de los archivos adjuntados. La clase Evidence es la representación de un archivo de evidencia en el sistema y se relaciona de forma genérica con NgenEvidenceMixin a través de la clase ContentType de Django.

La aplicación ContentType [42] sirve para relacionar modelos con otros modelos, como una clave foránea, pero con la ventaja de que el tipo de modelo con el cual se relacione puede ser diferente. En resumen, ContentType permite generar relaciones genéricas entre modelos.

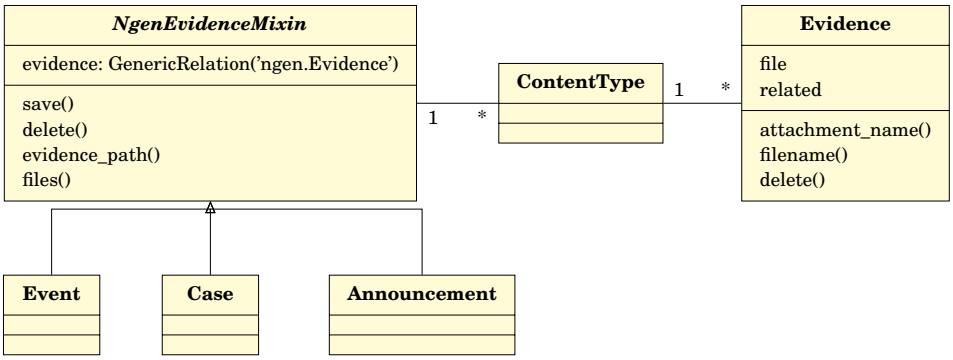

Figura 4.3: Implementación de la gestión de evidencias

4.1.2.1 Almacenamiento de evidencias

La solución al problema de almacenamiento de evidencias repetidas planteada en 3.1.3 es implementada a través de la modificación de la clase "Storage" [43] de Django. Se implementó la clase "HashedFilenameStorage" que realiza un hashing de los archivos de evidencia y los almacena. En caso de que el archivo ya exista en el espacio del objeto se retorna el error correspondiente a través de la API.

4.1.3 Artefactos

La implementación de los artefactos, explicados en 3.1.4, se abstrae utilizando las siguientes clases.

La clase Artifact es la representación de un artefacto en particular, tiene un tipo (ip, dominio, hash, etc.) y un valor. Cada artefacto puede tener varios tipos de enriquecimientos representados por ArtifactEnrichment.

Los artefactos tienen la particularidad de poder relacionar cualquier objeto que los comparta. Este comportamiento es realizado a partir de la Clase ArtifactRelation, que relaciona N artefactos con N clases, que pueden ser heterogéneas, a través de una relación genérica. Esto permite a cualquier clase de Ngen generar una relación N a N con artefactos. La clase abstracta ArtifactRelated es la encargada de realizar dicha relación.

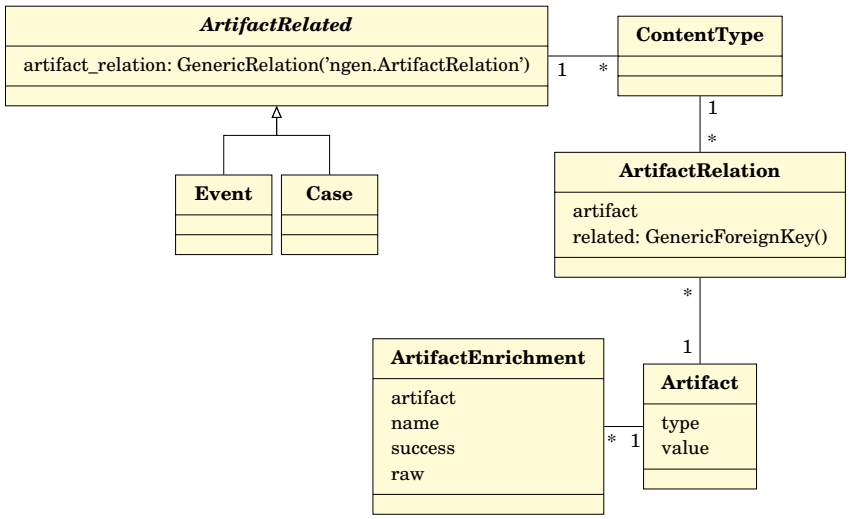

Figura 4.4: Implementación de artefactos

4.1.4 Clasificación

4.1.4.1 Árboles y combinación

Varios de los objetos detallados anteriormente tienen una estructura de árbol y comportamiento similar. Para simplificar el desarrollo de estas clases abstractas se crearon NgenTreeModel y NgenMergeableModel.

La clase abstracta NgenTreeModel, define una relación con sigo misma llamada "parent" que es la que permitirá generar la estructura en árbol. El comportamiento es heredado del paquete django-treebeard [44] y su clase "AL_Node" ("Adjacency List Node"

o nodo lista de adyacencias) que abstrae muchos métodos para el uso de objetos en estructuras de árbol.

La clase abstracta NgenMergeableModel, es la encargada del comportamiento de combinación de los casos y eventos 3.2.1 discutidas en el capítulo de Clasificación. Hereda el comportamiento de árbol de NgenTreeModel y agrega las reglas de combinación también detalladas en 3.2.1.

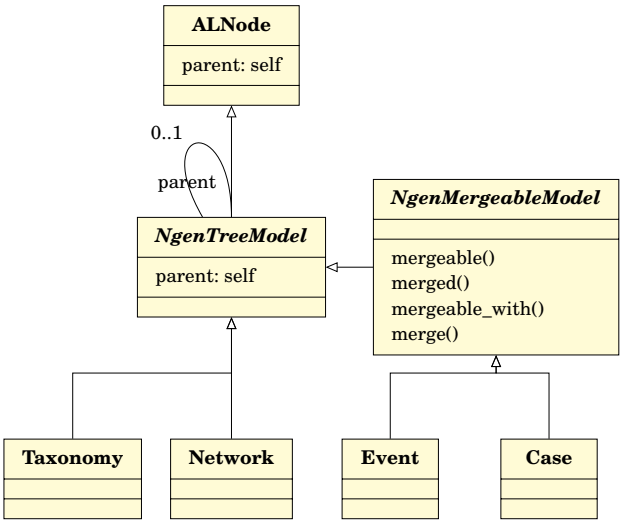

Figura 4.5: Implementación de árboles y combinación

4.1.4.2 Prioridad

El objeto Priority de Ngen abstrae las definiciones de prioridad vistas en 3.2.3. Se realizaron algunos cambios para mejorar la interpretación desde el punto de vista del incidente.

ITIL	Ngen
Priority Code	Severity
Response Time	Attend Time
Resolution Time	Solve Time

Table 4.1: Implementación de atributos de ITIL en Ngen

El atributo "Priority code" puede también verse como la representación de la severidad de la prioridad. Cuanto menor es el código, mayor es la severidad. Los tiempos de respuesta pasan a la nomenclatura de attended y solved visto en la abstracción de los estados 3.3 representando los tiempos cuando un caso es atendido o resuelto.

La clase abstracta NgenPriorityMixin, es la encargada de la relación con la clase Priority y de asignar la prioridad por defecto en caso que no haya sido agregada. Esto último se realiza a través del método de clase default_priority() que busca y retorna una instancia de Priority a partir del valor por defecto configurado en Ngen.

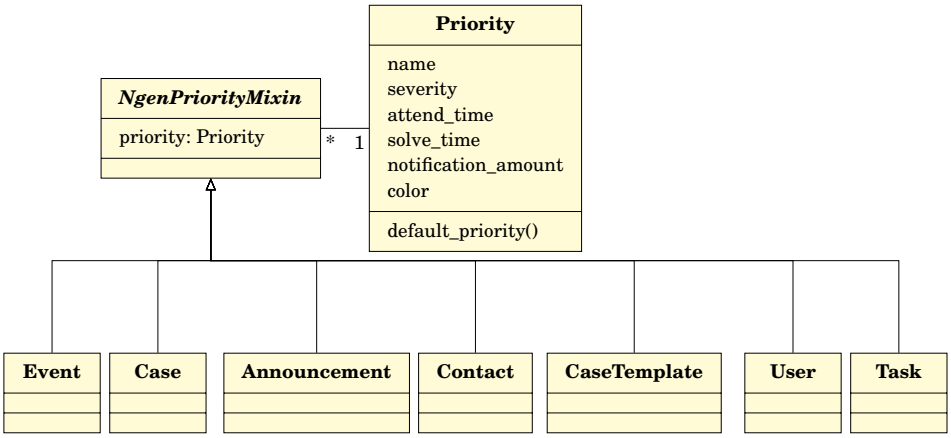

Figura 4.6: Implementación de prioridades

4.1.4.3 Templates

Los Templates discutidos en 3.2.6, son implementados por la clase CaseTemplate.

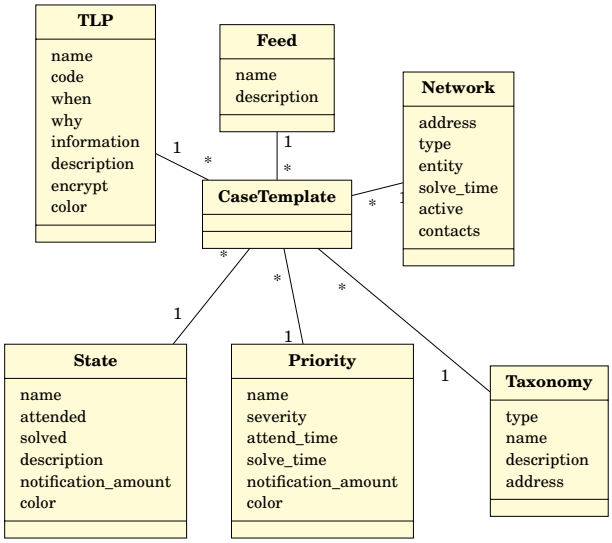

Figura 4.7: Implementación de templates

4.1.4.4 Taxonomías y feeds

Una taxonomía puede tener varias playbooks, y una playbook estar compuesta de varias tareas a realizar. Las tareas se irán completando por los operadores según avancen con la investigación de un caso. Los casos tendrán eventos de taxonomías específicas y estas tendrán playbooks con tareas a realizar. Estas tareas deben completarse para cada uno de sus eventos.

Esto crea una relación entre eventos y tareas, en donde se registra la concreción de las mismas, comentarios y tiempos de resolución. A esta relación la llamaremos TodoTask o tareas a realizar. Los reportes están representados por la clase Report y las fuentes de información por la clase Feed.

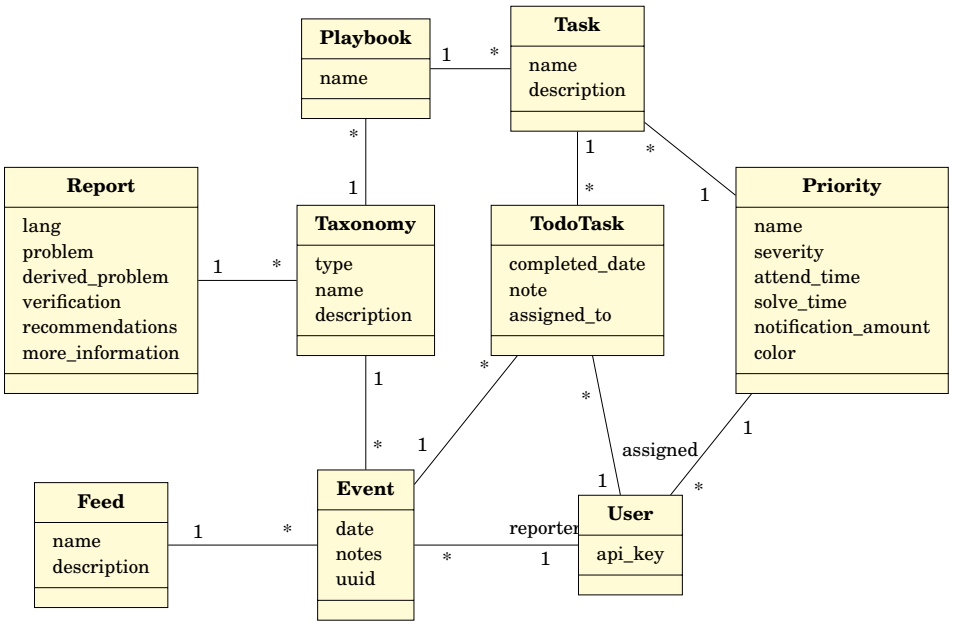

Figura 4.8: Implementación de taxonomías, reportes, playbooks y feeds

4.1.5 Gestión

El grafo de estados es implementado a través de las clases State, que representa los estados, y la clase Edge, que representa las aristas entre ellos.

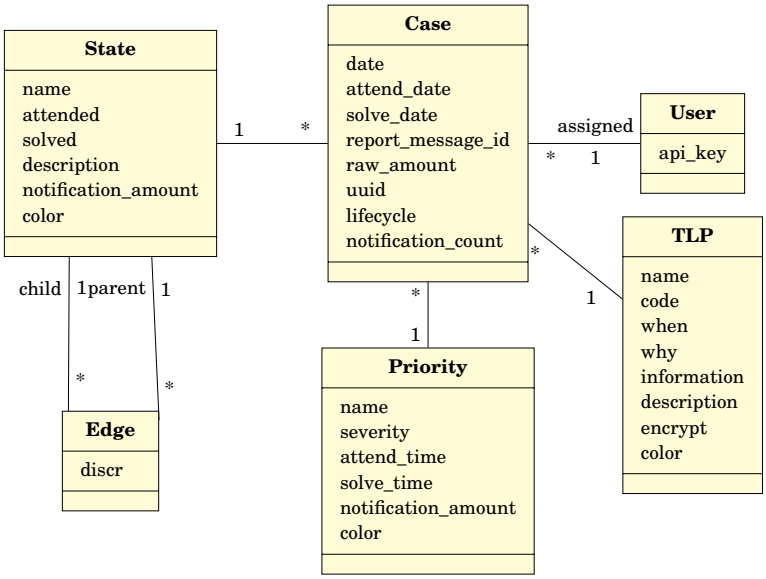

Figura 4.9: Implementación de grafo de estados

4.1.6 Anuncio

La clase abstracta Communication permite, a cualquier clase que herede de ella, enviar un email. La clase Announcement utiliza esta interfaz para realizar comunicaciones a toda la comunidad o parte de ella. Los casos la utilizan en su ciclo de vida para comunicar las acciones tomadas y los reportes correspondientes.

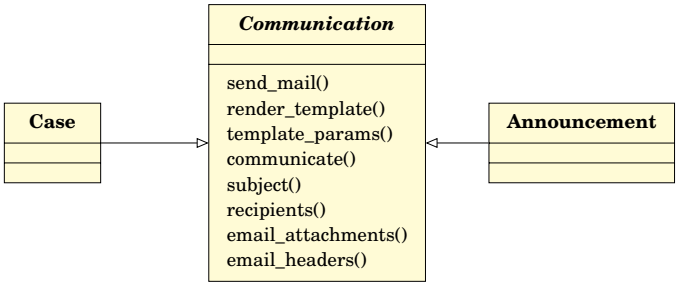

Figura 4.10: Implementación de comunicación

4.1.7 Retroalimentación

Los comentarios se implementan a través de la clase comment y una relación genérica desde caso y evento.

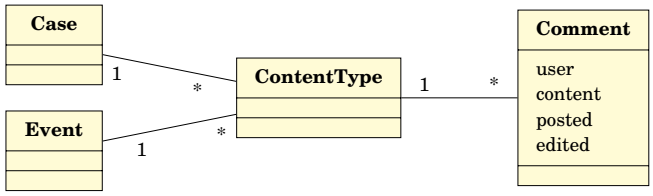

Figura 4.11: Implementación de los comentarios

4.1.8 Interacciones

4.1.8.1 Redes y contactos

El árbol de redes es implementado a través de la clase Network que hereda su comportamiento de NgenTreeModel. Las entidades que dan contexto a las redes son implementadas por NetworkEntity. Y por último los contactos implementados por Contact y Priority.

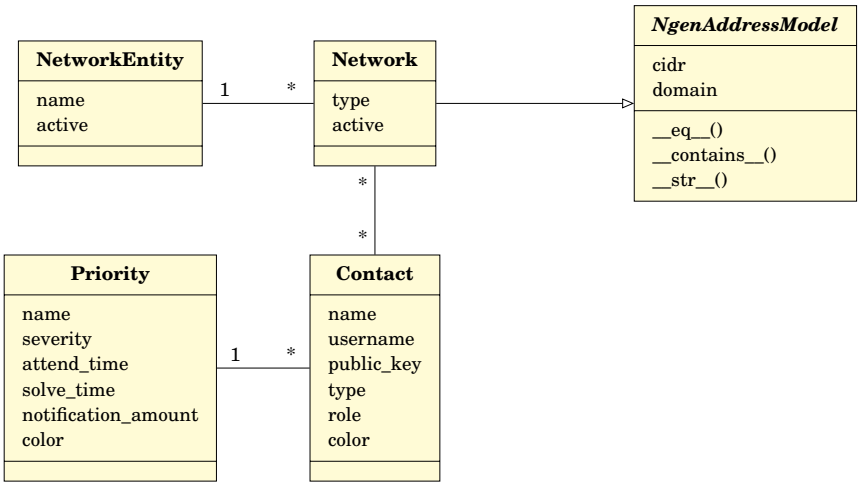

Figura 4.12: Implementación de redes y contactos

4.1.8.2 Address

El objeto Address es una abstracción de los campos CIDR y dominio, vistos en redes y eventos, simplificando las operaciones con estos campos. Unifica la interacción de IPs y dominios, evitando preguntar con que campo estamos trabajando en un momento específico. Permite comparar la igualdad o inclusión de las redes sin importar si es CIDR o dominio. Además, permite asignar y obtener las direcciones de forma transparente.

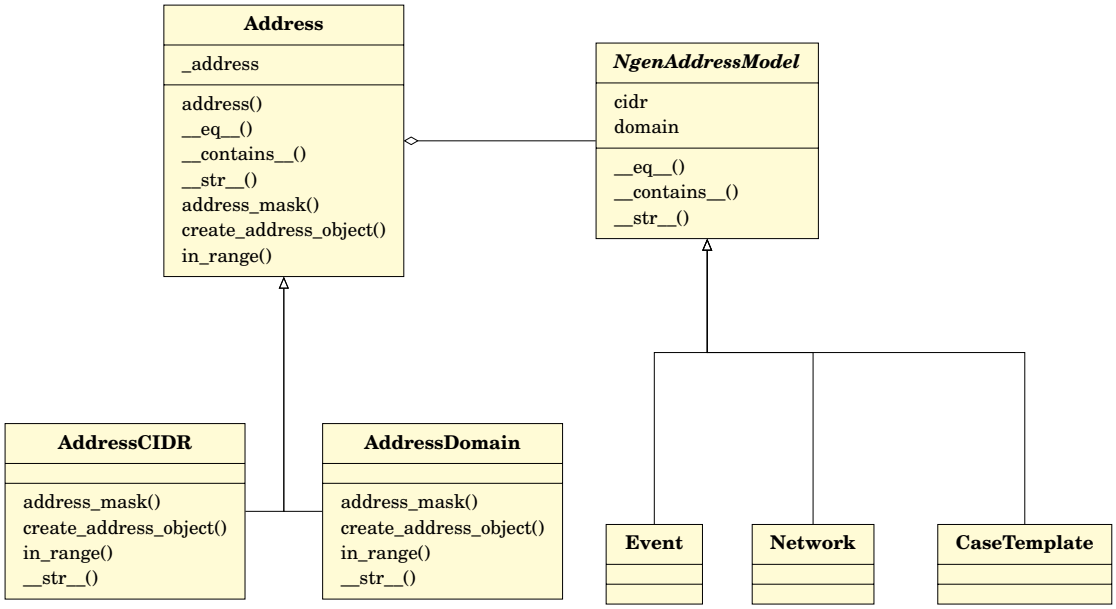

Figura 4.13: Implementación del objeto address

4.1.9 PyNgen

La librería PyNgen [45], desarrollada en Python y liberada como software libre por el CERT UNLP, abstrae la comunicación con la API de Ngen, brindando una interfaz de alto nivel, permitiendo realizar operaciones básicas sobre casos y eventos.

4.2 Automatización

Tareas como el envío de correo electrónico o el enriquecimiento de artefactos pueden tardar algunos segundos, por lo tanto la comunicación con la API se vería afectada al esperar que estas tareas se realicen. También, podrían fallar si el servicio que se utiliza está caído, apagado o falla por alguna razón. Si esto fuese así, el resultado de la petición de la API podría devolver un error o no retornar nada en absoluto..

4.2.1 Celery

Celery [46] es un gestor de tareas distribuido y asincrónico desarrollado en Python. Utiliza colas de tareas como un mecanismo para distribuir el trabajo entre subprocesos o máquinas. La entrada de una cola de tareas es una unidad de trabajo denominada tarea

o "task". Los procesos de trabajo, o "workers", dedicados monitorean constantemente las colas de tareas para realizar nuevos trabajos.

Celery se comunica a través de mensajes, generalmente utilizando un intermediario, o "brocker", para mediar entre clientes y trabajadores. Para iniciar una tarea, el cliente agrega un mensaje a la cola, el intermediario luego entrega ese mensaje a un trabajador.

4.2.1.1 RabbitMQ

RabbitMQ [47] es un intermediario o "broker" para mensajería. Es el broker por defecto y recomendado por Celery. Brinda a sus aplicaciones una plataforma común para enviar y recibir mensajes, y sus mensajes un lugar seguro para vivir hasta que se reciban.

4.2.1.2 Redis

Celery también permite guardar los resultados de cada "worker". Esto facilita la resolución de problemas o bugs cuando un trabajador falla. De otra forma no habría manera de saber si el resultado de un trabajador fue exitoso o no.

Redis [48] es un almacenamiento clave/valor, lo que lo hace muy eficiente para obtener los resultados de una llamada de tarea.

4.2.1.3 Tareas periódicas con Celery Beat

Celery beat es un planificador o "scheduler". Inicia tareas en intervalos regulares, que luego ejecutan los trabajadores.

Hay varias tareas que no son realizadas "on the fly" o bajo demanda, algunas tareas son ejecutadas periódicamente cada cierto tiempo.

Por ejemplo, el ciclo de vida automático, explicado en 3.3.1, permite abrir, cerrar y notificar casos automáticamente.

Esto se realiza a partir de tareas configuradas y ejecutadas por Celery cada 1 minuto.

4.2.2 Enriquecimiento con Cortex

Una de las principales dificultades a la hora de trabajar con enriquecimientos es que cada servicio o herramienta devuelve un resultado distinto. Esta heterogeneidad hace, a veces, difícil la tarea de almacenar este tipo de información.

Una de las herramientas que más ha resuelto este tipo de problemáticas es Cortex, que ya se ha detallado en 2.3.1.2.

Como se describió anteriormente, Cortex utiliza varios servicios de enriquecimiento y cada uno de estos tiene un analizador asignado. Estos pueden enriquecer y crear nuevos artefactos a partir del original.

Para poder realizar un enriquecimiento sobre un artefacto se debe crear un Job, estos objetos son la instancia de ejecución de los analizadores. Cuando un Job es creado, Cortex lo ejecuta de forma asincrónica y luego guarda el resultado por un tiempo, que puede ser configurado. Esto sirve de cache para consultas consecutivas del mismo artefacto.

Utilizando la librería de Python Cortex4py [49], desarrollada por creadores de Cortex, podemos comunicarnos con la API del sistema de una manera transparente. Esta librería nos permite ejecutar "Jobs" de forma asincrónica y trabajar con sus resultados a través de objetos con interfaces homogéneas.

Al resultado de la ejecución de un Job se lo denomina "Report". Este objeto permite saber si el Job ha sido ejecutado o no y si ha sido ejecutado con éxito para luego obtener el resultado JSON. Este resultado puede contener también otros artefactos obtenidos en el enriquecimiento.

Como se puede apreciar, la utilización del servicio Cortex con su librería Cortex4py permite solucionar fácilmente varias de las problemáticas planteadas anteriormente.

Luego de tener los resultados y los nuevos artefactos, el sistema de Ngen guarda los resultados en la base de datos y vuelve a enriquecer a los nuevos artefactos para generar aún mas información.

4.3 Estadísticas

Otra de las ventajas de tener una base de datos paralela como Elasticsearch es su facilidad para ser graficada.

A partir de la herramienta de gráficos estadísticos Grafana [50], se pueden realizar estadísticas y gráficos ad hoc.

Permitiendo la creación de tableros de control que simplifican el análisis de datos, brindando la visualización gráfica de la gran cantidad de información que puede resultar de la gestión de incidentes.

4.4 Fuentes de información

La herramienta IntelMQ, anteriormente nombrada en 3.2.4.1, es la principal fuente de información de Ngen. Todas las fuentes de información son agregadas a este sistema,

luego normalizadas y enviadas a la API de Ngen como nuevos eventos.

4.4.1 IntelMQ

IntelMQ [3] es una solución para equipos de seguridad de TI (CERT y CSIRT, SOC, departamentos de abuso, etc.) para recopilar y procesar fuentes de seguridad mediante un protocolo de cola de mensajes.

Es una iniciativa impulsada por la comunidad llamada IHAP (Proyecto de automatización de gestión de incidentes o Incident Handling Automation Project) que fue diseñada conceptualmente por CERT europeos.

La estructura de IntelMQ se basa en el uso de sus entidades básicas, los Bots. y como pueden ser relacionados unos con otros consumiendo sus entradas y salidas.

IntelMQ tiene 4 tipos de bots:

1. Collectors: producen mensajes y los transmiten al sistema. Son los bots que recolectan información de distintos servicios como FTP [1], emails o APIs de fuentes de inteligencia.

2. Parsers: Convierten datos no estructurados en mensajes estructurados. Convierten fuentes disponibles públicamente a un formato que IntelMQ pueda entender.

3. Experts: Operan sobre datos analizados y los enriquecen o modifican.

4. Outputs: Envían datos analizados a otros sistemas, como por ejemplo Ngen.

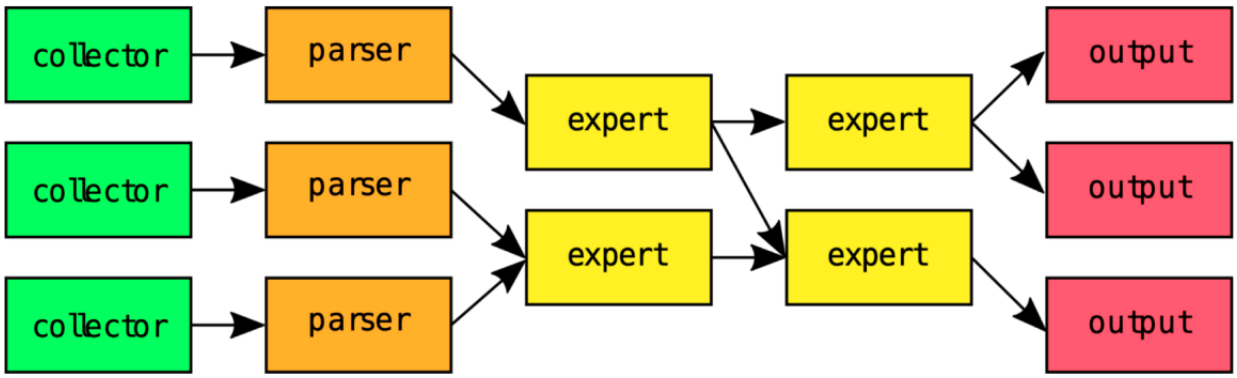

Figura 4.14: Arquitectura de entidades básicas de IntelMQ [51]

[1] FTP: File Transfer Protocol

La conexión con este sistema ha permitido el desacople completo entre el análisis y normalización de las fuentes de información.

4.5 Código y repositorios

Para el desarrollo del conjunto de las ideas planteadas en este capítulo fue utilizada la herramienta de versionado de código libre Git [52] y la plataforma Github. El sistema Ngen es de libre distribución y se encuentra bajo la licencia GNU v3 [20].

4.5.1 Ngen PHP

La primera versión de Ngen, desarrollada en el lenguaje de programación PHP y el framework Symfony [53], fue publicada en el repositorio `https://github.com/CERTUNLP/ngenbundle` [54] de Github el 31 de Agosto del 2015, luego de un año de desarrollo en repositorios privados del CERT UNLP y se realizó mantenimiento hasta el 21 de Mayo del 2021. Actualmente este repositorio se encuentra archivado y de solo lectura.

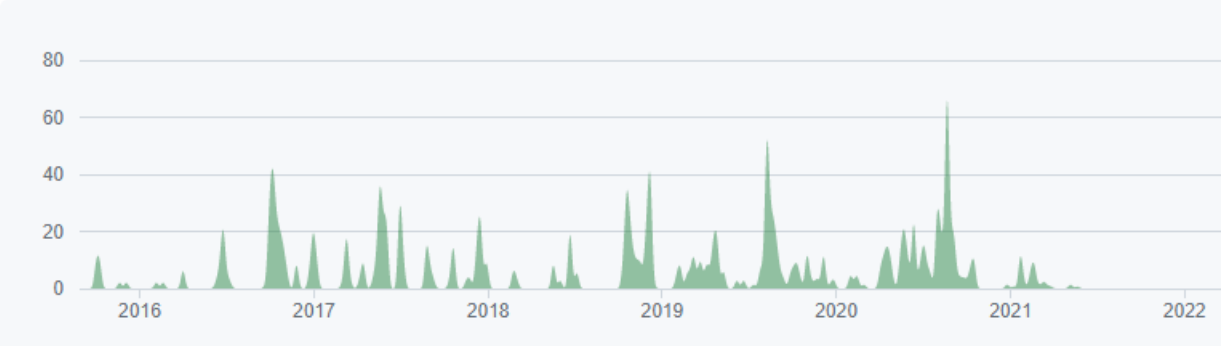

Figura 4.15: Gráfico de commits por año de Ngen PHP

4.5.2 Ngen Python

La actual versión de Ngen, detallada en este documento, fue desarrollada en el lenguaje de programación Python utilizando el framework Django [39]. Todo el trabajo se encuentra publicado en forma de software libre y abierto a la comunidad en general en el repositorio `https://github.com/CERTUNLP/ngen` [55] de Github. Siendo publicados sus primeros commits luego de la finalización del mantenimiento de la versión anterior de Ngen, en Junio del 2021.

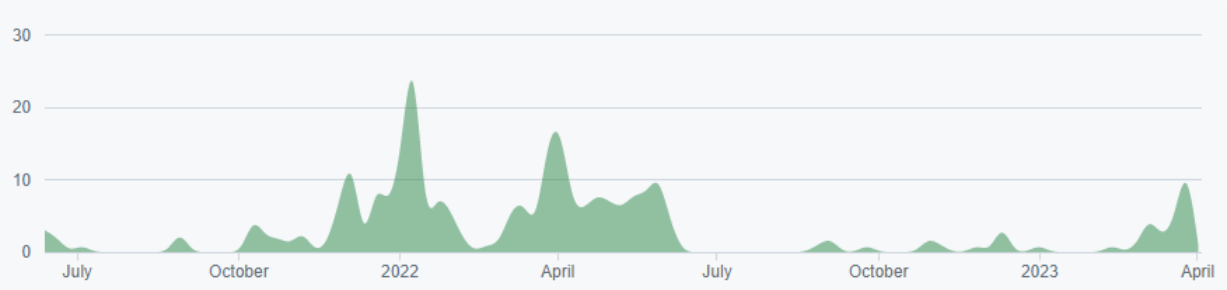

Figura 4.16: Gráfico de commits por año de Ngen

4.5.3 Docker

Ambas versiones de Ngen pueden ser instaladas utilizando el ambiente de contenedores Docker [56] y su herramienta Docker Compose [57], que permite configurar varios contenedores en conjunto y conectarlos de manera transparente. La utilización de contenedores facilita el ciclo de vida del desarrollo de Ngen y evita que quien quiera utilizar o probar Ngen, deba administrar un servidor para poder hacerlo.

Ademas, las imágenes de Docker generadas para producción son publicadas en el perfil del CERT UNLP [58] de Docker Hub [59], la librería de contenedores de Docker.

CAPÍTULO 5

CONCLUSIONES

Como integrante del CERT UNLP por varios años, me he encontrado muchas veces con el problema de responder un incidente de manera rápida y eficiente. Aunque, con el correr del tiempo, pudimos observar que muchos de estos incidentes y sus eventos relacionados, se repetían y eran predecibles. Como equipo pudimos detallar, aquellos eventos que se comportaban de manera similar y aprender de ellos.

En ese entonces, año 2014, todas las herramientas para la gestión de incidentes eran solo sistemas de tickets, que no cumplen con todas las necesidades requeridas por un CSIRT. Estas herramientas no nos permitían desarrollar nuestras ideas de manera eficiente y automática.

Ngen surge como un reflejo de las necesidades de nuestro contexto laboral y profesional. No solo soluciona muchas de nuestras tareas diarias sino que permite tener una visión amplia de los eventos de nuestra comunidad. Además, mejora el bienestar laboral gracias a la automatización de varias tareas cotidianas y repetitivas, que nos permiten realizar otras tareas más complejas en paralelo.

Los sistemas más populares como TheHive y RTIR, no cumplen con los requisitos de un CSIRT en su totalidad. Esto significa que aún hoy en día y, al menos desde el 2014, los sistemas de gestión de incidentes no completan las necesidades de un CSIRT, y es por eso que Ngen actualmente sigue siendo un aporte importante al mundo de la gestión de incidentes.

5.0.1 Desafíos

El desarrollo de Ngen tuvo varios desafíos. En un principio solo se contaba con un pequeño sistema en PHP [1] plano, que solo cumplía con algunos casos de uso diarios. Pero no tenía capacidad alguna de automatización ni escalabilidad. Luego, se realizó la primera versión de Ngen con con el framework Symfony [53] de PHP, que ya contemplaba muchas de las funciones que hoy tiene el Ngen actual.

Luego de varias puestas a prueba en ámbitos académicos y gubernamentales, se decidió crear una nueva versión que materializara toda esa experiencia y además sea desarrollado con las últimas tecnologías. Es en ese momento, la OEA crea el "Fondo de innovación en ciberseguridad" [60] A los cuales Ngen es propuesto y termina siendo beneficiado [61] con una beca para realizar la nueva versión con todas sus mejoras.

Uno de los principales cambios fue el lenguaje de programación, hubo que trasladar las soluciones desarrolladas en PHP a un lenguaje totalmente distinto como Python. Esta nueva versión, la cual es detallada en esta tesina, se desarrolló en completamente en Python utilizando el framework Django. Este cambio no solo trajo mejoras en el desarrollo, que es mucho más simple, sino también en su performance.

La manipulación de datos datos como IPs, se volvió mucho más sencilla gracias a la base de datos PostgreSQL [35] y sus funciones nativas para direcciones de redes IP.

La cantidad de líneas de código necesarias para el correcto funcionamiento del sistema bajó considerablemente ya que mucha funcionalidad, que antes requirió trabajo manual de código, ahora es resuelta por parte de las librerías nativas de Python.

5.0.2 Experiencias

En todos estos años Ngen fue probado en varios contextos, desde académicos, en la UNLP, como gubernamentales, en el CSIRT nacional de Jamaica.

En el 2018 Ngen fue parte de una misión de la OEA [2] [62] en el Gobierno de Jamaica, donde se instalaron instancias de Ngen para la gestión de incidentes de ese gobierno. Dicha misión tuvo soporte hasta el año 2021.

De cada una de estas experiencias se han reunido requerimientos fundamentales para la adaptación de Ngen a casi cualquier contexto o ámbito de la gestión de incidentes que fueron plasmadas en implementación de esta tesina.

[1] PHP: lenguaje de programación web
[2] OEA: Organización de los Estados Americanos

5.0.3 Formación

Este sistema ayudó a mi formación profesional desde dos lugares fundamentales. El primero como analista de sistemas pude abstraer todos los procesos de gestión de incidentes con los que trabajábamos diariamente y plasmarlos en un sistema informático que puede realizarlos de manera automática. Además, el uso de las últimas tecnologías y lenguajes de programación nuevos brindaron muchísima experiencia curricular.

El segundo, mi experiencia como analista de seguridad creció considerablemente al poder entender en detalle todos los desafíos y complejidades que la gestión de incidentes puede tener sobre una organización y su comunidad. No solo en nuestro contexto sino también en ambientes nacionales, con todas sus diferencias y similitudes con nuestro ambiente académico.

5.0.4 Comunidad

Es probado, que Ngen se distingue del resto de los sistemas de gestión por su capacidad de documentar a la comunidad a la que pertenece y su comunicación con la misma.

La documentación dinámica de los distintos bloques de red con sus detalles de contacto, que pueden fácilmente ser administrados, permite tener una visión topológica de la comunidad. También posibilita una comunicación óptima al generar un feedback acorde.

5.0.5 Código abierto

En CERTUNLP creemos que el uso de herramientas open source ha sido de vital importancia en nuestro desarrollo tecnológico. Es por eso que siempre hemos abierto todas las soluciones informáticas que hemos realizado y que hoy nos funcionan cómo base para realizar todas las tareas de nuestro CSIRT de manera eficiente. Es por esto que creemos de vital importancia la transparencia con la comunidad, transmitiendo todos los conocimientos aprendidos.

Ngen fue liberado bajo la licencia de copyleft GPL v3 y es nuestra contribución al mundo de la gestión de incidentes.

5.0.6 Conclusiones personales

Ngen cumple con todas las necesidades planteadas en este documento como: la integración con otros sistemas, fuentes de información, enriquecimiento, automatización

de la gestión y comunicación con la comunidad. Considero que se han logrado todos los objetivos planteados en este documento.

5.1 Trabajo futuro

A continuación se presentan varios puntos para trabajar en el futuro, que aportarían mejoras.

5.1.1 Cifrar y firmar mensajes

Aplicar cifrado de emails para comunicaciones TLP RED o cualquier información sensible. Además, firmar cada comunicación con una clave privada para asegurar la veracidad de los mismos y generar confianza con la comunidad que recibe dichos mensajes.

5.1.2 Generar más modularidad en el código

Con una modularidad mayor se puede configurar a Ngen para activar y desactivar módulos bajo demanda.

Por ejemplo, desactivar el manejo de redes, sistema de estados, enriquecimiento,etc.

Esto permite que el sistema se adapte mejor a los distintos contextos de cada CSIRT.

5.1.3 Integración con sistemas CMBD

Los sistemas Configuration Management Data Base o CMDB, se utilizan para almacenar los activos de una empresa junto a sus configuraciones, topología, dueños, responsables y hasta lugar físico en la oficina.

Este tipo de sistemas son muy utilizados hoy en día por grandes organizaciones para mantener actualizada su estructura de activos y los responsables de cada uno.

Integrando Ngen con este tipo de información puede facilitar o reemplazar en uso del módulo Network.

5.1.4 Conexión entre instancias de Ngen

Una función interesante, inspirada en el sistema LUCIA 2.3.4, es la sincronización entre distintas instancias.

Es común que las organizaciones se compongan de varias sub organizaciones que pueden no tener visibilidad entre ellas o pueden tener visiones de la red completamente distintas.

Cuando un CSIRT ofrece sus servicios a este tipo de organizaciones, puede no tener una visión completa de su comunidad y podría ser necesario administrar varias instancias de Ngen para poder solventar esta falta de visibilidad.

5.1.5 Liberarlo

Una meta importante es la liberación de Ngen a cualquier entidad u organización, académica o gubernamental, que desee utilizarlo. Que sea de uso para la sociedad.

5.1.6 Internacionalización

Todo sistema utilizado internacionalmente debe considerar el soporte de varios lenguajes.

BIBLIOGRAFÍA

[1] Best Practical Solutions.
RT for Incident Response. Best Practical Solutions.
URL: https://bestpractical.com/rtir (visited on 02/19/2023).

[2] MISP.
MISP Open Source Threat Intelligence Platform and Open Standards For Threat Information Sharing. MISP Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing.
URL: https://www.misp-project.org/ (visited on 02/19/2023).

[3] CEF framework.
certtools/intelmq.
original-date: 2014-06-24T10:11:39Z.
Feb. 18, 2023.
URL: https://github.com/certtools/intelmq (visited on 02/20/2023).

[4] TheHive.
TheHive Project.
URL: https://www.thehive-project.org (visited on 01/30/2021).

[5] Cespi.
CERTUNLP. CERTUNLP.
URL: https://www.cespi.unlp.edu.ar/cert (visited on 02/19/2023).

[6] Georgia Killcrece et al.
State of the Practice of Computer Security Incident Response Teams (CSIRTs):
Fort Belvoir, VA: Defense Technical Information Center, Oct. 1, 2003.
DOI: 10.21236/ADA421664.
URL: http://www.dtic.mil/docs/citations/ADA421664 (visited on 01/07/2021).

[7] Paul R Cichonski et al.
"Computer Security Incident Handling Guide".

In: (), p. 79.
URL: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-61r2.pdf.

[8] Moira West-Brown et al.
Handbook for Computer Security Incident Response Teams (CSIRTs).
report.
Carnegie Mellon University, Apr. 1, 2003.
DOI: 10.1184/R1/6574055.v1.
URL: https://kilthub.cmu.edu/articles/report/Handbook_for_Computer_Security_Incident_Response_Teams_CSIRTs_/6574055/1 (visited on 10/20/2022).

[9] FIRST.
About FIRST. FIRST — Forum of Incident Response and Security Teams.
URL: https://www.first.org/about (visited on 11/06/2022).

[10] FIRST.
CSIRT Services Framework Version 2.1. FIRST — Forum of Incident Response and Security Teams.
Nov. 2019.
URL: https://www.first.org/standards/frameworks/csirts/csirt_services_framework_v2.1 (visited on 02/19/2023).

[11] Scott Hollenbeck and Andy Newton.
Registration Data Access Protocol (RDAP) Object Tagging.
Request for Comments RFC 8521.
Internet Engineering Task Force, Nov. 2018.
URL: https://datatracker.ietf.org/doc/rfc8521 (visited on 02/19/2023).

[12] Leslie Daigle.
WHOIS Protocol Specification.
Request for Comments RFC 3912.
Internet Engineering Task Force, Sept. 2004.
URL: https://datatracker.ietf.org/doc/rfc3912 (visited on 02/19/2023).

[13] CCN-CERT.
LUCIA.
URL: https://www.ccn-cert.cni.es/soluciones-seguridad/lucia.html (visited on 02/19/2023).

[14] CERT Société Générale.
FIR (Fast Incident Response).
original-date: 2015-03-11T16:23:34Z.
Feb. 18, 2023.
URL: https://github.com/certsocietegenerale/FIR (visited on 02/19/2023).

[15] StrangeBee.
Cortex - Overview. StrangeBee.
URL: https://www.strangebee.com (visited on 02/19/2023).

[16] CERT Banque de France.
CERT - Banque de France.
URL: https://cert.banque-france.fr/static/home.html (visited on 02/21/2023).

[17] Jérôme Leonard.
TheHive, Cortex and MISP: How They All Fit Together. TheHive Project.
June 19, 2017.
URL: https://blog.thehive-project.org/2017/06/19/thehive-cortex-and-misp-how-they-all-fit-together/ (visited on 02/20/2023).

[18] strangebee.
thehive4py.
original-date: 2016-12-16T10:53:47Z.
Feb. 15, 2023.
URL: https://github.com/TheHive-Project/TheHive4py (visited on 02/19/2023).

[19] Best Practical Solutions.
Request Tracker. Best Practical Solutions.
URL: https://bestpractical.com/request-tracker (visited on 02/20/2023).

[20] GNU.
The GNU General Public License v3.0 - GNU Project - Free Software Foundation.
URL: https://www.gnu.org/licenses/gpl-3.0.html (visited on 02/28/2023).

[21] Best Practical Solutions.
Constituencies - RTIR 5.0.3 Documentation - Best Practical.
URL: https://docs.bestpractical.com/rtir/5.0.3/Constituencies.html (visited on 02/20/2023).

[22] CCN-CERT.
CCN-CERT.
URL: https://www.ccn-cert.cni.es/ (visited on 02/20/2023).

[23] CCN-CERT.
LUCIA Presentación.
May 2015.
URL: https://www.ccn-cert.cni.es/documentos-publicos/877-lucia-presentacion/file.html.

[24] CERT Société Générale.
CERT Société Générale.
URL: https://cert.societegenerale.com/ (visited on 02/20/2023).

[25] CERT Société Générale.
FIR Wiki. GitHub.
URL: https://github.com/certsocietegenerale/FIR/wiki/Home (visited on 02/20/2023).

[26] Julie Ryan.
Leading Issues in Information Warfare and Security Research.
Google-Books-ID: oukNfumrXpcC.
Academic Conferences Limited, 2011.
241 pp.
ISBN: 9781908272089.

[27] Paul J. Leach, Rich Salz, and Michael H. Mealling.
A Universally Unique IDentifier (UUID) URN Namespace.
Request for Comments RFC 4122.
Internet Engineering Task Force, July 2005.
URL: https://datatracker.ietf.org/doc/rfc4122 (visited on 02/20/2023).

[28] ITIL.
ITIL - ITIL.
URL: https://www.itlibrary.org/ (visited on 02/20/2023).

[29] ITIL.
Checklist Incident Priority | IT Process Wiki. IT Process Wiki - the ITIL® Wiki.
URL: https://wiki.en.it-processmaps.com/index.php/Checklist_Incident_Priority (visited on 02/20/2023).

[30] ENISA.
Reference Incident Classification Taxonomy. ENISA.
URL: https://www.enisa.europa.eu/publications/reference-incident-classification-taxonomy (visited on 03/02/2023).

[31] MITRE.
MITRE ATT&CK®.
URL: https://attack.mitre.org/ (visited on 03/02/2023).

[32] MISP.
taxonomies and classification as machine tags.
URL: https://www.misp-project.org/taxonomies.html#_mapping_of_taxonomies (visited on 02/20/2023).

[33] FIRST.
Traffic Light Protocol (TLP). FIRST — Forum of Incident Response and Security Teams.
Aug. 2022.
URL: https://www.first.org/tlp (visited on 02/21/2023).

[34] INCIBE.
Traffic Light Protocol (TLP). INCIBE-CERT.
June 16, 2016.
URL: https://www.incibe-cert.es/tlp (visited on 02/21/2023).

[35] PostgreSQL Global Development Group.
PostgreSQL. PostgreSQL.
Feb. 20, 2023.
URL: https://www.postgresql.org/ (visited on 02/20/2023).

[36] PostgreSQL Global Development Group.
8.9. Network Address Types. PostgreSQL Documentation.
Feb. 9, 2023.
URL: https://www.postgresql.org/docs/14/datatype-net-types.html (visited on 02/20/2023).

[37] jimfunk.
Django PostgreSQL Netfields.
original-date: 2010-04-21T18:55:48Z.
Dec. 21, 2022.

URL: https://github.com/jimfunk/django-postgresql-netfields (visited on 02/20/2023).

[38] Andy Newton and Scott Hollenbeck.
JSON Responses for the Registration Data Access Protocol (RDAP).
Request for Comments RFC 7483.
Internet Engineering Task Force, Mar. 2015.
URL: https://datatracker.ietf.org/doc/rfc7483 (visited on 02/20/2023).

[39] Django.
Django. Django Project.
URL: https://www.djangoproject.com/ (visited on 03/06/2023).

[40] Encode.
Django REST framework.
URL: https://www.django-rest-framework.org/ (visited on 03/06/2023).

[41] Jazzband.
django-auditlog.
original-date: 2013-10-20T12:10:40Z.
Mar. 13, 2023.
URL: https://github.com/jazzband/django-auditlog (visited on 03/16/2023).

[42] Django.
ContentType. Django Project.
URL: https://docs.djangoproject.com/es/4.1/ref/contrib/contenttypes/ (visited on 03/17/2023).

[43] Django.
How to write a custom storage class. Django Project.
URL: https://docs.djangoproject.com/en/4.1/howto/custom-file-storage/ (visited on 04/10/2023).

[44] django-treebeard.
django-treebeard.
original-date: 2010-04-09T06:10:02Z.
Mar. 16, 2023.
URL: https://github.com/django-treebeard/django-treebeard (visited on 03/16/2023).

[45] CERTUNLP.

CERTUNLP/pyngen: Ngen REST. A python library for using Ngen.
URL: https://github.com/CERTUNLP/pyngen (visited on 02/20/2023).

[46] Celery.
Introduction to Celery — Celery 5.2.7 documentation.
URL: https://docs.celeryq.dev/en/stable/getting-started/introduction.html (visited on 02/28/2023).

[47] RabbitMQ.
Messaging that just works — RabbitMQ.
URL: https://www.rabbitmq.com/ (visited on 02/28/2023).

[48] Redis.
Redis. Redis.
URL: https://redis.io/ (visited on 02/28/2023).

[49] TheHive Project.
Cortex4py.
original-date: 2017-03-20T14:59:48Z.
Nov. 26, 2022.
URL: https://github.com/TheHive-Project/Cortex4py (visited on 02/20/2023).

[50] Grafana Labs.
Grafana: The open observability platform. Grafana Labs.
URL: https://grafana.com/ (visited on 03/06/2023).

[51] Kaplan Aaron.
"IntelMQ hands-on workshop".
TF-CSIRT/FIRST meeting Malaga, Jan. 31, 2020.
URL: https://www.first.org/resources/papers/malaga20/PUBLIC-Aaron-Kaplan-IntelMQ-malaga-20200131.pdf.

[52] Git.
Git.
URL: https://git-scm.com/ (visited on 04/05/2023).

[53] SensioLabs.
Symfony, High Performance PHP Framework for Web Development.
URL: https://symfony.com/ (visited on 02/24/2023).

[54] CERTUNLP.

Ngen PHP.
original-date: 2015-08-31T19:22:54Z.
Jan. 28, 2023.
URL: https://github.com/CERTUNLP/NgenBundle (visited on 04/04/2023).

[55] CERTUNLP.
Ngen Django Backend.
original-date: 2021-06-17T19:42:32Z.
Dec. 14, 2022.
URL: https://github.com/CERTUNLP/ngen (visited on 04/04/2023).

[56] Docker.
Docker: Accelerated, Containerized Application Development.
May 10, 2022.
URL: https://www.docker.com/ (visited on 04/05/2023).

[57] Docker.
Docker Compose overview. Docker Documentation.
Apr. 5, 2023.
URL: https://docs.docker.com/compose/ (visited on 04/05/2023).

[58] Docker.
certunlp's Profile | Docker Hub.
URL: https://hub.docker.com/u/certunlp (visited on 04/05/2023).

[59] Docker.
Docker Hub Container Image Library | App Containerization.
URL: https://hub.docker.com/ (visited on 04/05/2023).

[60] OEA.
Fondo de innovación en ciberseguridad.
Aug. 1, 2009.
URL: https://www.oas.org/es/sms/cicte/fondo-innovacion-ciberseguridad/ (visited on 02/24/2023).

[61] OEA.
Cisco y la Fundación Citi anuncian ganadores del Fondo de Innovación en Ciberseguridad. OEA - Organización de los Estados Americanos.
Aug. 1, 2009.
URL: https://www.oas.org/es/centro_noticias/comunicado_prensa.asp?sCodigo=C-041/21 (visited on 02/24/2023).

[62] OEA.
OEA - Organización de los Estados Americanos: Democracia para la paz, la seguridad y el desarrollo.
Aug. 1, 2009.
URL: https://www.oas.org/es/ (visited on 02/24/2023).