Tipos de CSIRT

Es posible implementar un CSIRT en un conjunto muy diverso de organizaciones, lo que implica que el concepto puede extenderse y adaptarse según las demandas de las organizaciones. Por ejemplo, en grandes instituciones o corporaciones, un CSIRT típicamente estará centrado en investigar incidentes de seguridad, dejando la tarea de implementación de políticas de seguridad y monitoreo de incidentes (InfoSec) a un equipo dedicado a tal efecto. Por otro lado, en el caso de pequeñas empresas u entidades con pocos recursos, los CSIRT también son responsables de las funciones de Infosec, sus procesos, herramientas y políticas.   
De esta manera, una posible clasificación de tipos de CSIRT según la organización que lo aloja, sería la siguiente:

SIEM dentro de un CSIRT

El término SIEM fue acuñado en 2005 por los analistas Amrit Williams y Mark Nicolett de la compañía estadounidense Gartner, una empresa especializada en investigación y consultoría de incidentes de seguridad, unificando los acrónimos en inglés SIM (security information management) y SEM (security events management) para describir metodologías muy similares pero ligeramente diferentes de ciberseguridad. Esta superposición de tareas hizo evidente que un nuevo término podría englobar ambos conjuntos de funciones, con el fin de disponer de un único acrónimo que pudiese identificar a una plataforma capaz de resolver los objetivos de los sistemas predecesores.   
 Los sistemas SIM tenían como objetivo el proceso de recolectar, monitorear y analizar datos relacionados a incidentes de seguridad de la información con datos provenientes de registros (logs) generados automáticamente por los sistemas de computación.  
 En cuanto a los sistemas SEM, tenían a su cargo el proceso de centralizar los registros provenientes de distintas fuentes, tales como sistemas, puntos finales, aplicaciones o servicios, para mejorar las detecciones de eventos y la administración de estos últimos mediante un proceso de respuesta a incidentes estandard o implementado bajo un protocolo secuencial.  
 De esta manera, el nuevo acrónimo SIEM significa Administración de Eventos de Seguridad de la Información, por sus siglas en inglés. Como plataforma que combina las funciones de los sistemas anteriormente descritos, sus capacidades comprenden la siguiente lista de tareas:

* Recolectar, analizar y presentar de manera eficiente datos relacionados a la seguridad.
* Análisis en tiempo real de eventos de seguridad
* Generar reportes y almacenar datos relacionados a la seguridad
* Administración de niveles y tipos de acceso e identidad.
* Auditoría de registros
* Respuesta a incidentes y operaciones de seguridad.

A medida que los sistemas de conexión se vuelven cada vez más complejos y deben soportar redes cada vez más heterogéneas con una creciente diversidad de dispositivos y servicios que a su vez incrementan las demandas de confiabilidad y disponibilidad de los activos de las organizaciones, también se multiplicaron las amenazas que intentan descubrir y explotar las posibles vulnerabilidades en todos los puntos de la infraestructura. Por esta razón, en simultáneo se está desarrollando un conjunto cada vez más numeroso y especializado de herramientas que intentan cubrir los permanentemente planos emergentes de análisis.   
 La situación descripta anteriormente se ejemplifica con los sistemas actualmente disponibles para monitoreo de puntos finales (HIDS), firewalls, monitoreo de activos de infraestructura de red como switches y routers, detección de malware en aplicaciones, detección y prevención de intrusiones en la red (NIDS y NIPS), alertas de pérdida de información, monitoreo de intentos de acceso a recursos y usuarios, vigilancia de archivos, nombres de usuario y hosts, etc.

En consecuencia, podemos observar que en un CSIRT existe una multitud de herramientas con un objetivo específico, que generan mensajes y alertas de distinto tipo de acuerdo a su misión puntual y es necesario un sistema capaz de procesar, filtrar, normalizar, presentar y almacenar toda esta información. Este es el objetivo del SIEM dentro del CSIRT y allí radica su importancia, ya que es necesario poder contar con un sistema que permita controlar a toda la orquesta de subsistemas que componen el área de monitoreo de un CSIRT.

## Estado del arte de los CSIRT y la ciberseguridad en el mundo

Existen diversos tipos de CSIRT funcionando actualmente alrededor del mundo, de una naturaleza pública o privada, con distinta área de cobertura ya sean puntuales (internos de una organización), regionales o nacionales y diferentes funciones principales, desde la búsqueda de vulnerabilidades, pasando por la investigación de amenazas, protección y monitoreo de infraestructura y activos críticos hasta la formulación de políticas públicas y educación ciudadana.   
 Los CSIRT privados generalmente son provistos por empresas especializadas, la mayoría provenientes de la industria de las telecomunicaciones que han derivado esfuerzos para cubrir un sector de mercado que es a su vez, un emergente de su sector industrial original. Es por esta razón, que las corporaciones han aprovechado su propio conocimiento de las infraestructuras que ellas proveen, para desarrollar soluciones a medida que se integran de forma nativa en sus productos ya existentes, tratando de esta manera de constituir un ecosistema de productos que se complementan entre sí y que en la mayoría de los casos representan soluciones propietarias y cerradas. Algunos casos son las corporaciones estadounidenses Cisco, IBM y AT&T, las empresas chinas Huawei y Alibaba, las compañías rusas R-Vision y BI.Zone, entre otras.  
 Por otro lado, los CSIRT estatales son aquellos centros de respuesta a incidentes que tuvieron como origen una necesidad de los gobiernos de controlar y proteger sus intereses en el ciberespacio, tales como el monitoreo de la actividad en sus infraestructuras críticas de red tales como puntos de interconexión a la red global, servidores DNS nacionales, sistemas autónomos y todos los activos estratégicos que están conectados a las anteriores, como centrales nucleares, hidroeléctricas, bases militares, puertos, sistemas de radares, universidades, comunicación interna entre los diversos actores del Estado, etc.

El crecimiento de la demanda en la cantidad y calidad de las conexiones, asociadas con la expansión de los servicios públicos prestados en línea, el comercio electrónico y la consolidación de las comunicaciones digitales en todas sus formas como canal omnipotente de vinculación social de las sociedades contemporáneas, implicó el traslado de todas las consecuencias de la actividad humana al mundo digital. Es por ello que los CSIRT gubernamentales cobraron una importancia capital como instrumento de los Estados para su presencia y acción en internet. Actividades como el comercio en línea y la banca digital han experimentado un crecimiento exponencial en la última década, trayendo aparejados el incremento de actividades ilícitas tales como estafas, fraudes, robo de identidades y cuentas bancarias. La irrupción de las redes sociales implicó el crecimiento de extorsiones, la organización de grupos criminales dispersos por áreas geográficas sin límite, el comercio de productos ilicitos, el terrorismo y una larga lista de acciones vinculadas a la naturaleza de las interacciones humanas. Los servicios digitales de procesamiento y almacenamiento de información en la nube se transformaron en la columna vertebral de las organizaciones que vieron volcada su actividad al mundo digital, generando en el proceso un sinnúmero de oportunidades para eventuales atacantes que buscan detectar y explotar las vulnerabilidades de los sistemas en los cuales descansa la infraestructura de estas organizaciones, con fines tan variados como obtener dinero a cambio de un rescate o sembrar el caos en intentos de activismo que derivan en ciberterrorismo.

El caso del hackeo a la infraestructura del sistema nacional de salud britanico con el ransomware Wannacry en mayo del 2017 es un ejemplo de un ciberataque que afectó a la atención y a la disponibilidad de los servicios de muchos hospitales y centros de salud, además de comprometer la información sensible de miles de pacientes. Otro ejemplo lo constituye el ataque de diciembre de 2015 a la red eléctrica de Ucrania por un grupo de hackers rusos que provocó un apagón masivo en todo el país durante varias horas, implicando pérdidas económicas cuantiosas y el desconcierto de la población. Por otro lado, el sabotaje de las centrifugadoras de uranio de Irán en el 2010 mediante el empleo del malware Stuxnet por parte de las agencias de inteligencias de Israel y Estados Unidos en combinación con una sofisticada operación de espionaje son ejemplos de distintos tipos de ataques a infraestructuras y activos críticos de un país, donde las amenazas pueden provenir de un espectro muy amplio de actores, desde simples individuos o grupos activistas movilizados por una causa, grupos organizados con fines de ciberterrorismo u otras formas de ciberdelito, hasta los intrincados y en ocasiones devastadores, ataques perpetrados por actores estatales de otro país.

Todo lo expuesto llevó al establecimiento de CSIRTs, centros de investigación, comandos especializados en ciberseguridad a cargo de fuerzas policiales y militares, incluso la creación de legislación específica sobre la materia e innovación de la ya vigente por parte de muchos países en lo que ya es considerado un asunto de la seguridad nacional. De esta manera, la Unión Europea dispuso la constitución, en el año 2004, de la Agencia para la Ciberseguridad de la Unión Europea (ENISA), dedicada a lograr un alto nivel de ciberseguridad común entre los países miembros de la Unión, contribuir a la definición de las políticas de ciberseguridad, mejorar la confiabilidad de los dispositivos IOT, inducir a servicios y procesos a la implementación de esquemas de certificación en buenas prácticas de ciberseguridad y cooperar con los Estados miembros. En 2016, con la aprobación de la directiva NIS, en su artículo 12, se creó la red de CSIRT de la Unión Europea. La agencia ENISA coordina esta red, compuesta por los centros nacionales y regionales de los países miembros, contando a treinta y nueve CSIRT en la red en 2020, aunque según la misma organización, en Europa hay 555 centros de respuesta a incidentes de ciberseguridad que pertenecen a fuerzas armadas, empresas privadas, corporaciones.

En los Estados Unidos de América, el US-CERT es un CSIRT responsable de coordinar la red del Sistema de Ciber Concientización Nacional (NCAS), el cual es el sistema nacional de ciberseguridad integrado, encargado de identificar, analizar y priorizar vulnerabilidades y amenazas emergentes. El objetivo del sistema es proveer a los ciudadanos de ese país de información sobre actualizaciones y recomendaciones de seguridad de manera gratuita para que estos puedan implementarlas en sus sistemas.   
En 2020, el US-CERT identifico y alertó sobre las amenazas respecto a las actividades de Corea del Norte (programa HIDDEN COBRA), vulnerabilidades sobre el bootloader del sistema GRUB2, actividad maliciosa proveniente de actores estatales de China, intentos de intrusión en dispositivos IoT relacionados a infraestructuras críticas por parte de Rusia, etc. El US-CERT depende del departamento de seguridad nacional (DHS); EEUU también dispone de la Agencia de Ciberseguridad y Seguridad de la Infraestructura (CISA), diversos centros militares y de investigación en universidades.

De manera similar a Estados Unidos y Europa, en muchos países del mundo existen CSIRTs gubernamentales y / o privados, tales como el Centro de Respuesta a Emergencias de Computación de China (CNCERT), el Centro Nacional de Ciberseguridad del Reino Unido (NSCS), el CERT-Bund de Alemania, el RU-CERT de la Federación Rusa, JPCERT de Japón, el Centro de Ciberseguridad Australiano (ACSC), el BruCERT de Bangladesh, KN-CERT de la República de Corea, el Centro de Coordinación de Respuesta a Incidentes de Seguridad de la Infraestructura de Internet de Indonesia (ID-SIRTII/CC), el MonCIRT de Mongolia, VNCERT de Vietnam, CERT.br de Brasil, CSIRT de Chile, colCERT de Colombia, CSIRTS.MX de Mexico, entre otros.

Es importante destacar que la mayoría de los CSIRTS / CERTS nacionales forman parte de redes y organizaciones internacionales con el objetivo de compartir información y coordinar acciones ante amenazas en el ciberespacio internacional o regional, tales como la mencionada ENISA, la red de CERTs de Asia Pacífico (APCERT) o la red global FIRST. En cuanto a los CSIRTS de America Latina en general y los de nuestro país en particular, serán objeto de un análisis detallado en las subsecciones siguientes.

-------------------------------------------------------------------------------------------

Además existen diversos institutos de investigación alrededor del mundo que se encargan de distintos aspectos de la ciberseguridad, desde por ejemplo la capacitación de profesionales, hasta la certificación de estándares para la seguridad informática. Algunas de las organizaciones más conocidas son SANS Institute, Global Information Assurance Certification, InfoSec Institute, Open Web Application Security Project, entre otras.

----------------------------------------------------- Esto reveer si sería útil o no

## Realidad de la Región y casos de uso

Con el advenimiento de la internet comercial en todo el mundo a inicios de la década de 1990, nuestra región comenzó a incorporar sus primeros centros de respuesta a emergencias de computación en 1997, cuando inició sus operaciones en Brasil el primer centro de América Latina: el “CERT.br”. Dos años después, se crearía en nuestro país el “arCERT”, siendo el primero de su tipo en Argentina y el segundo en la región. Chile (2001), Uruguay (2008), Paraguay (2012) son algunos de los casos de la región en los que se han implantado CSIRT nacionales, pudiéndose observar la disparidad temporal en la adopción de políticas y acciones en el ámbito de la ciberseguridad en América Latina. Esto se puede apreciar con más claridad si comparamos la situación respecto de las decisiones en otros países del mundo: teniendo en cuenta que el primer CERT fue creado en EEUU por un proyecto conjunto entre la agencia de investigación para la defensa DARPA y la universidad de Carnegie Mellon en 1988, la red FIRST a inicios de la década siguiente y el establecimiento de CSIRT en países como Japón en 1996, podemos considerar que las iniciativas de Brasil y Argentina fueron contemporáneas a la de otras potencias tecnológicamente avanzadas del mundo. Sin embargo, la adopción en el resto de América Latina se extiende gradual y paulatinamente a lo largo de las dos décadas siguientes.  
 El caso más paradigmático de nuestra región lo constituye Brasil: el país más grande, poblado e industrializado de sudamérica fue pionero en la introducción de políticas de seguridad informática y en reconocer en su momento el impacto que las redes de internet tendrían en el futuro. Actualmente dispone de 33 centros dedicados a la vigilancia y detección de amenazas en el ciberespacio, entre los que se encuentran, además del anteriormente mencionado “CSIRT.br”, el “CTIR.FAB” de la Fuerza Aérea Brasileña, “CSIRT USP” de la Universidad de San Pablo, el “CSIRT Petrobras” de la corporación petrolera homónima, el “CCTIR/EB” del Ejercito Brasileño, el “CSIRT.globo” que atiende la red de la compañía de televisión Globo TV y el “CTIR gov” encargado de la red del gobierno federal, entre otros.

A nivel regional existen varias organizaciones dedicadas a la promoción, formación, investigación e implementación de políticas y recursos de ciberseguridad, de las cuales LACNIC y la OEA ocupan un lugar destacado.

La Organización de los Estados Americanos (OEA) constituye el principal foro político de la región y fuente de la cual emergen muchas de las políticas que se implementan transversalmente en el continente, específicamente desde 2004 cuando en la IV conferencia de la CICTE (Comite Interamericano contra el Terrorismo, dependiente de la OEA) realizada el 29 de enero de ese año en Montevideo, Uruguay, se propuso por primera vez el marco para la creacion de una red interamericana de CSIRT de vigilancia y alerta. Este propuesta contemplaba el desarrollo de una red de puntos nacionales de contacto entre los CSIRT con responsabilidad nacional (“CSIRT Nacional”) de los Estados miembros de la OEA, que “funcionará las 24 horas del día, los siete días de la semana, capaz de responder apropiada y rápidamente a las crisis, incidentes y amenazas relacionadas a la ciberseguridad”. Se especificaba que la colaboración en esta red debería ser capaz de:

* realizar diagnosticos rapidos y precisos de un problema
* difundir rápidamente alertas de ataques reales sucediendo en la comunidad internacional;
* difundir rápidamente alertas de vulnerabilidades detectadas en la comunidad internacional;
* alertar a la comunidad mundial de actividades sospechosas y apoyar colaboraciones que investiguen y diagnostiquen esas actividades;
* proveer información para las estrategias de mitigacion y remediacion que combatan los ataques y amenazas; y
* minimizar la duplicación de los esfuerzos de análisis entre los equipos

El documento afirma que “la colaboración ayuda a equilibrar el conocimiento técnico que existe entre los equipos para limitar el daño y asegurar la operación continua de los servicios críticos". Para ello enuncia los principios que deben regir esta red panamericana de CSIRT:

* Indigena: El programa debe ser operado y controlado por entidades propias de cada nación participante, designadas por sus respectivos gobiernos.
* Sistémico: El sistema debe operar de forma multifacética, requiriendo una fuerza de trabajo entrenada y en alerta, compartir información regularmente en torno a las amenazas y vulnerabilidades actuales, re-evaluando e implementando constantemente las mejores prácticas y una apropiada interacción con los hacedores de las políticas públicas.
* En marcha: debido a la inherente evolución diaria de Internet, cualquier programa exitoso debe ser mantenido y actualizado regularmente. La seguridad de internet no se logrará con un arreglo de una única vez.
* Explicable: la “seguridad” en la “ciberseguridad”. Las reglas estrictas en aspectos tales como la manipulación de la información deben ser comprendidas y practicadas o los usuarios podrían dejar de confiar y los esfuerzos que hacen al sistema más seguro se verían disminuidos, lo que sería contraproducente.
* Construir sobre acuerdos existentes: hay un conjunto de entidades preexistentes en el hemisferio que proveen servicios de ciberseguridad en mayor o menor nivel. Cualquier sistema nuevo debería construirse sobre esas instituciones preexistentes para evitar la duplicación e incentivar la participación activa.

En cuanto a la identificación de las organizaciones existentes, el documento hace referencia a la necesidad de afiliación a algunos de los foros y redes reconocidas a nivel mundial de la época, tales como el foro global FIRST, la implementación de certificaciones como los del Centro de Coordinación CERT (CERT/CC) de la universidad Carnegie Mellon y las recomendaciones vertidas en libros como “State of the Practice of Computer Security Incident Response Teams”, “Creating a Computer Security Incident Response Team: A Process for Getting Started” y “Handbook for Computer Security Incident Response Teams (CSIRTs)” todos del mencionado CERT/CC. Adicionalmente se hace mención a un programa del departamento de defensa de Estados Unidos de certificación y acreditación de proveedores de servicios para la red de computadoras de defensa.

Finalmente, el informe presentado en la IV Reunión Regular del CICTE instaba a los países miembros a tomar un curso de acción que consistía en: conducir un censo para identificar los CSIRT existentes -junto a sus membresías y los servicios que proveen-, establecer un consenso mínimo de los servicios que todos los CSIRT miembros de la red deben ofrecer, redactar una resolución para enviarla a la Comisión de Seguridad Hemisférica (CSH, organismo dependiente de la OEA) y a la Asamblea General de las Naciones Unidas instando a los Estados miembros a crear sus CSIRT, implementar un consenso de servicios y estándares similares a las prácticas en Europa y Asia, así como establecer un sistema de asistencia técnica e intercambio de información para los CSIRT y asegurar el financiamiento para todo el programa.

Otro actor relevante para nuestra región es el Registro de Direcciones de Internet de América Latina y el Caribe (LACNIC), una organización internacional no gubernamental establecida en Uruguay en el año 2002, cuya función es “asignar y administrar los recursos de numeración de internet (IPv4 e IPv6), números autónomos y resolución inversa para la región” [N]. Además de administrar ciertos recursos críticos de la infraestructura de Internet de América Latina, esta organización promueve el desarrollo de internet en la región mediante políticas activas de cooperación entre los países y organizaciones miembros (unas 10000 aproximadamente) en 33 territorios de nuestro subcontinente. LACNIC declara entre sus objetivos apoyar y defender los intereses de la comunidad regional así como colaborar en la generación de las condiciones para que Internet sea un instrumento efectivo de inclusión social y desarrollo económico en América Latina y el Caribe.

[QUÉ ES LACNIC]

[PROYECTO AMPARO]

[LACNIC CSIRT]

[CONCLUSIONES DE LA PARTE LATINOAMERICANA]:

Hay pocos CSIRT en los paises de america latina

Si bien algunos países comenzaron temprano a crear CERTs y CSIRTs, la mayoría se sumó mucho más tarde. Esto habla de la conciencia sobre la ciberseguridad en la agenda de la región.

Si bien desde 2004 se empezó a hablar de la creación de una Red Inter-americana (o panamericana) de CSIRT (OEA), todavía no existe una red que integre a los CSIRT del continente

# Objetivo General

El objetivo de esta tesis es monitorear la red en puntos estratégicos o donde se tiene activos críticos para detectar amenazas o actividades sospechosas y de esta manera tener una respuesta proactiva en casos que se produzca algún incidente de ciberseguridad. El objetivo concreto es recolectar los datos obtenidos del sondeo de la red, almacenarlos y en caso de detección de algún tipo de amenaza informática alertar a los operadores de seguridad correspondiente. Todo esto que se mencionó constituye a un SIEM.

También, si previamente se programó, realizar algún tipo de acción para mitigar la amenaza generando un informe de lo sucedido.

Además otro objetivo propuesto fue la utilización de de herramientas de código abierto y gratuitas. Este proyecto forma parte de uno más grande que es la construcción de un CSIRT, que como se mencionó anteriormente es una pieza fundamental del equipo.

El objetivo de esta tesis es realizar un SIEM que funciona como parte de la organización de un CSIRT. El sistema SIEM debe obtener datos mediante el sondeo de la red en puntos estratégicos o donde se tiene activos críticos, para detectar amenazas o actividades sospechosas y de esta manera tener una respuesta proactiva en casos que se produzca un incidente de ciberseguridad. Dentro de los objetivos del SIEM está el almacenamiento de los datos, la correlación de los mismos, la presentación y búsqueda según a alguna o a un conjunto de clasificaciones en particular como por ejemplo la fuente que produjo el dato, la red, el activo, etc. Además en caso que ocurra un incidente alerta a los analistas de seguridad por distintos medios.

#### CSIRTS, CERT estatales y privados

Dentro de los CSRITs privados podemos encontrar los que funcionan dentro de organizaciones no estatales como son los casos las redes LINK y Banelco. Luego están las empresas particulares que ofrecen el servicio de CSIRT exterior a la misma, como el de caso de Platinum.

# Descripción de Requerimientos

Requerimientos funcionales:

* Envío de alertas sobre eventos relevantes mediante servicios específicos
* Se desea poder analizar en busca puntos críticos de la red con mayor detalle al que se posee actualmente.
* Implementación de un sistema parecido al de ticket para tener un registro de cómo se aborda una sospecha de ataque o un ataque en sí
* Automatización de respuestas a eventos de seguridad
* Correlación de eventos
* Visualizar los eventos en un tablero de mando y control
* Normalizar eventos
* Almacenamiento de eventos en una base de datos
* soporte para múltiples usuarios
* Colectar logs

Requerimientos no funcionales:

* Diagrama de topología de red
* La solución debe ser escalable respecto de la evolución de la red
* Manipulación de las bases de datos resultantes para su agregación bajo demanda
* Utilizar software gratuito y libre
* Que corra sobre un SO de código abierto
* Instalación sistemática y automatizada
* Inventariar y clasificar los activos de la organización

**Nuevos temas para el marco teórico**

Tipos de ataques

Cadena de mando

Componentes de un SIEM (colector de logs, estado de la red, etc) [cap 4]

Anatomía del SIEM [cap 5]

Respuesta a Incidentes [cap 6]

En base a la cadena de mando, la motivación de utilizar un SIEM

CIA:

Confidencialidad, Integridad y Disponibilidad componen a la triada de seguridad, el objetivo de un programa de seguridad es que se garantice la integridad, disponibilidad y confiabilidad de los activos de información de una organización.

**Componentes de un SIEM**

La implementación de un SIEM requiere la utilización de varios componentes, que individualmente cada uno de estos sirven para llevar a cabo una tarea básica. Un componente que se puede considerar fundamental para la implementación de un SIEM es el gestor de logs, con esta pieza se puede ir guardando una cantidad de eventos determinadas y en caso que la situación lo justifique se podría revisar manualmente los eventos. Esto último lleva a otra pieza necesaria para poder escalar el SIEM que es la correlación de eventos. Gracias a esto se pueden evitar varias búsquedas manuales. A continuación se profundizará sobre estos componentes.

Gestión de logs

Como se mencionó anteriormente este es un componente clave para el desarrollo del SIEM. Si no está recopilando al menos algunos de los eventos que produce la red, no se podrá extraer ninguna información de esos eventos y es probable que no se logre ninguna gestión de seguridad.

Al momento de implementar este componente se deberá definir los límites dentro de los cuales se restringirá el mismo. Los límites se pueden definir respondiendo algunas preguntas básica como:

1. ¿Cuánto tiempo se deben conservar los logs?
2. ¿Cuánta cantidad de logs se necesitará que se retenga?
3. ¿Qué tipo de logs del sistema se debe conservar (y eventualmente analizar)?

La primera pregunta plantea el tiempo de retención y destrucción de los datos. Este periodo puede depender de las regulaciones o leyes y de las políticas de la organización a la que sirva esta solución. Además pueden haber controladores legales que dictan cómo se debe destruir estos datos.

La segunda pregunta está orientada a definir cuál es la cantidad de datos que se necesita almacenar. Incluso una red pequeña puede generar una cantidad de eventos y logs que pueden llegar superar el almacenamiento disponible. Con esto en cuenta se deberá definir cual es la cantidad razonable de datos a almacenar en combinación con los requisitos de retención y destrucción de datos.

Por último está la tercera pregunta, que expone los múltiples formatos en los que se pueden obtener la información de eventos. Se debe elegir que tipo de información se conservará, ya que cada fuente de log (por ejemplo dispositivos de red como routers, switch o puntos finales como estaciones de trabajo) generalmente se puede configurar para proporcionar un log de prácticamente todo lo que está haciendo o solo de la información de estado más básica.

Para el tratamiento de eventos y logs existen varios tipo de estándares, métodos de obtención y formatos de los cuales los se pueden mencionar algunos de los más comúnmente utilizados

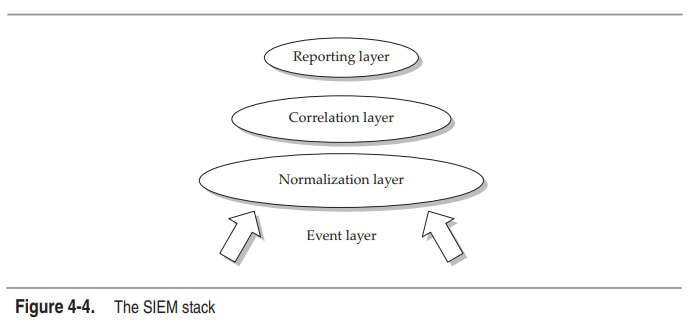
* Syslog: Es un estándar de la industria para que los dispositivos registren e informen eventos que realizan o situaciones que encuentran. La mayoría de los dispositivos de red, como enrutadores, conmutadores, firewalls y hosts, son capaces de producir un flujo de mensajes de syslog que se pueden enviar a una ubicación central para su procesamiento y / o almacenamiento. Estos mensajes cuentan con un encabezado que está definido en el RFC 5424 y el cuerpo del mensaje, que cada fabricante puede tiene la libertad de darle un formato específico.
* Alertas: Algunos tipos de dispositivos de red pueden usar métodos patentados para entregar información de eventos. Por lo general, estos pueden incluir tecnologías como antivirus y dispositivos del sistema de detección / prevención de intrusiones (IDS / IPS).
* Datos de flujo: Los datos de flujo son producidos por dispositivos de red y proporcionan información sobre flujos de datos específicos entre puntos finales. Se produce un solo mensaje de flujo que incluye información sobre los dos dispositivos (las direcciones IP del origen y destino), la cantidad de datos transmitidos y el servicio que utilizó la conexión (como por ejemplo HTTP a través del puerto 80). Los datos de flujo son un método muy útil para recopilar vistas de alto nivel del tráfico que transita por la red. La información de flujo se produce en formatos específicos de proveedores como NetFlow (Cisco Systems), J-Flow (Juniper) o QFlow (Q1Labs), así como el estándar sFlow (RFC 3176) compatible con múltiples fabricantes.
* Datos de evaluación de la vulnerabilidad: Los datos de herramientas de evaluación de la vulnerabilidad dan información de qué sistemas pueden ser vulnerables a qué tipo de vectores de ataque. La información de la herramienta EV es particularmente útil cuando se trata de verificar si un ataque detectado por un dispositivo, como un sensor IDS, tiene como objetivo un sistema que podría verse comprometido.

Es importante conocer previamente la topología básica de la red, que se quiere monitorear, para así determinar qué dispositivos son críticos y recopilar la información de logs necesaria de estos utilizando por ejemplo syslog. Además los dispositivos diseñados para detectar tráfico anormal, como los sistemas de detección de intrusiones, también son fuentes útiles para recopilar información.

Correlación de eventos

Como se vio anteriormente un paso fundamental para la creación de un SIEM es por lo menos contar con un gestor de logs. Revisando estos logs se podría determinar por ejemplo si el tráfico que no debería permitirse, según las políticas de la organización, ya ha atravesado el firewall. Si bien con lo mencionado anteriormente se puede detectar algunas amenazas, para obtener más valor de los eventos y la información que se ha recopilado, será necesario relacionarlos entre sí, correlacionar los datos obtenidos. Por ejemplo un mensaje de syslog de un firewall por si solo no podria significar nada, pero en combinación con otros eventos como mensajes de un router, un servidor de base de datos y un sensor de detección de intrusos, podría indicar un ataque contra un sistema vulnerable.

Con la gestión de log y la correlación mencionadas se puede descomponer al SIEM en capas y de esta manera tener una abstracción de alto nivel de los componentes básicos del mismo como ve a continuación:



La capa de eventos es donde se recopila logs y otros mensajes de eventos de los sistemas de la red. La capa de normalización es donde se convierten los mensajes relacionados que tienen un formato diferente a una sintaxis común. La capa de correlación es donde los eventos se relacionan entre sí para crear incidentes, y la capa de informes es donde se crea la salida y / o se toman las acciones basadas en el procesamiento de los eventos que se han ingresado en el sistema SIEM.

Los mensajes de eventos se pueden recopilar en formatos estándar como syslog y SNMP en la capa de eventos y luego almacenarse en un repositorio común. Sin embargo, dado que no existe un estándar para el contenido de los mensajes de eventos, se puede esperar que los productos de la misma clase (como firewalls o switch) pero de diferentes fabricantes produzcan mensajes diferentes para eventos similares. Esta diferencia para los diversos formatos de mensajes se aborda con la capa de normalización. Existen normalizadores para eventos ya conocidos como por ejemplo para logs de servidores apache, pero en caso de no encontrar el adecuado para algún tipo de aplicación o dispositivo se puede diseñar un normalizador propio.

En entornos heterogéneos se requiere la normalización antes de que sea posible la correlación. Una vez que los eventos están bajo un mismo formato se puede aplicar herramientas de correlación como scripts de Perl SEC que cuenta con un repositorio de reglas predefinidas para detectar varios tipos de ataque ya conocidos. Además se pueden programar script propios para correlacionar eventos particulares de la red a la que sirve el SIEM.

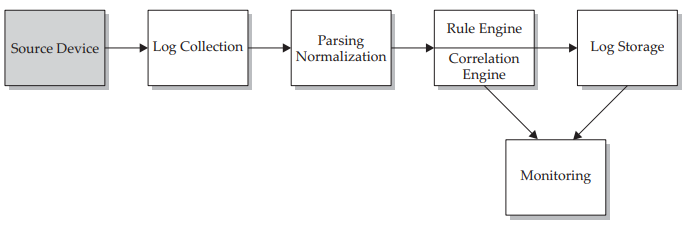
**Anatomía de un SIEM**

Un SIEM se puede comparar con una máquina compleja en el sentido de que un SIEM tiene varias partes, y cada una de las cuales realiza un trabajo específico, que deben funcionar correctamente juntas o, de lo contrario, todo el sistema fallará. Puede haber variaciones en el SIEM estándar, con partes específicas adicionales, pero un SIEM simple se puede dividir en seis partes o procesos separados. Estas piezas individuales son:

* el dispositivo de origen
* la recopilación de registros
* el análisis / normalización de los logs
* el motor de reglas
* el almacenamiento de logs
* la supervisión y recuperación de eventos.

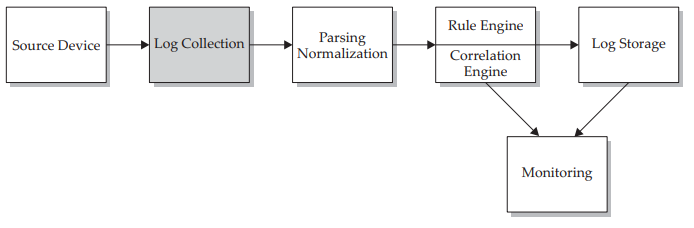
Cada una de estas partes puede funcionar independientemente de las demás, pero sin que todas funcionen juntas, el SIEM no funcionará correctamente. A continuación se explicará más en detalle estas etapas del SIEM

Dispositivo fuente



La primera parte de un SIEM es el dispositivo de origen que alimenta con información al SIEM. El dispositivo de origen puede ser un dispositivo de red, como un router, switch o algún tipo de servidor, pero también pueden ser logs de una aplicación o cualquier otro dato que se pueda adquirir. El dispositivo de origen no es una parte real del SIEM, cuando se considera el SIEM como una aplicación pero es una pieza vital del proceso SIEM general. Saber de qué fuentes desea recuperar logs en las primeras etapas del establecimiento de la arquitectura del SIEM se ahorrará una cantidad significativa de trabajo

Colección de logs



El siguiente paso en el flujo de logs de los dispositivos o aplicaciones, es obtener de alguna manera todos estos diferentes registros y enviarlos al SIEM. La mecánica real de cómo se recuperan los registros varía según la aplicación del SIEM y de las herramientas que se hayan seleccionado para trabajar dentro de este sistema, pero en la forma de los procesos de recolección de registros se pueden dividir en dos métodos fundamentales de recolección:

* El dispositivo de origen envía sus logs al SIEM, que se denomina método push.
* El SIEM solicita los registros cada un determinado tiempo y así obtiene logs de los dispositivos de origen, que se denomina método pull.

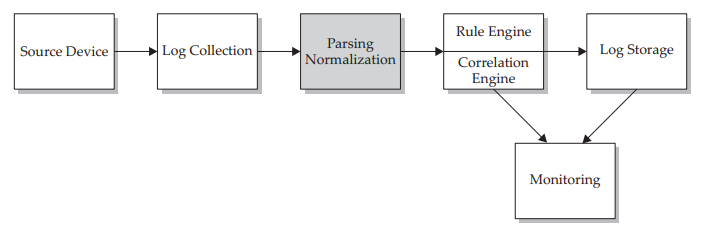
Cada uno de estos métodos tiene aspectos positivos y negativos cuando se usa en un entorno, pero ambos logran obtener los datos del dispositivo de origen en el SIEM.

El método push tiene la ventaja de que es fácil de instalar y configurar en el SIEM. Por lo general, solo necesita configurar un receptor y luego apuntar los dispositivos fuente al receptor. Un ejemplo común de esto sería syslog. Una desventaja es que estándares como syslog utilizan el protocolo UDP significa que nunca puede asegurarse de que los paquetes lleguen a destino, ya que UDP es un protocolo sin conexión. Otra desventaja es que si no se colocan los controles de acceso adecuados en el receptor del SIEM, un sistema mal configurado o un usuario malintencionado podría inundar el SIEM con información falsa.

El método pull tiene la ventaja de ser más seguro que el anterior ya que con este método se puede configurar credenciales de acceso para que el SIEM pueda conectarse al dispositivo origen y adquirir la información de los logs. Una desventaja del método pull es que se debe tener en cuenta al utilizar la recopilación de logs es que es posible que los registros no ingresen al SIEM en tiempo real. La colección de registros por el método pull se puede configurar para ejecutarse en determinados períodos de tiempo, lo que podría significar cada dos segundos o cada dos horas por ejemplo.

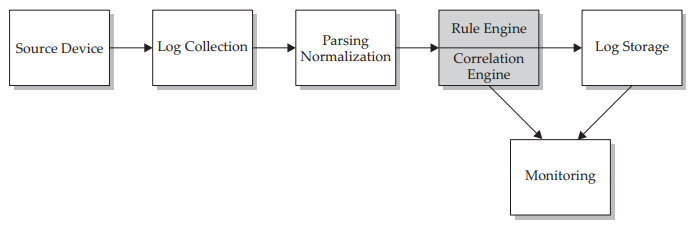
Además de los dos métodos mencionados anteriormente para colectar log existen otros diseñados por fabricantes, como por ejemplo para obtener registros en de servidor que corre una base de datos Oracle. También, si la situación lo amerita se pueden desarrollar métodos propios para la colección de log, si la aplicación o dispositivo origen son llevado a cabo por un particular.

Análisis / normalización de logs



Lo que debe suceder para que los logs sean útiles en el SIEM es re formatearlos en un formato estándar único que pueda utilizar el SIEM. El acto de cambiar todos estos diferentes tipos de registros en un solo formato se llama normalización. Cada tipo de SIEM manejará el acto de normalización de diferentes maneras, pero el resultado final es que todos los registros, sin importar el tipo de dispositivo o fabricante, tengan el mismo aspecto en el SIEM.

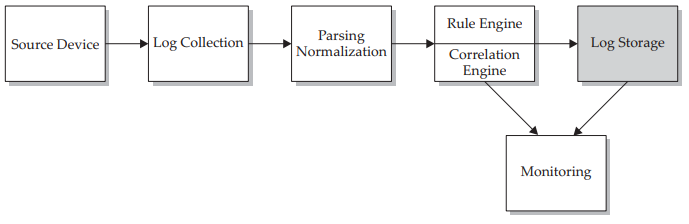
Motor de reglas / motor de correlación



El motor de reglas amplía la normalización de eventos de diferentes fuentes para activar alertas dentro del SIEM debido a condiciones específicas en estos registros. El método para escribir las reglas de correlación en el SIEM generalmente comienza de manera bastante simple, pero puede volverse complejo. Por lo general, las reglas se escriben utilizando una forma de lógica booleana para determinar si se cumplen condiciones específicas y examinar la coincidencia de patrones dentro de los campos de datos. Con un SIEM, en lugar de activar varias reglas para los diferentes tipos de eventos de una misma categoría (por ejemplo una alerta de inicio de sesión de administrador), se puede escribir una sola regla utilizando la lógica interna del SIEM para activar una regla basada en múltiples variables.

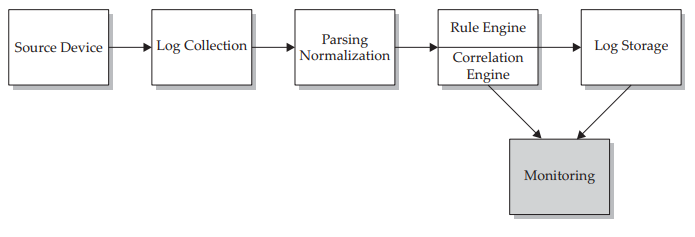
El motor de correlación es un subconjunto del motor de reglas. Lo que hace el motor de correlación es hacer coincidir múltiples eventos estándar de diferentes fuentes en un solo evento correlacionado. La correlación de eventos estándar en un evento correlacionado se realiza para simplificar los procedimientos de respuesta a incidentes para su entorno, mostrando un solo evento que se desencadena por múltiples eventos provenientes de varios dispositivos fuente. Por ejemplo para la detección de un posible compromiso de fuerza bruta, se necesitaría hacer coincidir lógicamente una cantidad de eventos de inicio de sesión fallidos, con la misma dirección IP de origen, con la misma dirección IP de destino y luego un inicio de sesión exitoso en el servidor de destino, desde la fuente origen durante un período de tiempo específico. Esto es lo que hace el motor de correlación para el SIEM: agrupa eventos individuales, que pueden formar parte de un posible incidente malicioso, en un solo evento que se muestra en la consola de un operador que monitorea el entorno. Por lo tanto, en lugar de tener que buscar en los registros tratando de encontrar eventos individuales y la relación entre esos eventos, se puede utilizar la lógica incorporada del SIEM para reunir los eventos de su red en un evento correlacionado.

Almacenamiento de logs



Para trabajar con los volúmenes de registros que ingresan al SIEM, se necesita una forma de almacenarlos con fines de retención y consultas históricas. El almacenamiento de registros en una base de datos es la forma en que la mayoría de los SIEM almacenan los registros. La aplicación de la base de datos debe estar optimizada para ejecutarse con el SIEM.

Monitoreo



La etapa final en la anatomía de un SIEM es el método de interactuar con los registros almacenados en el mismo. Una vez que se comience a obtener logs en el SIEM y los eventos se hayan procesado,se necesita una forma de hacer algo útil con toda la información, de lo contrario, los registros solo están en el SIEM para fines de almacenamiento. Un SIEM deberá tener una consola de interfaz, que estará basada en una aplicación web o en un software que corre junto con los demás en el servidor SIEM y se cargará en la estación de trabajo, para que los analistas puedan administrar las información y las configuraciones del sistema.

La interfaz en la aplicación SIEM real permitirá a los administradores de incidentes o ingenieros de sistemas una vista única de su entorno. Normalmente, para ver la información que recopila el SIEM, los administradores de incidentes o ingenieros tendrían que ir a los diferentes dispositivos y ver los logs en sus formatos nativos. En cambio con el SIEM se facilita la visualización y el análisis de todos estos registros diferentes porque el SIEM normaliza los datos. La consola será la forma principal de interactuar con los datos almacenados en el SIEM.

**Respuesta a incidentes utilizando un SIEM**

A medida que una organización y el departamento de tecnología de la información (IT por sus siglas en inglés) maduran a un nivel ya estable, una extensión natural y necesaria del programa de seguridad es un programa de respuesta a incidentes (IR por sus siglas en inglés). El programa de IR es un subconjunto del programa de seguridad general de la organización que se ocupa de las infracciones inesperadas a la política definida como "uso aceptable y esperado" de los sistemas de IT. El objetivo principal de un programa de IR es desarrollar un equipo, infraestructura y procedimientos para identificar brechas de seguridad rápidamente y luego ajustar el entorno de IT rápidamente para minimizar o detener las pérdidas de activos de IT, al tiempo que se minimiza el impacto en los objetivos principales de la organización (como mantener operativos los servicios elementales que ofrece la organización). Idealmente significa que el equipo de IR debe detener la brecha de seguridad o el ataque sin ralentizar o cerrar las funciones principales de la organización. Durante un incidente extremo, donde las pérdidas alcanzan proporciones críticas, la respuesta anterior puede considerarse y puede ser la correcta. Definir y gestionar esta respuesta equilibrada es donde entra en juego la necesidad de un programa de IR cuidadosamente planificado y documentado.

El programa de seguridad generalmente se define mediante políticas y procedimientos escritos que comunican las reglas de la organización y establecen el marco para implementar y administrar la postura de seguridad deseada por la organización. Cuando se está desarrollando un sistema de información para una determinada funcionalidad, incluye los componentes de seguridad básicos y obvios. Estos conceptos básicos incluyen un diagrama de la red, firewalls (al menos en las conexiones de borde con el mundo exterior), cuentas de usuario, listas de control de acceso (ACL por sus siglas en inglés) en recursos de red, protección antivirus, parches de seguridad para sistemas operativos y aplicaciones, etc.

En algunos casos, el diseño inicial de la red puede incluir dispositivos de seguridad como sistemas de detección y / o protección de intrusiones (IDS / IPS) y escáneres de vulnerabilidades. Para proteger la confidencialidad, integridad y disponibilidad de los activos de información más valiosos de la organización, uno de los primeros pasos es comprender qué activos son los más valiosos y garantizan el más alto nivel de protección.

Por lo general, se toman los siguientes pasos para desarrollar esta comprensión del valor de cada activo:

1. *Identificar todos los activos de información*: Los activos pueden ser de naturaleza muy diferentes. Puede ser desde un simple documento hecho con algún editor de texto, a un servidor, un switch troncal, un router de borde o la información que brindan los sensores de temperatura y humedad del aire de un datacenter. Se debe desarrollar un inventario de todos los activos que componen el sistema de IT
2. *Asignar un valor a cada activo*: El valor del activo no es solo lo que se pagó por él. El valor incluye, tener en cuenta los costos de entrega e instalación y el costo de reemplazo. Pero eso no es todo lo que se debe tener en cuenta al asignar un valor al activo. Se debe considerar el impacto en la organización si el activo se perdiera de una forma u otra.
3. *Categorizar cada activo para simplificar el diseño de la jerarquía protectora*: La valoración integral de todos los activos de información es un proyecto grande y puede resultar en un inventario de miles de estos, cada uno con un valor potencialmente diferente. Para simplificar la estructura de seguridad resultante, que se está diseñando para proteger estos activos en un nivel apropiado, se deben crear varias categorías (normalmente de tres a cinco categorías diferentes) que representan varios niveles de valor y protección requerida para los activos de esa clase.

De esta manera, a medida que la organización madura y se desarrolla, el programa de seguridad de IT tiene una base establecida en la que confiar.

TO-DO:

1. Requerimientos: Ver los que están al principio, enumerarlos y modificar un poco cada uno. Clasificarlos en funcionales y no funcionales
2. Completar el marco teórico:
   1. Amenazas y vulnerabilidades a las que responde el SIEM: terminar de agregar las amenazas y sus descripciones.

# Iteración 1: “Análisis de herramientas disponibles”

Uno de los SIEM más populares es Splunk [1], que está desarrollado por un socio de Cisco. Entre las soluciones propietarias también están Enterprise Security Manager [2] de McAfee. Se trata de soluciones ampliamente utilizadas en SOCs. Otra solución popular y de código abierto es ELK [3], que integra aplicaciones como Elasticsearch, Logstash y Kibana. AlienVault [4] es una solución de entrada de bajo costo con características sorprendentemente robustas para pequeñas y medianas empresas. No ofrece capacidades avanzadas, pero para organizaciones que buscan su primer producto SIEM, AlienVault es difícil de superar.

Por otro lado, en el mismo curso de CCNA Cybersecurity Operations 1.1 se me menciona a Security Onion como una posible solución SIEM, ya que es una suite de código abierto que contiene múltiples componentes para el monitoreo de seguridad de redes y herramientas para análisis de alertas.

Security Onion es una distribución de Linux de código abierto y gratuita para la búsqueda de amenazas, supervisión de la seguridad a nivel empresarial y la gestión de registros. Incluye herramientas como:

* Elasticsearch
* Logstash
* Kibana
* Snort
* Suricata
* Zeek
* Wazuh
* Sguil
* Squert
* CyberChef
* NetworkMiner

Similar a Security Onion se encontró el proyecto Sweet Security, que es una suite de herramientas para el monitoreo de redes de seguridad de redes. A diferencia de Security Onion no es una distribución de Linux, solo es un script de Python que se encarga de descargar y configurar las aplicaciones. Funciona en Ubuntu 16.04, Debian Jessie y Raspbian Jessie. Un inconveniente de este proyecto, es que no recibe soporte desde el año 2017.

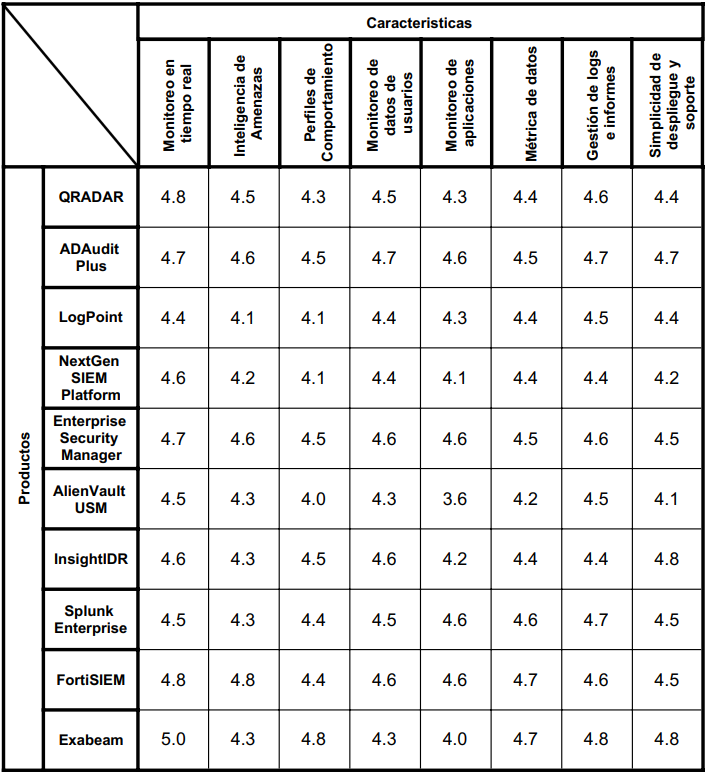
**QRADAR**

Empresa desarrolladora: IBM

Grafico de torta 1 Gráfico de torta 1 Gráfico de torta 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CARACTERÍSTICAS | QRADAR | ADAudit Plus | LogPoint | NextGen SIEM Platform | Enterprise Security Manager |
| Monitoreo en tiempo real | 4.8 | 4.7 | 4.4 | 4.6 | 4.7 |
| Inteligencia de Amenazas | 4.5 | 4.6 | 4.1 | 4.2 | 4.6 |
| Perfiles de Comportamiento | 4.3 | 4.5 | 4.1 | 4.1 | 4.5 |
| Monitoreo de datos de usuarios | 4.5 | 4.7 | 4.4 | 4.4 | 4.6 |
| Monitoreo de aplicaciones | 4.3 | 4.6 | 4.3 | 4.1 | 4.6 |
| Métrica de datos | 4.4 | 4.5 | 4.4 | 4.4 | 4.5 |
| Gestión de logs e informes | 4.6 | 4.7 | 4.5 | 4.4 | 4.6 |
| Simplicidad de despliegue y soporte | 4.4 | 4.7 | 4.4 | 4.2 | 4.5 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CARACTERÍSTICAS | AlienVault USM | InsightIDR | Splunk Enterprise | FortiSIEM | Exabeam |
| Monitoreo en tiempo real | 4.5 | 4.6 | 4.5 | 4.8 | 5.0 |
| Inteligencia de Amenazas | 4.3 | 4.3 | 4.3 | 4.8 | 4.3 |
| Perfiles de Comportamiento | 4.0 | 4.5 | 4.4 | 4.4 | 4.8 |
| Monitoreo de datos de usuarios | 4.3 | 4.6 | 4.5 | 4.6 | 4.3 |
| Monitoreo de aplicaciones | 3.6 | 4.2 | 4.6 | 4.6 | 4.0 |
| Métrica de datos | 4.2 | 4.4 | 4.6 | 4.7 | 4.7 |
| Gestión de logs e informes | 4.5 | 4.4 | 4.7 | 4.6 | 4.8 |
| Simplicidad de despliegue y soporte | 4.1 | 4.8 | 4.5 | 4.5 | 4.8 |



# Iteración 2: “Elección de las herramientas y arquitectura de red”

Como se describió en los requerimientos, la naturaleza de código abierto y libre del sistema base resulta una característica excluyente en la elección de la plataforma sobre la que desarrollamos la solución SIEM, así como los componentes integrantes de la suite elegida. Es por esta razón que elegimos Security Onion como plataforma base para el desarrollo del proyecto integrador, teniendo en cuenta sus características y herramientas durante la comparación con otras alternativas libres.

## Security Onion como sistema de gestión de eventos



Nuestra elección de Security Onion como plataforma estuvo basada en sus características destacables respecto de otras soluciones, como el soporte de una amplia y activa comunidad, el desarrollo continuo de mejoras, actualizaciones y correcciones, su capacidad polimórfica y funcional de actuar como IDS, plataforma SIEM o cluster de almacenamiento; esto deriva en la posibilidad de desarrollar distintas arquitecturas de una manera fácil y asistida para el despliegue y la consiguiente optimización de los recursos de hardware y de red.

Otras de las propiedades sobresalientes es la capacidad de integración directa con un conjunto casi universal de los sistemas IDS disponibles, tanto libres como privativos; esto se ve facilitado por un paquete de configuraciones iniciales predefinidas, su elevado nivel de automatización en el momento de instalar y poner a punto mediante scripts la infraestructura inicial del sistema; tales como el almacenamiento, normalización y gestión de logs (pila ELK), los sistemas IDS y los demás nodos de la arquitectura, junto a los sistemas de gestión de usuarios.

## Arquitectura del sistema de gestión de eventos

### Arquitectura de alto nivel

|  |
| --- |
|  |
| *Figura N + 1: Security Onion: Arquitectura de alto nivel* |

En la Figura N + 1 se observa la distribución y flujo de datos entre los componentes principales (la pila Elastic) y secundarios (Curator, ElastAlert, freqServer y domainStats) de Security Onion; junto a los sistemas de detección IDS como Bro, Snort, Suricata, Syslog, etc. Se distinguen las conexiones con los puntos de administración de los analistas del SIEM / CSIRT y los servicios web externos para el envío y recepción de alertas, notificaciones, análisis de tráfico, entre otros. Un punto a destacar es que la pila Elastic se encuentra desplegada en contenedores Docker.

### Arquitectura de despliegue

Security Onion sigue un modelo cliente - servidor y admite múltiples arquitecturas de despliegue, recomendando tres modelos generales: monolítico, densamente distribuido y distribuido; donde el servidor central es también denominado ”nodo master” y los clientes pueden ser llamados “nodos Forward” o “sensores” según se trate de nodos con una versión de Security Onion especialmente configurada para cumplir una función de procesamiento de frontera o de sistemas IDS, respectivamente. En primer lugar se han detallado los tipos de nodos para posteriormente describir las distintas arquitecturas que los implementan.

#### Tipo de Nodos

Los tipos de nodos que componen las posibles arquitecturas son:

* Nodo Master: Este nodo ejecuta su propia copia de la base de datos Elasticsearch, con la que gestiona las búsquedas a través del cluster y estructurar los otros nodos en el momento de su despliegue. Lo anterior implica que puede realizar las configuraciones necesarias para los nodos de los tipos “densos” y los de almacenamiento, pero no los de sensores o Forward, por carecer estos últimos de elementos de una pila Elastic. Este nodo permite a un analista conectarse mediante un enlace de supervisión para realizar consultas de los datos.
  + Este nodo contiene los siguientes componentes:
    - Elasticsearch
    - Logstash
    - Kibana
    - Curator
    - ElastAlert
    - Redis
    - Wazuh / OSSEC
    - Sguild

Elasticsearch, Kibana y Logstash son componentes de la pila Elastic, que trataremos en la siguiente sección junto a ElastAlert. El objetivo de Curator y Redis es administrar y optimizar las bases de datos de los nodos de almacenamiento; Wazuh es un IDS y Security Onion lo utiliza para el monitoreo de sí mismo, configurando un sistema HIDS ad hoc propio, aunque es posible desplegarlo en otros nodos o puntos de interés. Sguild permite consultar eventos de una base de datos MySQL desde dentro de Security Onion y muestra los resultados en una GUI. Además, actúa como intermediario de otros componentes secundarios como Squert, del que detallaremos sus funciones y comportamiento en una sección posterior.

* Nodos Forward: este nodo cumple la función de procesar el tráfico y reenviar los resultados al nodo master. Los logs generados por Snort / Suricata y Bro son enviados mediante syslog al Logstash del nodo master utilizando un túnel ssh, donde finalmente son guardados en la base de datos Elasticsearch del nodo master, pudiendo éste optar a su vez por volver a enviar los logs hacia los nodos de almacenamiento. Los logs pueden ser consultados a través de una búsqueda en el cluster.
  + Los componentes de un nodo Forward son:
    - Zeek (sucesor de Bro)
    - Snort / Suricata
    - Netsniff-ng
    - Wazuh / OSSEC
    - Syslog-ng

Zeek, Snort / Suricata y Netsniff-ng son procesadores de tráfico (IDS), donde Snort y Suricata serán tratados en una sección posterior. Syslog-ng es utilizado para recolectar logs de los IDS y enviarlos al Logstash del master, donde serán procesados y tratados antes de ser escritos en Elasticsearch.

* Nodos Pesados: Es un nodo híbrido entre el nodo Forward y el nodo Master, que incluye todos los componentes del nodo Forward, además de una instancia completa de la pila Elastic. Los nodos pesados envían los resultados de las consultas de su instancia local de Elasticsearch a las solicitudes realizadas por el Elasticsearch del nodo master mediante un túnel de autossh.
  + Los componentes del nodo master son:
    - Elasticsearch
    - Logstash
    - Curator
    - Zeek
    - Snort / Suricata
    - Netsniff-ng
    - Wazuh / OSSEC
    - Syslog-ng (envía los logs a la instancia local de Logstash)

* Nodos de almacenamiento: su objetivo es extender las capacidades de almacenamiento y procesamiento del nodo master. Estos nodos despliegan una instancia local de la pila Elastic; de manera análoga a los nodos pesados, cuando se realiza una consulta por parte de la instancia Elasticsearch del nodo master, esta es procesada por la instancia local de la pila Elastic del nodo de almacenamiento y devuelta por un túnel autossh.
  + Los componentes del nodo de almacenamiento son:
    - Elasticsearch
    - Logstash
    - Curator
    - Wazuh / OSSEC

#### Tipos de Arquitectura

La versatilidad de disponer de múltiples arquitecturas permite adaptar la plataforma a las necesidades de la organización en la que se implante. A continuación, se describen cada una de las opciones posibles:

* Arquitectura monolítica: Consiste en un único servidor que ejecuta simultáneamente los componentes centrales o propios de un nodo master y los de un nodo sensor en conjunto con los componentes de la pila Elastic; es un modo híbrido y concentrado que no se recomienda para enlaces de red de alto rendimiento por los elevados requerimientos de hardware necesarios.   
  Este tipo de arquitectura se recomienda para propósitos de pruebas en laboratorio y en entornos de baja demanda de tráfico de red.

|  |
| --- |
|  |
| *Figura N + 2: Arquitectura monolítica de Security Onion* |

* Arquitectura densamente distribuida: consiste en uno o más nodos pesados conectados a un nodo master. Solo se recomienda en el caso de que no sea posible desplegar una arquitectura distribuida, ya que tiene las mismas deficiencias de rendimiento de la arquitectura monolítica y no es apropiado para entornos de producción y/o enlaces de red de alta velocidad.

|  |
| --- |
|  |
| *Figura N+3: Arquitectura densamente distribuida de Security Onion* |

* Arquitectura Distribuida: consiste en un servidor master, uno o más nodos Forward y uno o más nodos de almacenamiento. Es el tipo de despliegue recomendado en términos de eficiencia de requerimientos de hardware, balance de la carga y almacenamiento de datos y optimización general de los recursos disponibles en la organización.

|  |
| --- |
|  |
| Figura N+4: arquitectura distribuida de Security Onion |

### Elastic, ElastAlert, TheHive y Cortex

Security Onion incluye la pila Elastic, cuyos componentes son la base de datos Elasticsearch, Logstash quien se encarga de recibir, procesar, normalizar y agregar los datos resultantes a la base de datos; estos datos son visibles mediante Kibana. El proceso comienza cuando Logstash recibe los datos sin procesar provenientes de múltiples fuentes, son normalizados por este componente y enviados a Elasticsearch para su almacenamiento. Kibana permite consultar la base de datos mediante una interfaz gráfica de usuario y utilizar esa información para propósitos de análisis de amenazas.

ElastAlert es un framework que permite identificar y alertar sobre eventos anómalos o patrones de interés sobre los datos de Elasticsearch. También provee múltiples mecanismos para enviar alertas mediante distintas plataformas externas, tales como Slack, correo electrónico, JIRA, Telegram y muchos más. Tanto ElastAlert como los componentes de la pila Elastic están desplegados sobre contenedores Docker.

Es destacable que, aunque Security Onion cubre gran parte de los requerimientos de un SIEM, no posee los elementos que permiten completar un sistema de manejo y respuesta a incidentes; por esta razón y luego de una investigación sobre las alternativas posibles, se incluyó a TheHive y Cortex como complemento de Security Onion. TheHive permite la gestión de incidentes de manera detallada y la colaboración con otros CSIRT mediante el uso compartido de información sobre incidentes en tiempo real; mientras que Cortex hace posible la automatización de las respuestas y operaciones ante incidentes utilizando los datos enviados por TheHive.

### Recibiendo, procesando y visualizando eventos: La pila Elastic

Como se pudo observar en la *Figura 2* del marco teórico, luego de recolectar los datos provenientes de múltiples fuentes, es necesario normalizarlos y agregarlos a la base de datos; estas tareas son llevadas a cabo por los componentes de la pila Elastic, en este caso Logstash y Elasticsearch, respectivamente.

Logstash es una tubería (en adelante “pipeline”) de procesamiento de datos gratuito y abierto del lado del servidor que ingesta datos de una multitud de fuentes, los transforma y luego los envía a su destino. Las fuentes de entrada admitidas por logstash son extremadamente amplias, como por ejemplo: syslog, STDIN, TCP, UDP, SNMP, IMAP, entre otras. Posteriormente, Logstash toma los datos sin estructura y los normaliza para crear conjunto ordenado mediante la identificación y conversión de la información a un formato común. Para realizar la tarea anterior, dispone de una gran variedad de filtros que facilitan el procesamiento general, independientemente de la fuente de datos. En este proyecto se utilizó a grok como filtro de las fuentes de información. Con los datos ya normalizados, es posible darles un formato específico para un destino en particular, ya que Logstash admite múltiples destinos para la etapa final del pipeline; desde una base de datos, archivos finales o servicios web. Security Onion, por defecto, almacena estos datos normalizados en un formato JSON en la misma pila Elastic, es decir la base de datos Elasticsearch.

Elasticsearch es una base de datos del tipo NoSQL distribuida y orientada al almacenamiento de documentos. Los datos normalizados provenientes de Logstash son documentos almacenados en índices en Elasticsearch. Cada índice está compuesto por uno o más shards (fragmento), por lo tanto un shard es un subconjunto de documentos, siendo el elemento básico de Elasticsearch y el que permite la escalabilidad del mismo. Un shard es también una instancia de un “índice de Lucene'', que indexa y almacena un documento en un segmento. Lucene es una librería desarrollada en Java para hacer búsquedas en una base de datos, constituyéndose en un motor de búsqueda que indexa y administra consultas en un conjunto de segmentos. La figura N+5 muestra la arquitectura de alto nivel del almacenamiento en Elasticsearch.

|  |
| --- |
|  |
| Figura N+6: Arquitectura de almacenamiento en Elasticsearch |

El índice y segmento correspondiente a un documento al momento de ser almacenado, corresponde a un identificador basado en el hash del contenido del propio documento a guardar. Elasticsearch dispone de una REST API con los métodos para administrar los documentos utilizando el identificador de estos. Otro aspecto a considerar en términos de rendimiento de Elasticsearch sobre el hardware de su host, es el tamaño de los shards. Estos se pueden definir en las configuraciones de Elasticsearch y su importancia radica en que un tamaño demasiado pequeño de los shards provocará un uso ineficiente del hardware, ya que estos deberán ser sometidos a un “merge” con una frecuencia mucho mayor a la habitual. Por el contrario, un tamaño demasiado grande de los shards demandarán tiempos demasiado largos de recuperación del cluster de Elasticsearch en caso de algún inconveniente.

Todos los datos almacenados en Elasticsearch pueden ser visualizados por Kibana, una interfaz gráfica perteneciente a la pila Elastic. Kibana permite visualizar los datos en gráficos circulares, de barras, histogramas, etc e interactuar con ellos; también es posible realizar análisis de ubicación cuando se disponen de los metadatos correspondientes mediante el complemento Elastic Maps, realizar análisis de series temporales de una manera rápida y sencilla, dispone de herramientas de inteligencia artificial, que mediante aprendizaje no supervisado permite detectar anomalías y patrones mediante las proyecciones sobre los datos. Otra de sus características es la posibilidad de realizar gráficos de correlación y entrecruzamiento, seleccionando campos de interés y filtros lógicos creados por el usuario. Es de destacar que para algunas de estas características es necesario la instalación de plugins complementarios y aunque en su inmensa mayoría son gratuitos, algunos pueden ser pagos ya que utilizan servicios web de la nube de los desarrolladores.

|  |
| --- |
|  |
| Figura N+7: Captura de pantalla de Kibana |

### Analizando y clasificando eventos: ElastAlert

A pesar de que Kibana permite consultar los datos almacenados en Elasticsearch y presentarlos de diversas maneras que resultan en una gran utilidad, carece de la capacidad de generar alertas cuando los datos coinciden con algun patron, especialmente cuando estos datos son escritos y consultados en tiempo real en la base de datos. Con este objetivo, la plataforma integra a ElastAlert, siendo un componente confiable, modular y simple de configurar. Su funcionamiento se basa en dos componentes principales: reglas y alertas; las primeras son utilizadas para comparar con los datos resultantes de las consultas que se hacen en forma constante a Elasticsearch, esta comparación consiste en hallar patrones o firmas definidas en las reglas dentro de los datos obtenidos de la consulta; si el resultado de la búsqueda es positivo, una alerta es disparada para notificar el evento.

Las alertas consisten en mensajes que permiten notificar a otro sistema con el objetivo de que este último realice una acción sobre las causas del evento que detectó la regla o bien informar a los analistas y/o responsables definidos. En cualquiera de los dos casos, las alertas pueden incluir toda la información recabada en un formato definido, tales como plantillas o cualquier arreglo configurado a tal fin.

Según la naturaleza de los eventos a clasificar, las reglas cuentan con un conjunto común de paradigmas de monitoreo, estos permiten identificar y generar alertas aprovechando las características de las anomalías al mismo tiempo que optimizan los recursos del resto del CSIRT en términos de hardware y atención de los analistas. Algunos de estos paradigmas se basan en el comportamiento, tales como la frecuencia que consiste en generar una alerta cuando se detectan N cantidad de eventos en un intervalo definido, el cambio de tasas de ocurrencia por arriba o abajo de un límite establecido como normal para un determinado tipo de eventos, cuando en los datos se encuentran presente campos que han sido previamente establecidos como parte de una lista blanca, negra u algún campo cuyo valor coincida con otros tipos de filtros, entre otros. Es posible definir y configurar tantas reglas como alertas sean necesarias.

### 

### El panel de control general: TheHive y Cortex

Como se mencionó en las secciones anteriores, Security Onion requiere de otros elementos capaces de realizar la gestión integral de incidentes y sus respuestas, elementos que sean capaces de condensar y presentar información a los analistas del CSIRT encargados de monitorear y responder a las anomalías e incidentes detectados. TheHive es la herramienta que se eligió para esta tarea ya que es una plataforma de respuesta a incidentes de seguridad gratuita y de código abierto, cumpliendo así con uno de los requerimientos no funcionales del proyecto, referido al tipo de licencia y accesibilidad al código. Otra de las razones para la elección de esta plataforma en particular ha sido su escalabilidad y su integración con MISP, lo que permite compartir información sobre las amenazas detectadas con otros CSIRT de organizaciones aliadas. Las tres capacidades centrales son la elaboración de casos, la respuesta a estos y la anteriormente mencionada colaboración con otros SOC / CSIRT.

|  |
| --- |
|  |
| Figura N+8: Alertas recibidas en el panel de TheHive |

En cuanto a la elaboración de casos y tareas asociadas, estas se crean en base a las alertas recibidas (figura N+8), donde el primer paso consiste en la creación de un caso para luego asociar este a una o varias de las alertas presentes utilizando la plantilla disponible (figura N+9), posteriormente es posible agregar tareas asociadas al caso, las cuales se pueden asignar a distintos analistas; a continuación es posible sumar métricas y campos personalizados, reducir el tiempo de búsqueda y recopilación de datos así como automatizar algunas tareas de recopilación de antecedentes en el manejo de incidentes mediante el uso del tablero (dashboard) dinámico, tal como se observa en la figura N+10. En el proceso de creación del caso, thehive permite agregar cualquier otra información que se considere relevante, como etiquetas, archivos sospechosos de contener malware, etc a modo de evidencias.

|  |
| --- |
|  |
| Figura N+9: Plantilla para la creación de nuevos casos |

|  |
| --- |
|  |
| Figura N+10: Dashboard dinámico de TheHive |

Luego de la creación de un caso, es posible sumarle todos los “observables” que sean necesarios, donde los observables son todos aquellos campos que se pueden agregar de forma manual y que constituyen fuentes de información para analizar cada caso. Una vez configurado un caso, estos son examinados por scripts llamados “analyzers” que correlacionan y filtran los datos del caso contra los provistos por otras instancias MIPS u otras fuentes de información como la propia base de datos local, servicios de resolución DNS, plataformas como Shodan, VirusTotal, Google Cloud Visión, entre muchas otras. Los observables también se pueden obtener por datos de las alertas recibidas, los cuales son previamente configurados en ElastAlert. Como se mencionó anteriormente ElastAlert realiza consultas a Elasticsearch y con los resultados busca patrones de interés para realizar una notificación, obtenida esta última extrae datos que se consideran de interés para ser enviados a The Hive. Un ejemplo de esto puede ser un número de IP, tipo de protocolo, fecha que se generó el log, puerto de origen y/o destino. La alerta que llega a The Hive contiene todos estos datos, considerados observables.

Luego de que el caso fue creado o sobre la misma alerta, el analista puede dar curso a una respuesta mediante “responders” que son scripts en los cuales se encuentra la respuesta del CSIRT a la amenaza. Tanto los responders como los analyzers se encuentran bajo la responsabilidad de Cortex, el subsistema encargado de procesar los casos de TheHive. Al final de esta sección, se presentan los diagramas de casos de uso correspondientes a TheHive y Cortex.

|  |
| --- |
|  |
| Figura N+11: Algunos de los analyzers disponibles en Cortex |

|  |
| --- |
|  |
| Figura N+12: Ejemplos de responders utilizables en Cortex |

|  |
| --- |
|  |
| Figura N+13: Casos de uso de gestión de usuario y configuraciones |

|  |
| --- |
|  |
| Figura N+14: Casos de uso de alertas y casos |

### Automatizando acciones: TheHiveHooks

Debido a que ciertos eventos tienen una elevada frecuencia de repetición y además su respuesta está perfectamente definida, sería de gran utilidad en estos tipos de ocasiones relevar a los analistas de la tarea que comprende tomar una alerta, elaborar un caso, ejecutar los analyzers y luego activar la respuesta correspondiente mediante los responders. Con este objetivo se implementó una automatización del proceso utilizando webhooks, que en el caso de TheHive se trata de una aplicación web (TheHiveHooks) que expone una REST API receptora de todos los cambios en TheHive y los envía hacia un punto final HTTP, donde un programa (en este caso Cortex) consume los datos recibidos. De esta manera, TheHive envía cada acción realizada en el (crear caso, actualizar caso, agregar tarea) vía los webhooks hacia el punto donde Cortex estará escuchando para actuar en consecuencia. Para esto fue necesario declarar la API mencionada anteriormente en el archivo de configuración *application.conf* de TheHive y luego desarrollar los *handlers* de los eventos para activar los responders apropiados. La importancia de automatizar respuestas a tipos de incidentes en tanto en cuanto un recurso escaso y valioso, sino en la capacidad de responder con eficacia y eficiencia a ciertos tipos de ataques potencialmente devastadores como un DDoS, donde miles de eventos (incidentes) se producen en simultáneo y la capacidad humana de responder a esta avalancha utilizando el protocolo descrito con anterioridad, es prácticamente nula. A continuación se presenta un diagrama simplificado de secuencia que involucra TheHiveHooks con TheHive y Cortex en la respuesta a un evento particular y la creación de un caso:

|  |
| --- |
|  |
| Figura N+17: Diagrama simplificado de secuencia del proceso de respuesta con TheHiveHooks y promover alertas a casos |

## Integración con los sistemas de detección

### En secciones anteriores se mencionó que Security Onion cuenta con componentes para realizar tanto monitoreo de red (NIDS) como monitoreo de puntos finales (HIDS). Durante la configuración inicial del sistema se pueden especificar los NIDS a utilizar, para una configuración rápida de los sensores.

### Suricata, Snort y Ossec

Suricata y Snort son motores de detección de amenazas en el tráfico de red. Ambos NIDS se basan en firmas o reglas para realizar la detección de amenazas, estas firmas son actualizadas constantemente conforme a la aparición de nuevos tipos de ataques, exploits y malware. Si bien estos NIDS son gratuitos y de código abierto Snort ofrece la versión paga, la cual cuenta con soporte para descargar las firmas actualizadas a la fecha. Por defecto Snort cuenta con las reglas básicas para la detección de amenazas bien conocidas.   
 Suricata, por otro lado, es desarrollado y mantenido por los colaboradores de la OISF, los cuales también dan soporte a las firmas ya que se actualizan las existentes y se agregan nuevas en forma permanente. Estas actualizaciones en las reglas son descargadas periódicamente mediante PulledPork, una utilidad que también es usada por Snort cuyo fin es descargar reglas y firmas desde distintos centros de investigación reconocidos en todo el mundo, como el SANS institute, Emerging Threats, entre otros.

A continuación se muestran las diferencias entre Snort y Suricata en una tabla comparativa:

|  |  |  |
| --- | --- | --- |
|  | Snort | Suricata |
| Desarrollador | CISCO | Open Information Security Foundation (OISF) |
| Lanzamiento | 1998 | 2009 |
| Lenguaje del código | C | C |
| Sistema operativo | Linux, Windows y Mac OS X | Linux, Windows y Mac OS X |
| Hilos | Monohilo | Soporte múltiples hilos |
| Soporte IPv6 | Si | Si |
| Reglas de Snort | Si | Si |
| Reglas de Emerging Threats | Si | Si |
| Formato de logs | unified2 | unified2 |
| Compatible con Aanval | Si | Si |

Por otro lado está Ossec (actualmente llamado Wazuh), que es un IDS orientado a hosts (HIDS). Al igual que los NIDS anteriores está basado en firmas para la detección de amenazas, además también es gratuito y de código abierto. Las reglas pueden descargarse del repositorio disponible en github.

## Arquitectura del despliegue

|  |
| --- |
|  |
| Figura N+18: Arquitectura de Despliegue |

En la imagen superior (Figura N+18) se muestra la arquitectura de despliegue del proyecto. La descripción, de izquierda a derecha, es: el proveedor ISP de conexión a internet y por consiguiente al exterior de la organización, el switch de capa 3 al que están conectadas las dependencias cuyos enlaces fueron seleccionados para ser monitoreados para este proyecto, los nodos Forward de Security Onion y un switch de la red interna del CSIRT. Se observa que los enlaces “Dependencia 1 - switch capa 3” y el de “switch capa 3 - nodo Forward de Security Onion Dependencia 1” tienen el mismo color; esto se debe a motivos de representar el hecho de que el switch capa 3 fue configurado para reenviar el tráfico entre el enlace de este y la dependencia 1 hacia el nodo Forward mencionado. Una situación análoga ocurre entre la Dependencia 2 y el nodo Security Onion Forward Dependencia 2.

El último eslabón de la conexión, el switch de capa 2, es el encargado de la red interna del CSIRT. A él se encuentran conectados las computadoras de los analistas y el nodo Master de Security Onion, los nodos Forward anteriormente mencionados y el servidor que aloja a TheHive y Cortex. Finalmente, los analistas pueden consultar y administrar los servidores correspondientes a los nodos Master y Forward de Security Onion así como al servidor que contiene a TheHive y Cortex.

# Iteración 3: “Configuración y despliegue en un ambiente de prueba”

En este proyecto se desarrolló sobre un ambiente de prueba primero y de producción después, sobre un servidor central y un sistema operativo de virtualización sobre el que se crearon un conjunto de máquinas virtuales, cada una alojando un servidor con nodos Forward, Master y el correspondiente a TheHive - Cortex. Se utilizó de guia los componentes, software y arquitectura de conexión entre ellos, mencionados en las iteraciones precedentes.

## Configuración del ambiente de prueba

El primer paso consistió en examinar los requisitos de hardware mínimos y recomendados por cada uno de los fabricantes de los sistemas y subsistemas elegidos, al mismo tiempo que se analizaron, por un lado, las demandas de tráfico de red en el ambiente de prueba y por el otro los requerimientos sobre los datos y capacidades que se esperan obtener del proyecto. Se procedió a realizar un diagrama topológico en la infraestructura objetivo, con esta información y los datos anteriormente mencionados, se procedió a realizar una estimación del hardware necesario para el servidor central que albergó las correspondientes máquinas virtuales de este proyecto.

Los requerimientos de hardware necesarios se incluyen a continuación discriminados según el tipo de nodo desplegado:

Nodo Master:

* Cantidad de CPU: 8 núcleos (vCPU), arquitectura X86-64 exclusivamente.
* RAM necesaria: a partir de 16 GB
* Almacenamiento: 1 Tb o lo necesario para cumplir la demanda de retención de logs.

Nodo Forward:

* Cantidad de CPU: 2 núcleos (vCPU) Para enlaces de 200 Mbps, por lo tanto para este proyecto fueron necesarios 10 vCPU (enlace de 1 Gbps). Arquitectura X86-64.
* RAM: desde 16 a 128 GB. En este proyecto utilizamos 32 GB.
* Disco: a partir de un tráfico de 50 Mbps se necesitan 540 GB debido principalmente a la acumulacion de pcaps. En este proyecto se utilizaron 200 GB ya que se contaba con restricciones de hardware disponible.

The Hive y Cortex:

* Cantidad de CPU: 8 núcleos (vCPU), arquitectura X86-64 exclusivamente.
* RAM: a partir de 8 GB
* Disco: a partir de 60 GB

Siguiendo el diagrama de la arquitectura de despliegue de la sección anterior, se optimizó al máximo el uso de los recursos del servidor disponible para permitir el despliegue de cuatro nodos: dos Forward y un Master de Security Onion, así como un cuarto conteniendo a TheHive y Cortex.

### Configuración del entorno de virtualización

Para el entorno de virtualización se utilizó VMWare, concretamente la suite vSphere HyperVisor v6.7.0 u3. Este sistema operativo basado en Unix permite gestionar los recursos de hardware disponibles, almacenar imágenes de distintos sistemas operativos y crear máquinas virtuales con estos últimos. Durante el proceso de creación de una máquina virtual, se selecciona el sistema operativo deseado y es posible asignar distintas cantidades de memoria principal, secundaria, cantidad de vCPU, número y tipo de enlaces de red, entre otros parámetros.

|  |
| --- |
|  |
| Figura N: Diagrama de una máquina virtual desde el punto de vista de un Hypervisor |

### Definición y configuración de las redes a observar

Se decidió monitorear dos dependencias en base a un análisis del ancho de banda de las dependencias existentes, por lo tanto se seleccionaron las que mayor volumen de tráfico registraban en función de un registro histórico y mediciones propias realizadas a lo largo de una semana. Las dependencias seleccionadas tenían un enlace con ancho de banda de 1 Gbps cada una, con velocidades promedio consideradas como la suma entre entrada y salida, entre 11,95 y 47,24 Mbps respectivamente; con picos poco frecuentes de 300 Mbps de tráfico, que no se contemplaron en los requisitos de hardware, por lo tanto habrá una posible pérdida de paquetes en estos casos. Se realizó un “port mirroring” de los puertos del switch capa 3 a los que están conectados estas dependencias y se los conecto con los respectivos enlaces de monitoreo de sendos nodos Forward de Security Onion.

Los gráficos a continuación muestran el volumen del tráfico medido en dos periodos de tiempo distintos: Durante un día (exceptuando las horas en las que la actividad era mínima) y a lo largo de una semana. Si bien estos registros que se presentan a continuación corresponden a una sola de las dependencias, la restante tenía un comportamiento análogo.

|  |
| --- |
|  |
| Figura N+1: Tráfico correspondiente a una dependencia, medido durante un día, obviando las horas donde este es casi nulo |

|  |
| --- |
|  |
| Figura N+2: Tráfico medido durante el periodo correspondiente a una semana |

## Configuración inicial del sistema base

Es posible instalar Security Onion en su versión 16.04 de dos maneras, sea mediante una ISO provista por los desarrolladores o bien mediante una serie de paquetes en una distribución Ubuntu. En este último caso será necesario contar con la distribución Ubuntu en su versión 16.04, ya que las distribuciones de Security Onion siguen a las distribuciones respectivas de Ubuntu; esto fue cierto hasta el año 2020 cuando se lanzaron nuevas versiones de Security Onion con soporte a otras distribuciones Linux: CentOS 7 y Ubuntu 18.04 y 20.04 aunque en el futuro se podrá desplegar en otros tipos de sistema Linux ya que desde la versión 2.x en adelante, el sistema se despliega en contenedores.

### Instalación y configuración de Security Onion

Como se mencionó en la sección anterior, existen dos maneras de instalar Security Onion: a partir de una imagen ISO o mediante paquetes / contenedores. Se eligió para este proyecto la segunda opción, el despliegue mediante paquetes de la distribución 16.04 de Security Onion ya que al momento del desarrollo de este trabajo integrador era la versión estable del sistema. Por consiguiente, se dispuso de un sistema operativo Ubuntu Server 16.04 con la particularidad de tener dos discos montados: el principal para el sistema operativo y el secundario para los datos recolectados en un directorio /nsm: índices en el caso de un servidor Master y capturas de paquetes o logs en el caso de un nodo Forward. Luego de finalizada la instalación de Security Onion, es necesario elegir el rol (Master o Forward) del nodo mediante el asistente y posteriormente realizar la configuración del mismo. Para esto último, se cuenta con la guia del asistente integrado que permite elegir y configurar las interfaces disponibles (observación o administración); en el caso de un nodo Forward, elegir el motor IDS (Snort o Suricata). El último paso consiste en elegir entre dos tipos de modo de funcionamiento: Producción o Evaluación.

|  |
| --- |
|  |
| Figura N+3: Asistente de instalación de Security Onion 16.04 |

|  |
| --- |
|  |
| Figura N+4: El asistente de instalación permite elegir el modo de despliegue |

Con el objetivo de cumplir uno de los requerimientos no funcionales del proyecto, que implica la automatización del despliegue (instalación y configuración) del sistema, se utilizó una herramienta de administración automatizada de servidores llamada Ansible en su versión 2.8.4 para la cual se desarrollaron scripts YAML conteniendo la secuencia de instalación de los paquetes, configuraciones, rol del nodo (Forward o Master) y librerías requeridas para el apropiado funcionamiento del sistema.

### Instalación y configuración de TheHive - Cortex

Para la instalación del gestor de incidentes, que tiene como componentes a TheHive y Cortex, se utilizó el sistema operativo Debian 10. En primer lugar se instaló TheHive, para ello fue necesario realizar la instalación previa de los componentes necesarios como las librerías de Java, Python y Elasticsearch; este último requirió una configuración en su archivo elasticsearch.yaml:

|  |
| --- |
|  |
| Figura N+5: Configuración añadida a elasticsearch.yaml para la instalación de TheHive |

Finalmente, los últimos pasos para la instalación de TheHive consisten en habilitar e iniciar el servicio de elasticsearch, agregar el repositorio que contiene los paquetes de TheHive, instalarlo y luego habilitar el servicio para poder iniciarlo.

En cuanto a Cortex, el proceso es similar al anteriormente descrito para TheHive, donde una vez descargados e instalados los paquetes de Cortex con su correspondiente secuencia de habilitación e inicio; se procedió a descargar del repositorio los responders y analyzers respectivos. Por último, se modifica el archivo de configuración de Cortex para indicar la ubicación del directorio que contiene los responders y analyzers mencionados anteriormente.

Posteriormente se actualizó la base de datos elasticsearch mediante la GUI web de Cortex, se creó un superusuario y luego las organizaciones donde se administrarán usuarios comunes y analyzers; es necesario crear un usuario con el rol de administrador de organizaciones. Las organizaciones tendrán habilitados y configurados determinados responders y analyzers según sea necesario.

El último paso del proceso consiste en comunicar TheHive y Cortex entre sí. Para ello se genera una API key en Cortex que será usada como parte de las modificaciones necesarias al archivo application.conf de TheHive. Las modificaciones completas que se realizaron al mencionado archivo se pueden apreciar en la siguiente imagen:

|  |
| --- |
|  |
| Figura N+6: Modificación al archivo application.conf de TheHive para la comunicación con Cortex |

## Automatización de acciones

El mecanismo elegido para automatizar las acciones fue mediante webhooks, para ello fue necesario implementar un entorno de virtualización en el mismo servidor donde se encuentran alojados TheHive y Cortex. Para ello se optó por utilizar el módulo de python “venv”, lo que requirió la instalación de Python 3.6 como primer paso. En segundo lugar se modificó el archivo application.conf de TheHive que se mencionó en la sección anterior para permitir la comunicación con el puerto del entorno de virtualización. En tercer lugar se verificó que en el nodo Master de Security Onion las reglas de ElastAlert tengan los campos necesarios configurados como observables ya que estos serán necesarios posteriormente dado que en TheHive están creados los observables que esperan esta información (source\_ip, destination\_ip, source\_port, destination\_port, alert, classification, category, etc).

Satisfechos los pasos anteriores, la instalación siguió los siguientes pasos:

1. Se creo una carpeta con el nombre webhooksenv
2. Un entorno virtual fue creado y activado en la mencionada carpeta
3. Se procedió a instalar las librerias necesarias: Flask, Gunicorn, Wheel, Request y Netaddr.
4. Se desactivo el entorno virtual y se habilitó en el firewall el puerto 5000
5. Se agregaron y modificaron valores al archivo de parámetros que utiliza webhooks.
6. Se inició y comprobó el estado del servicio.

Finalmente, se modificaron los archivos de configuración de TheHive y Cortex para actualizar la información necesaria referida a los webhooks.

# Iteración 4: “Reportes de incidentes y acciones automáticas”

En Security Onion y otros sistemas, el elemento descriptor que identifica y procesa a cada definición de incidente en particular es la regla. Las reglas comprenden una serie de campos que describen con precisión la naturaleza de un incidente dado y por lo tanto, existen tantas reglas como amenazas en circulación.

Cuando un nuevo malware es descubierto por el equipo de algún CSIRT con la capacidad de investigación suficiente o reportado a un laboratorio apropiado para este fin, es posible realizar un estudio de sus características y una vez identificadas estas últimas, proceder a crear una regla y agregarla al repositorio correspondiente para que otros CSIRT actualicen sus IDS con esta nueva definición y así contar con un filtro (la regla) que permita detectar este malware. Las reglas tienen un conjunto de campos donde se detallan características del paquete y su contexto, tales como el puerto de origen y destino, protocolo empleado, dirección IP, etc y unos campos dedicados a la naturaleza del incidente (clasificación, mensaje, prioridad, etc). Algunos de estos campos son comunes a todas las reglas y permiten agruparlas para administrar eficientemente las alertas generadas cuando una regla coincide con la descripción de un incidente. Dado que estos campos también se pueden considerar observables, es posible utilizarlos por TheHive y Cortex para automatizar respuestas.

## Análisis de prioridades de los incidentes

Como se indicó anteriormente, la estructura de las reglas consisten en dos partes bien definidas: un encabezado (header) que es obligatorio y un conjunto de campos opcionales. Dentro del header encontramos la acción (alerta, notificación, etc), el protocolo (tcp, udp), puertos de origen y destino, el sentido del evento (entrante o bidireccional) y las direcciones IP de origen y destino.

La segunda parte de las reglas incluye dos tipos de campos: los que describen la naturaleza del evento y aquellos que contienen información del paquete de datos. Dentro del primer grupo encontramos aquellos tales como msg (descripción del evento), sid (id de la firma), classtype (clasificación de reglas o alertas), priority (prioridad de la firma y/o alerta), target (especifica de qué lado está el objetivo, es decir puerto de origen y puerto de destino), entre otros. El segundo grupo contiene datos extraídos que provienen desde de la capa de red hasta la de aplicación de la pila OSI. Se pueden mencionar a los campos “GeoIP” (localización geográfica de la IP), “Fragbits” (presencia del bit de fragmentación), “ACK” (presencia del campo ACK en paquete TCP), “itype” (número del tipo de mensaje ICMP), “http.method” (tipo de método HTTP usado), entre otros.

|  |
| --- |
|  |
| Figura N+7: Estructura general de una regla |

Como se describió en los párrafos precedentes, dado que los campos están presentes en todas las reglas, es posible hacer uso de algunos de ellos para agrupar reglas que describen amenazas pertenecientes a un mismo grupo o categoría de malware, intentos de intrusión, reconocimiento, escalado de privilegios, etc y por lo tanto son útiles para gestionar los incidentes.

Es posible configurar esta gestión a través de un archivo que relaciona campos como categorías de eventos con prioridades de la alerta generada. Este archivo llamado “classification.config” se encuentra bajo el directorio que almacena las reglas descargadas desde diversas fuentes; en particular relaciona los campos “classtype” con “priority”, de manera tal que cualquier regla cuyo campo classtype contenga a los descritos en este archivo, generará una alerta con prioridad definida también en este. De esta manera, es posible administrar un enorme número de reglas agrupadas en un reducido grupo de categorías y modificar el nivel de prioridad que tendrá en el sistema las alertas que generan.

El objetivo de asignar distintos niveles de prioridad a las alertas generadas por los eventos que sucedan radica en la naturaleza de los eventos, su importancia y la gestión de la atención de los analistas del CSIRT. Esto se debe a las necesidades de optimizar el uso de los recursos técnicos y humanos del centro de respuesta a incidentes para cumplir de la manera más eficiente posible con los objetivos y políticas de la organización a la cual pertenece. De esta manera, la naturaleza de los incidentes determina su elegibilidad para una respuesta automatizada al tener en cuenta por un lado su estructura bien conocida y por el otro su alta tasa de repetición en un periodo determinado. En estos casos, sería inutil destinar valiosos recursos como la atención de un analista ya que se conoce perfectamente la estructura del incidente y por lo tanto la respuesta apropiada o en aquellos casos en los que aún conocida su estructura, el incidente proviene en simultáneo de múltiples fuentes en muy poco tiempo, de manera que la capacidad humana de responder de a uno a la vez estaria tan sobrepasada que no sería efectiva. Estos son los casos de ataques de reconocimiento y los de denegación distribuida de servicio, entre otros.

De aproximadamente cuarenta y siete (47) categorías de incidentes disponibles por defecto, consideramos para el máximo nivel de prioridad a siete clasificaciones dado su nivel de ocurrencia y nivel de impacto para la organización.

* Web-application-attack: esta categoría engloba a un conjunto enorme de malware y ataques a nivel de capa de aplicación. Gusanos, ransomware, ataques de reconocimiento entre otras amenazas comparten esta categoría. Sobre el caso particular de los ataques de reconocimiento, se aplicaron filtros para separarlos de los demás ya mencionados.
* Unsuccessful User: intentos repetidos de ganar acceso en ciertos activos e infraestructura de la organización.
* Attempted-dos: intentos de ataque de denegación de servicio y su variante distribuida
* Known client side exploit attempt: intento de ejecución de exploits en el lado del cliente.
* Exploit Kit Activity Detected: detección de actividad de un kit de exploits
* A suspicious filename was detected: detección de nombres de archivos sospechosos
* Network Trojan: detección de un virus troyano de red.

En base a las alertas que provienen de los IDS y que son consultadas a Elasticsearch por ElastAlert, este último notifica a TheHive para dar lugar al proceso de gestión de incidentes, su eventual análisis y correlación histórica para finalmente emitir una notificación a los destinatarios finales utilizando los métodos definidos en los responders que Cortex dispone mediante scripts desarrollados en python.

## Automatización de notificaciones

La automatización de acciones puede dividirse en dos categorías: acciones activas ya sea sobre el incidente, las causas que lo provocan (vulnerabilidades, puertos, direcciones del atacante, etc) o bien mediante notificaciones de la detección a los analistas del CSIRT y/o los responsables designados (analistas de un NOC, administradores de redes, encargados o usuarios de los activos afectados, entre otros). Se eligió esta última categoría por considerar que era la más apropiada para notificar y generar el mayor impacto al cubrir gran parte de los medios de comunicación disponibles en la organización. Cuando un incidente es reportado por ElastAlert a TheHive tal como se describió al final de la sección anterior, se inicia una serie de pasos en la cual intervienen webhooks y Cortex como se puede apreciar en la figura siguiente:

|  |
| --- |
|  |
| Figura N+8: Secuencia de respuesta automatizada |