**Ingeniería en computación**



*Proyecto Integrador*

**Implementación de una solución para gestión de eventos de seguridad de una red de datos**

*Autores*

Sergio David Figueroa y Federico Nicolás Sepúlveda

*Director*

Mgter. Ing. Miguel Ángel Solinas

# 

# Resumen

# Índice

[**Resumen**](#_kol2h6qgysv9) **2**

[**Índice**](#_klumoiel3imn) **2**

[**Índice de figuras**](#_iwevs3auf9l6) **3**

[**Índice de tablas**](#_7129zk3d87tq) **4**

[**Agradecimientos**](#_kbys14dyjqjf) **4**

[**Glosario**](#_clves114mmg4) **4**

[**Introducción**](#_msgxh4ei5l0s) **7**

[**Objetivo General**](#_78o1ww6a21kk) **7**

[Motivación](#_6in6ygfnp5ic) **8**

[**Marco Teórico**](#_d9ffu4qmfjif) **9**

[Personal](#_ixls8dg7djkd) 10

[Procesos](#_uav59ix7td1f) 13

[Tecnología](#_uxifivrhnbyi) 13

[Agregando contexto a los incidentes](#_mutedczcpitr) 14

[Definición de conductas normales](#_v38go8e8oip2) 15

[Inteligencia de amenazas](#_zf482rispops) 15

[Obstáculos para el manejo eficiente de incidentes del SOC / CSIRT](#_549bommxiogm) 16

[Multiplicidad de herramientas de monitoreo y su integración](#_ykrisz53yqps) 16

[SIEM: Definición y sus funciones](#_fcw24esumwr1) 17

[Análisis de herramientas disponibles](#_l27jioqn7rm) 20

[Soluciones comerciales](#_qfy2fmcg4w6m) 22

[Soluciones gratuitas y de código abierto](#_6o7syte386lv) 26

[Ámbitos de actuación de los CSIRT](#_vv7uqf175hj2) 27

[**Descripción de Requerimientos**](#_6rai3ouvzmmy) **30**

[Requerimientos funcionales](#_y1qkb8o6l61l) 30

[Requerimientos no funcionales](#_m534rw324oov) 32

[**Iteración 1: “Elección de las herramientas y arquitectura de red”**](#_8wfx5kidqzr9) **34**

[Security Onion como sistema de gestión de eventos](#_6xxk90feqwsh) 34

[Arquitectura del sistema de gestión de eventos](#_oujxtgyjrw7g) 35

[Arquitectura de alto nivel](#_5tjrs2d54xrk) 35

[Arquitectura de despliegue](#_u3jf3sp8ta81) 36

[Tipo de Nodos](#_kn9i2cd2rftp) 36

[Tipos de Arquitectura](#_86kyhzv16i7a) 38

[Elastic, ElastAlert, TheHive y Cortex](#_eg8ed8fgbvvq) 40

[Recibiendo, procesando y visualizando eventos: La pila Elastic](#_g14uofnzjar) 41

[Analizando y clasificando eventos: ElastAlert](#_pqzcyedw8mey) 44

[El panel de control general: TheHive y Cortex](#_lq03o39atr0w) 45

[Automatizando acciones: TheHiveHooks](#_y03qr0todr02) 54

[Integración con los sistemas de detección](#_bto1am9p5t5z) 56

[Suricata, Snort y Ossec](#_5el7xg6aq71u) 56

[Arquitectura del despliegue](#_juvwd52ve8ho) 58

[**Iteración 2: “Configuración y despliegue en un ambiente de prueba”**](#_xdnrscy4e329) **58**

[Configuración del ambiente de prueba](#_5b96qatzvv33) 59

[Configuración del entorno de virtualización](#_vbv9ozuusue0) 60

[Definición y configuración de las redes a observar](#_4gy5pwndbx0t) 60

[Configuración inicial del sistema base](#_pr1m3xgke3pt) 62

[Instalación y configuración de Security Onion](#_dhdn30qeke8m) 62

[Instalación y configuración de TheHive - Cortex](#_5m89g5riahoi) 64

[Automatización de acciones](#_3p0estdaov1m) 66

[**Iteración 3: “Reportes de incidentes y acciones automáticas”**](#_a0fkq9pxl854) **67**

[Análisis de prioridades de los incidentes](#_csgr7xq0owzd) 67

[Automatización de acciones](#_d4bzhip9yjkb) 69

[**Iteración 4: “Testing y corrección para puesta a prueba”**](#_eojcxztr0qz5) **71**

[Test de detección de ataques varios](#_sozhocrqz06a) 71

[Test de reportes de incidentes](#_143fjw3fk3os) 71

[Test de acciones automáticas](#_qu0xa8uwm63h) 71

[**Conclusión**](#_e2lxx5wud3th) **71**

[**Futuros trabajos**](#_eomtxgzcuip2) **71**

[**Bibliografía**](#_jok3ar6e8r1a) **71**

[**Anexos**](#_mm3s3xi5bc29) **71**

# Índice de figuras

[Figura 2](#_i1jwb047kfjj)

# Índice de tablas

# Agradecimientos

# Glosario

CCNA: acrónimo en inglés de Cisco Certified Network Associate, un certificado de validación profesional emitido por la corporación Cisco para técnicos que operan sus productos.

CERT: siglas en inglés de Computer Emergency Response Team, en español equipo de respuesta a incidentes de computadoras. Término registrado comercialmente por la universidad estadounidense de Carnegie Mellon.

CICTE: siglas del Comite Interamericano contra el Terrorismo, entidad dependiente de la OEA.

CPU: siglas en inglés de Central Processing Unit, en español Unidad de Procesamiento Central.

Creative Commons:

CSIRT: siglas en inglés de Computer Security Incident Response Team, en español equipo de respuesta a incidentes de seguridad de computación. Es el equipo de profesionales, sistemas y toda la infraestructura (hardware y software) de detección y respuesta a incidentes de ciberseguridad de una organización.

DDoS: siglas en inglés de Denied Distribution Of Service, en español denegación distribuida de servicio.

DNS: siglas en inglés de Domain Name Service, en español Servicio de Nombres de Dominio, es un protocolo de red de la capa de aplicación.

Dirección MAC: siglas en inglés de Media Access Control, en español control de acceso a medios, es un conjunto de bytes que constituyen la dirección física (única) que identifica a un dispositivo conectado a una red.

EMCFFAA: Estado Mayor Conjunto de las Fuerzas Armadas Argentinas.

Firewall:

GNU: acrónimo recursivo en inglés de “GNU is Not Unix”, en español GNU No es Unix.

GPL: siglas en inglés de General Public Licence, en español licencia pública general, es un tipo de licencia GNU.

GUI: siglas en inglés de Graphical User Interface, en español Interfaz gráfica de Usuario.

HIDS: siglas en inglés de Host Intrusion Detection System, en español sistema de detección de intrusiones en un host o punto final.

IDS: siglas en inglés de Intrusion Detection System, en español sistema de detección de intrusiones. Es un componente de software destinado al procesamiento de firmas basadas en la información recolectada del tráfico de red, mediante una sonda colocada en un enlace de la infraestructura de comunicaciones de datos. Los IDS son un componente vital de un CSIRT debido a que realizan la identificación a priori de eventos en el tráfico de datos y su clasificación como un incidente.

InfoSec:

IPV4 e IVP6: siglas en inglés de los protocolos de Internet versiones 4 y 6, respectivamente.

IPS: siglas en inglés de Intrusion Protection System, en español sistema de protección de intrusiones.

IT-ISAC:

LACNIC: siglas del Registro de Direcciones de Internet de América Latina y el Caribe, organización no gubernamental establecida en Uruguay.

Licencia GFDL:

Licencia AGPL:

Licencia APACHE:

Licencia BSD: siglas en inglés de Berkeley Software Distribution, licencia de software libre desarrollada en dicha universidad homónima de Estados Unidos.

Linux: núcleo de código (kernel) abierto de familias del mismo nombre de sistemas operativos de software libre.

Lockheed Martin:

Log: equivalente en inglés a “registro” en español. Término utilizado específicamente para registros de datos con un formato definido.

Malware:

MIT: siglas en inglés de Massachusetts Institute of Technology, universidad de los Estados Unidos cuyo nombre es usado para un tipo de licencia de código libre desarrollada en esa universidad.

MSSP: siglas en inglés de managed security service provider, en español proveedores de servicios de seguridad gestionados. Empresas que prestan servicios de seguridad informática a organizaciones.

NIDS: siglas en inglés de Network Intrusion Detection System, en español sistema de detección de intrusiones a nivel de red.

NIPS: siglas en inglés de Network Intrusion Protection System, en español sistema de protección de intrusiones a nivel de red.

NSM: siglas en inglés de Network Security Monitoring, en español monitoreo de seguridad de redes.

OEA: Organización de Estados Americanos.

RAM: siglas en inglés de Random Access Memory, en español memoria de acceso aleatorio.

Ransomware:

SaaS: siglas en inglés de Software as a Service, en español software como servicio, es un modelo de negocio de software a demanda.

SEM:

SIM:

SIEM:

SOC: siglas en inglés de Security Operations Center, en español centro de operaciones de seguridad de red.

SYN: bit usado en el protocolo TCP para indicar la sincronización del número de secuencia al comienzo de una comunicación utilizando el protocolo antes mencionado.

TCP: siglas en inglés de Transmission Control Protocol, en español protocolo de control de transmisión. Uno de los protocolos fundamentales en la comunicación de datos.

VPN: siglas en inglés de Virtual Private Network, en español red privada local

# Introducción

Este proyecto consiste en la implementación de un Sistema de Administración de Eventos y Seguridad de la Información (SIEM, por sus siglas en inglés) para la Universidad Nacional de Córdoba. Un SIEM consiste en varias herramientas como bases de datos, filtros para normalizar la información, tablero para visualizar los datos y generador de alertas entre otras. Por otro lado tenemos el monitoreo de la red en tiempo real que utiliza un Sistema de Detección de Intrusiones (IDS, por sus siglas en inglés). Este último envía los datos ya procesados al SIEM para que los almacene en la base de datos.

Además como se pretende que el SIEM funcione dentro de un Equipo de Respuesta a Incidentes de Seguridad Informática (CSIRT, por sus siglas en inglés) se necesita integrar un gestor de incidentes. Este último sirve para tener un registro de los incidentes ocurridos, permite administrar las tareas del equipo de analistas, compartir y solicitar información con otros CSIRT entre otras funciones.

# Objetivo General

El objetivo de esta tesis es el desarrollo e implantación de un sistema SIEM dentro del proyecto general de la creación del CSIRT de la Universidad Nacional de Córdoba, con el fin de otorgar al mencionado centro de respuesta, el instrumento capaz de obtener, analizar y presentar datos sobre las amenazas detectadas por los demás subsistemas del CSIRT.

# Motivación

La tecnología y la digitalización de la información convierten a los datos en un activo muy importante de las organizaciones y de los individuos en general. Es fundamental saber cómo proteger los datos para evitar ser víctima de un ciberdelito o parte involuntaria de una ciber operación a gran escala. A pesar de que actualmente las técnicas de seguridad hacia los datos y la infraestructura de redes están en auge, las herramientas de seguridad como firewalls, IDS y otras que permiten prevenir ataques informáticos no son suficientes para mitigar y tener un seguimiento de actividades maliciosas o potencialmente maliciosas para lograr fortalecer la infraestructura y prevenir futuros incidentes. Resulta necesario contar con un sistema global que permita integrar un variado conjunto de utilidades que brindan soluciones puntuales y específicas, para crear una defensa inteligente y eficiente de los activos de información de una organización.

Actualmente la infraestructura de red y los sistemas asociados conviven en un ambiente de *saturación de la información* que implica un alto costo de procesamiento y ponen a prueba permanentemente a los sistemas encargados de la optimización de los recursos de hardware y software con los que cuenta la infraestructura, tales como uso de CPU y memorias RAM de routers, switches y servidores, el almacenamiento secundario donde el desafío de retener un ingente volumen de datos generados por el exponencial y siempre creciente tráfico de la red amenaza constantemente con el colapso de los medios disponibles sin importar su capacidad de almacenamiento, entre otros problemas, configuran una avalancha constante de información que sería imposible de analizar siquiera una parte de ella en un momento determinado utilizando métodos que impliquen el procesamiento en bruto. En esta situación, sería imposible distinguir un evento puntual y nocivo dentro de esta cantidad gigantesca de información que se genera permanentemente en la red, de un evento normal o de tráfico legítimo y en caso de identificar un potencial incidente, este tendría unas probabilidades muy altas de ser un falso positivo. Esto último es característico de los sistemas basados en el análisis de firmas, como los IDS, IPS o antivirus. Es necesario diseñar, desarrollar, implementar, configurar y probar un sistema capaz de orquestar un gran abanico de herramientas diseñadas cada una con un objetivo puntual, combinando las capacidades de todos sus subsistemas para identificar eficientemente las amenazas reales y responder en consecuencia, minimizando los falsos positivos y daños colaterales.

# 

# Marco Teórico

Pequeña introducción sobre los riesgos a los que se exponen las redes modernas. Esto me llevará a la necesidad de justificar un SOC. Luego para desplegar un SOC/CSIRT, tomó como referencia el documento del SANS Inst. y esto me da lugar de forma natural, a presentar un SIEM.

CSIRT/SOC, su organización, la propuesta del SANS.

SIEM, sus funciones,etc…

Soluciones propietarias de SIEM, las versiones libres a las que dan soporte.

Soluciones open source para implementar un SIEM. Mencionar más de una para poder justificar la opción por la que voy a utilizar.

Los distintos tipos de CSIRT me van a permitir justificar la construcción de un CSIRT en la UNC.

Las infracciones a las políticas de seguridad y los ataques han concentrado la atención sobre las capacidades de detección, investigación y mitigación de incidentes de las organizaciones. Si bien muchas veces no es posible evitar un incidente de seguridad, es necesario poder detectar y responder rápidamente para evitar un daño completo. Para ello es necesario realizar inversiones inteligentes basadas en un plan de seguridad de seguridad que comprenda la realidad y necesidades específicas de la organización, ya que un gran monto de dinero o equipos adquiridos por si solos no garantizan una mayor protección.

Este plan debe incluir personal especializado, procedimientos e infraestructura adaptados a la organización, con una gestión de objetivos a cumplir a corto, mediano y largo plazo.

Para las organizaciones sin una capacidad formalizada de manejo de incidentes, la creación desde cero de un centro de operaciones de seguridad (SOC) o también conocido como CSIRT (Equipo de Respuesta a Incidentes de Seguridad de Computación) que permita la visibilidad, las alertas y la investigación centralizadas puede ser un proceso muy complejo y costoso. Lo anterior no implica que para disponer de la capacidad de seguridad sea necesaria una gran inversión, ya que es posible desarrollar una solución a escala de la organización en base a un análisis apropiado.

El centro de operaciones de seguridad debe tener una perspectiva flexible y escalable para mantener el ritmo de las tácticas de los adversarios, acompañando el crecimiento y evolución de la organización. Para alcanzar este objetivo, una vez identificadas las necesidades de la organización, el proceso de creación del CSIRT requiere la colaboración y comunicación entre los tres pilares fundacionales de un SOC: el personal, la tecnología y los procesos como se muestra en la Figura 1.

|  |
| --- |
|  |
| Figura 1: Pilares de un CSIRT |

## Personal

En cuanto al personal, estos comprenden tanto a los encargados de dar respuesta a los incidentes como a los analistas del CSIRT. Si bien la propia organización puede designar a sus integrantes para asumir estas funciones, existen otras alternativas como la tercerización mediante empresas especializadas que proveen el servicio (llamadas MSSP por sus siglas en inglés) o contratar especialistas en respuesta a incidentes (IR) en el caso de una emergencia o un problema complejo. Otra vía consiste en la creación de equipos híbridos compuestos por personal perteneciente a la organización y especialistas de compañías especializadas. De acuerdo a una encuesta del SANS Institute del año 2014, el 61 % de las organizaciones relevadas manifestaron haber recurrido a personal de emergencia para cubrir incidentes críticos y el 58 % tenía un equipo de respuesta propio. Por lo que las organizaciones raramente cubren sus necesidades con miembros de su propia plantilla y en algunos casos las tareas recaen por completo en los servicios de personal externo. Esto se debe a que sin importar la estructura del equipo, el personal de un SOC debe contar con el entrenamiento necesario para tratar con los cambios constantes y en algunos casos desafiantes, de las tareas de análisis de seguridad, investigación, especialistas en amenazas y director de un SOC. La tabla 1 muestra las responsabilidades y la formación requerida para cada uno de los tipos de integrantes de un SOC.

|  |  |  |
| --- | --- | --- |
| Título profesional | Tarea | Entrenamiento requerido |
| Nivel 1 - Analista de alertas | Supervisa continuamente la cola de alertas; clasifica las alertas de seguridad; monitorea el estado de los sensores de seguridad y los puntos finales; recopila los datos y el contexto necesarios para iniciar el trabajo de Nivel 2. | Procedimientos de triaje de alerta; detección de intrusos; gestión de redes, información de seguridad y eventos y capacitación en investigación basada en host; y otra formación específica sobre herramientas. |
| Nivel 2 - Respondedor de incidentes | Realiza un análisis profundo de incidentes al correlacionar datos de varias fuentes; determina si un sistema crítico o un conjunto de datos se ha visto afectado; asesora sobre remediación; proporciona soporte para nuevos métodos analíticos para detectar amenazas. | Análisis forense de redes avanzado, análisis forense basado en host, procedimientos de respuesta a incidentes, revisiones de registros, evaluación básica de malware, análisis forense de redes e inteligencia de amenazas. |
| Nivel 3 - Experto en la materia | Posee un conocimiento profundo sobre redes, endpoints, inteligencia de amenazas, ingeniería forense e ingeniería inversa de malware, así como el funcionamiento de aplicaciones específicas o infraestructura de TI subyacente; actúa como un “cazador” de incidentes, sin esperar que se intensifiquen los incidentes; estrechamente involucrado en el desarrollo, ajuste e implementación de análisis de detección de amenazas | Entrenamiento avanzado en detección de anomalías; Entrenamiento específico de herramientas para la agregación y análisis de datos e inteligencia de amenazas. |
| SOC Manager | Administra recursos para incluir personal, presupuesto, programación de turnos y estrategia tecnológica para cumplir con los acuerdos de nivel de servicio; se comunica con la gerencia; sirve como persona de contacto organizacional para incidentes críticos para el negocio; proporciona una dirección general para el SOC y aporta información a la estrategia de seguridad general. | Gestión de proyectos, formación en gestión de respuesta a incidentes, habilidades generales de gestión de personas. |
| Tabla 1 | | |

Para orquestar el trabajo de los analistas, un SOC necesita un director que coordine los múltiples esfuerzos dentro del equipo, a veces inclusive en la respuesta a incidentes dentro y fuera del SOC. Su responsabilidad es priorizar el trabajo y organizar los recursos con el último fin de detectar, investigar y priorizar incidentes que puedan impactar en la organización. Otra de las misiones asignadas al director consiste en desarrollar un modelo de flujo de trabajo e implementar procedimientos operativos estandarizados para el proceso de manipulación de incidentes que guíen a los analistas a través de un proceso de clasificación y respuesta.

En la Figura 2 se observa una posible organización de un SOC.

|  |
| --- |
|  |
| Figura 2: Organización de un CSIRT |

## Procesos

Definir procesos repetibles de clasificación de incidentes e investigación, estandariza las acciones que pueden tomar los analistas del SOC y aseguran que no se perderán tareas importantes en el camino. Al crear un flujo de trabajo de gestión de incidentes repetible, se definen las responsabilidades y acciones de los miembros del equipo desde la creación de una alerta y la evaluación de nivel 1 hasta el escalamiento al personal de los niveles 2 o 3; esto incluye la asignación efectiva de los recursos.

Uno de los modelos de procesos de respuesta a incidentes más utilizado es el modelo DOE/CIAC, que consiste en seis etapas: preparación, identificación, contención, erradicación, recuperación y lecciones aprendidas.

## Tecnología

En el núcleo de un SOC se encuentran las tecnologías de recolección de datos, agregación, detección, análisis y administración. En cuanto a la recolección de datos, un sistema de monitoreo obtiene sus datos a partir de un conjunto variado de fuentes como puntos finales (PC, dispositivos móviles, servidores, etc), redes y generadores de logs y eventos. Como resultado de la disponibilidad de los datos de la red, logs y puntos finales, antes y durante el incidente, los analistas pueden utilizar el sistema de monitoreo como una herramienta de investigación revisando las actividades sospechosas del incidente en curso e incluso pueden usarla para gestionar la respuesta al mismo o sus causas. Es importante la compatibilidad de las tecnologías empleadas, en particular si la organización ya cuenta con una herramienta de monitoreo existente y se busca incorporar nuevos sistemas para integrarlos a las herramientas disponibles; la Figura 3 ejemplifica la necesidad de compatibilidad entre sistemas y componentes.

|  |
| --- |
|  |
| Figura 3: Compatibilidad entre tecnologías de detección |

### Agregando contexto a los incidentes

La incorporación de inteligencia de amenazas, activos, identidades y otras informaciones de contexto es otra manera en la cual una solución de monitoreo de seguridad puede ayudar al proceso de investigación del analista de un SOC. En determinados casos, una alerta es asociada con una red o una actividad relacionada con un host e inicialmente puede contener solo las direcciones IP del punto final sospechoso. Para que los analistas puedan investigar el sistema en cuestión, generalmente necesitan otra información, como el dueño y el nombre de dominio de la máquina, los registros DHCP para mapear la IP y la información del host al momento de la alerta. Si el sistema de monitoreo incorpora información de identidad y de los activos, provee de una gran ventaja en términos de esfuerzo y tiempo del analista, sin mencionar factores clave que el analista puede usar para priorizar los incidentes; en términos generales los activos de mayor valor deben priorizarse sobre los de menor valor.

### Definición de conductas normales

La habilidad de crear una referencia o línea base de la actividad de los usuarios, aplicaciones, infraestructura, redes y otros sistemas; por lo tanto establecer lo que se considera un comportamiento normal es una ventaja de la agregación de datos provenientes de varias fuentes. Contar con una definición de lo que se considera “normal”, facilita la detección de conductas sospechosas; es decir todas aquellas actividades que de alguna manera se encuentran fuera de la norma. Un sistema de monitoreo configurado y con una base de referencia adecuada envía alertas que pueden ser confiables y en ocasiones priorizadas automáticamente antes de llegar al analista de nivel 1. Sin embargo y, de acuerdo al citado informe del SANS Institute del año 2014, uno de los principales desafíos en la utilización de datos de log es la incapacidad de distinguir actividades sospechosas de las normales. La ausencia de una referencia de “normalidad” es un obstáculo común al que se enfrentan las empresas de monitoreo y muchas organizaciones.

La mejor práctica es usar plataformas que pueden crear líneas o patrones de referencia mediante el monitoreo de la red y la actividad de los puntos finales durante un periodo de tiempo, para determinar lo que es “normal” y proveer la capacidad de establecer umbrales de eventos como indicadores de alertas. Cuando un comportamiento inesperado una desviación de una actividad normal es detectada, la plataforma crea una alerta indicando que es necesaria una investigación más detallada.

### Inteligencia de amenazas

Los SOC bien establecidos o maduros desarrollan continuamente la capacidad de consumir y aprovechar inteligencia proveniente tanto de amenazas de sus incidentes pasados como de fuentes de inteligencia compartidas; ejemplos de estas últimas son los proveedores especializados de inteligencia de amenazas, socios industriales, divisiones policiales de cibercrimen, organizaciones de intercambio de información como la IT-ISAC, etc. La capacidad de un sistema de monitoreo para operacionalizar la inteligencia de amenazas y usarla para ayudar a detectar patrones en datos puntos finales, logs y registros de red, así como asociar anomalías con alertas del pasado, incidentes o ataques anteriores, pueden contribuir con el mejoramiento de la capacidad de detección de un sistema o usuario comprometido antes de que se produzcan nuevos incidentes.

### Obstáculos para el manejo eficiente de incidentes del SOC / CSIRT

Parte de los obstáculos que deben ser evitados por un SOC son aquellos que generan cuellos de botella en el proceso de respuesta a incidentes. Dado que este último consiste en el traslado de un incidente entre los niveles 1, 2 y 3 existe la posibilidad de crear “ruido blanco” debido a la presencia de una gran cantidad de alertas de poca importancia y / o falsos positivos; de prolongarse esta situación en el tiempo, se produce un fenómeno llamado “fatiga de alertas” que afecta a los analistas provocando una disminución en sus capacidades de atender alertas.

Al momento de elegir una herramienta de monitoreo, se debe considerar que incluya entre sus características la personalización del umbral de alertas y la posibilidad de combinar distintas alertas en un mismo incidente. Una herramienta de este tipo permite a los analistas clasificar las alertas más rápido, reduciendo las capas de evaluación necesarias antes de que el evento pueda ser confirmado y mitigado rápidamente.

### Multiplicidad de herramientas de monitoreo y su integración

A medida que los sistemas de conexión se vuelven cada vez más complejos y deben soportar redes 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 desarrolla un conjunto cada vez más numeroso y especializado de herramientas que intentan cubrir los 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; es necesario un sistema capaz de procesar, filtrar, normalizar, presentar y almacenar toda esta información. Este sistema es un SIEM.

## SIEM: Definición y sus funciones

El proceso de monitoreo de la seguridad de una red de datos compleja requiere recopilar diferentes tipos de datos para detectar, verificar y contener acciones ofensivas. Para ello se requieren tecnologías *Security Information and Event Management* (SIEM). Estas tecnologías proporcionan informes en tiempo real y análisis de eventos de seguridad a largo plazo, como se muestra en la Figura 4. Todo esto ayuda a la tarea de un analista de ciberseguridad cuando debe verificar acciones ofensivas sobre la red de una organización.

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.   
 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.

Como se puede observar en la figura 4, para cumplir con las tareas anteriormente descritas, un SIEM obtiene su información de diversas fuentes:

* Inteligencia de Amenazas, administración de identidades y almacenamiento de logs: la revisión de amenazas pasadas y la inteligencia compartida por otras organizaciones aliadas contribuyen a detectar más rápido patrones anormales de comportamiento.
* Telemetría de Netflow: es un protocolo de recopilacion de informacion del flujo de red IP.
* Captura de paquetes: replicación de paquetes de información con el fin de correlacionar posibles amenazas en el flujo de red.
* Dispositivos Antimalware: componentes especializados en la detección de malware, localizados en puntos finales.
* IDS (HIDS y NIDS): Sistemas de detección de intrusiones, orientados a redes (NIDS) o puntos finales (HIDS).
* Firewalls: software que bloquea y filtra conexiones con origen desde el sistema hacia el exterior y viceversa.
* IPS: Sistema de protección de intrusiones. Ofrecen protección activa frente a comportamientos inusuales ya que pueden tomar acciones programadas para evitar un intento de intrusión.
* Logs de servidores y servicios (Syslog): Syslog es un protocolo para el envío de logs en un formato estandarizado

|  |
| --- |
|  |
| Figura 4: Funciones y flujos de datos de un SIEM. |

Por lo tanto, la secuencia que sigue la información dentro un SIEM se puede graficar como se observa en la Figura 5

|  |
| --- |
|  |
| Figura 5: Procesos de acción de un SIEM |

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. La Figura 6 representa las responsabilidades primarias de un SIEM.

|  |
| --- |
|  |
| Figura 6: Capacidades Críticas de un SIEM |

## Análisis de herramientas disponibles

En los requerimientos no funcionales se mencionó que la implementación del SIEM debe realizarse con herramientas de software libre y gratuito. Sin embargo, en esta sección, se decidió analizar las principales soluciones que existen en el mercado, sean gratuitas, de software libre, de pago o código cerrado para tener una comparación más amplia y diversa.

Se realizó un relevamiento en sitios especializados y de referencia para el mercado mundial, con el objetivo de identificar las herramientas que actualmente dominan el mercado global de SIEM disponibles comercialmente. En el sitio web de Gartner se encontró un listado de los productos ordenados por valoración de los usuarios de dicha página. El resultado de los diez primeros productos junto a sus desarrolladores, es el siguiente:

|  |  |  |
| --- | --- | --- |
| 1) | QRadar SIEM | IBM |
| 2) | ManageEngine ADAudit Plus | ManageEngine |
| 3) | LogRhythm NextGen SIEM Platform | LogRhythm |
| 4) | LogPoint - SIEM | LogPoint |
| 5) | McAfee Enterprise Security Manager | McAfee |
| 6) | ArcSight Enterprise Security Manager (ESM) | Micro Focus |
| 7) | InsightIDR | Rapid7 |
| 8) | Elastic (ELK) Stack | Elastic |
| 9) | Splunk Enterprise | Splunk |
| 10) | Exabeam Security Management Platform | Exabeam |
| *Tabla 1: Ranking Gartner de soluciones SIEM 2020* | | |

En base a toda lo información que se recolectó se procedió a clasificar las herramientas en base a su naturaleza: pagas o libres .

|  |  |
| --- | --- |
| **Soluciones Pagas** | **Soluciones Libres** |
| **Splunk** |  |
| **McAfee Enterprise Security Manager** |  |
| **AlienVault USM** | **Graylog** |
| **QRadar SIEM** | **Elastic (ELK) Stack** |
| **ManageEngine ADAudit Plus** | **AlienVault OSSIM** |
| **LogRhythm NextGen SIEM Platform** | **Security Onion** |
| **LogPoint - SIEM** | **Sweet Security** |
| **ArcSight Enterprise Security Manager (ESM)** |  |
| **InsightIDR** |  |
| Tabla 2: Soluciones SIEM Comerciales y Libres | |

### Soluciones comerciales

En nuestro análisis de las soluciones comerciales disponibles, se decidió describir la situación del mercado internacional en términos de uso según los principales sectores industriales, la distribución geográfica de sus usuarios a nivel global y un énfasis en América Latina.

Respecto de los sectores industriales que hacen uso de sistemas de ciberseguridad y generan demandas de nuevas soluciones, se observó que el sector financiero a nivel mundial es el que lidera el consumo de soluciones SIEM, con la mayoría de los desarrolladores teniendo como clientes principales a empresas y organizaciones de ese sector. Esto se explica dado el alto nivel de digitalización de la banca y los servicios financieros, que por su masividad y naturaleza son objetivos prioritarios para cualquier atacante en el ciberespacio.

Finalmente, se hizo una comparación segmentada por características entre los principales productos SIEM del mercado internacional en base a las revisiones de sus usuarios que fueron recogidas por medios especializados y de referencia como “Gartner” y “Markets & Markets”.

|  |
| --- |
|  |
| Tabla 3. Valoración de las características |

En esta tabla comparativa se observó que FortiSIEM y ADAudit Plus, de las compañías Fortinet y ManageEngine respectivamente, son las soluciones mejor valoradas.

|  |
| --- |
| Gráfico |
| Figura 7 |

En las Figuras 7 y 8 se aprecia que el sector financiero es el principal demandante a nivel global, siendo las industrias manufactureras y de la salud las otras dos que se destacan con diferencia respecto al resto del universo de consumidores. Esto se debe a que la automatización de la industria primero y su evolución al actual modelo de “industrias 4.0”, con cadenas de producción, montaje y ensamble distribuidas geográficamente alrededor del globo, el uso masivo de sensores, redes y datos intrínsecos a cada fase de producción; produjo la necesidad de soluciones de ciberseguridad para evitar incidentes que pudieran afectar las líneas de producción, daños a la maquinaria e infraestructura y sobre todo evitar en la medida de lo posible el robo de información crítica o secretos industriales. De manera análoga al sector manufacturero, las clínicas, hospitales y centros de salud han sufrido el impacto de la digitalización de sus procesos tanto en el hardware médico, el almacenamiento y distribución de la información como en la protección de los sensibles datos privados de los pacientes y la estricta normativa que los regula.

Se pudo observar que hay un marcada distribución en los mercados segmentados geográficamente, como QRADAR, LogPoint y FortiSIEM concentrando su demanda en Europa, el Medio Oriente y África; mientras que Enterprise Security Manager, InsightIDR y AlienVault USM lo hacen en América del Norte y FortiSIEM en Latinoamérica. Sobre nuestra región, en la Figura 9 se observó que FortiSIEM acapara el 34 % del mercado, seguido de Enterprise Security Manager, ADAudit Plus y NextGen SIEM Platform con un 17, 14 y 13 % del mercado, respectivamente.

|  |
| --- |
| Gráfico |
| Figura 8 |

|  |
| --- |
| Gráfico |
| Figura 9 |

### Soluciones gratuitas y de código abierto

Para el análisis de soluciones libres se observó primeramente que no se cuenta con muchas variedad de productos que cubran esta necesidad, algunos inclusos como el proyecto *Sweet Security* no cuenta con soporte desde el año 2017. Luego se encontró las soluciones que son propietarias pero a su vez tienen su versión gratuita también, el inconveniente de estas versiones es que no pueden escalar en caso que se requiera por parte de la organización. Las limitaciones van desde la cantidad de logs que pueden procesar, hasta el límite de almacenamiento de los mismos.

Al igual que en el caso de las soluciones pagas, se realizó un análisis de cuáles son los sectores en la industria a nivel mundial donde predominan las soluciones gratuitas y se observó un uso más intensivo en las áreas de servicios, estos se puede deber a que las áreas de servicios generalmente, las manejan entes estatales en conjunto con instituciones privadas y prefieran modificar dichas soluciones para que se adapten a sus estructuras organizativas. En las figura 10 se puede observar el porcentaje de adopción por área en la industria.

|  |
| --- |
| Gráfico |
| Figura 10 |

Por otro lado, se realizó el estudio de la región y el porcentaje de uso de las soluciones y se pudo notar que la pila ELK en mayor o menor medida se utiliza en todas las regiones, como se puede ver en la Figura 11

|  |
| --- |
| Gráfico |
| Figura 11 |

Otra conclusión que se puede destacar del gráfico es que en américa del norte y del sur la solución SIEM más utilizada es AlienVault, mientras que en Europa, Medio Oriente y África es Graylog y en Asia/Pasifico optan usar la pila ELK.

También se puede remarcar que en Latinoamérica el segundo producto más utilizado después de AlienVault es Security Onion.

Cabe mencionar que tanto Graylog como Security Onion utilizan componentes de la pila Elastic. En el caso de Graylog, utiliza de base de datos a Elasticsearch que pertenece a dicha pila. Distinto es el caso de Security Onion, el cual utiliza la pila completa de Elastic ya que se trata de una distribución de Linux basada en Ubuntu que cuenta con un suite de diversas herramientas dedicadas a la ciberseguridad.

## Ámbitos de actuación de los CSIRT

En la actualidad existen en todo el mundo CSIRT pertenecientes a organizaciones que responden a distintos ámbitos de la sociedad y de diferente naturaleza (pública o privada). En términos generales, estos equipos se clasifican dependiendo de la comunidad a la que atienden, diferenciándose entre:

CSIRT para el sector de PYMES: En este caso, el tamaño de las empresas hace poco viable que las organizaciones de este sector puedan implementar de forma individual las funciones de un CSIRT. Por lo tanto, surge la necesidad de unificar esfuerzos y servicios en un solo CSIRT capaz de dar soporte a varias empresas. La naturaleza de estos CSIRT puede ser pública o privada, dependiendo del contexto en el que se encuentren estas compañías.

CSIRT académico: El área de responsabilidad de este tipo de equipos se circunscribe a instituciones académicas. Su tamaño, por lo tanto, puede variar dependiendo de las dimensiones de la comunidad, esto también condiciona los servicios que ofrezcan, el modo en que lo hagan y su grado de intervención.

CSIRT comercial: estos centros prestan distintos servicios a cambio de una contraprestación económica. Habitualmente utilizan acuerdos de servicios específicos con cada cliente.

CSIRT de proveedor: se centra en los productos o servicios específicos de un proveedor. Su objetivo es proveer servicios y soluciones para eliminar o reducir el impacto negativo de las vulnerabilidades de estos últimos, ya sea un producto tecnológico o un servicio TIC.

CSIRT del sector militar: Prestan servicios a organizaciones militares con responsabilidades en infraestructuras TIC necesarias con fines de defensa. Su comunidad está conformada por las instituciones militares y de entidades estrechamente relacionadas con éstas como, por ejemplo, del Ministerio de Defensa. Por ejemplo, en nuestro país es el Comando Conjunto de Ciberdefensa dependiente del Estado Mayor Conjunto de las Fuerzas Armadas.

CSIRT para protección de infraestructuras críticas: Los CSIRT de este sector se centran principalmente en la protección de las infraestructuras críticas y de las infraestructuras críticas de la información (administración, centrales y redes de energía, tecnologías de la información y las comunicaciones, sistema financiero y tributario, sector sanitario, espacio, instalaciones de investigación, alimentación, agua, transportes, las industrias nuclear y química entre otras).

CSIRT gubernamental: Bajo esta denominación se sitúan los equipos cuyo principal objetivo es asegurar la infraestructura TIC de un Gobierno/Estado y los servicios ofrecidos a la población. La Comunidad a la que están dirigidos son las administraciones públicas y sus distintos organismos. A su vez, esta comunidad puede dar servicio, a la administración local, provincial, nacional o a todas ellas. Estos CSIRT gubernamentales generalmente forman parte y están patrocinados por instituciones del Estado.

CSIRT Nacional: Este es un equipo con responsabilidad general de coordinación sobre todos los sectores y tiene una amplia responsabilidad sobre prácticamente todos los puntos tratados anteriormente. Este centro funciona como punto focal de contacto tanto en el entorno nacional como para requerimientos internacionales. ENISA, la Agencia de Ciberseguridad para la Unión Europea, define en un documento elaborado en diciembre de 2009, a este tipo CSIRT como aquel que actúa como el Punto Nacional de Contacto (POC) con otros CSIRT nacionales y/o internacionales. De hecho, podría considerarse como “CSIRT del último recurso”, por su papel de coordinación. En muchos casos el CSIRT nacional también actúa como CSIRT gubernamental o tiene su origen en él. De igual modo, señala el documento, el CSIRT nacional de facto suele ser aquel que ha sido establecido primero por un Gobierno y actúa como contacto con otros países. Cada CSIRT es único, en el sentido de que establece sus operaciones, su organización y su imperativo legal para satisfacer las necesidades de su país y su comunidad.

|  |
| --- |
|  |
| Figura 12: CSIRT Nacional como centro de coordinación |

Frente a esto, todos los CSIRT tienen un objetivo común, mantener seguras las redes de sus países. De este modo podemos concluir que, aunque cada CSIRT utiliza herramientas y procedimientos diferentes, todos comparten el mismo objetivo:

* + Designar un punto de contacto para la coordinación de la respuesta a incidentes.
  + Construir y mantener una red de contactos extensa, tanto nacional como internacional.
  + Monitorización de la situación actual y mejora de la concientización.

Es importante destacar que la constitución de un CSIRT nacional o gubernamental no es la única medida a tener en cuenta en una estrategia de ciberseguridad completa por parte de un Estado, pero sí una parte importante de la misma, teniendo en cuenta, además, que este tipo de equipos deberían asumir, además, la responsabilidad de la Protección de las Infraestructuras Críticas de Información (CIIP).

# Descripción de Requerimientos

La idea es presentar requerimientos desde arriba hacia abajo. Desde el nivel de sistema hasta el nivel detallado.

## Requerimientos funcionales

* Envío de alertas sobre eventos destacables mediante servicios específicos: es necesario configurar servicios idealmente ya utilizados por la organización para enviar notificaciones sobre las detecciones de anomalías o potenciales incidentes por parte del sistema, para que los responsables del CSIRT y los departamentos que lo requieran, puedan tener conocimiento sobre las actividades detectadas y dar una respuesta adecuada.
* Visualización de los eventos en un tablero de mando: un aspecto fundamental de un sistema SIEM es la capacidad de integrar la información que envían los distintos sensores que están monitoreando y procesando tanto el tráfico en cualquier punto de la red (NIDS) como en los hosts (HIDS), de una manera concentrada, filtrada y ordenada para que los analistas puedan tener una visión clara y global de lo que está ocurriendo en todo momento, con la posibilidad de tener a disposición todos los detalles necesarios de los eventos que sean de interés. Además, debe ser posible tomar las decisiones haciendo uso de otros subsistemas del CSIRT, con sus respectivas acciones, desde el mismo tablero de mando.
* Recolectar registros: Es un requerimiento fundamental que el sistema pueda almacenar los registros de todos los eventos o incidentes que sean necesarios y oportunamente seleccionados para su almacenamiento en las bases de datos.
* Implementación de un sistema de notificaciones y pedidos para el manejo de incidentes: es necesario implementar un sistema de pedidos, o en su defecto adaptar el que la organización posea, para notificar al área responsable de la infraestructura o activo comprometido y que esta sea capaz de tomar las decisiones que considere apropiadas; tanto para mitigar las posibles consecuencias de un ataque como para fortalecer los activos. También será útil para constituir un registro sobre los pedidos realizados, obteniendo de esta manera un historial auditable.
* Analizar en busca puntos críticos de la red con mayor detalle al que se posee actualmente: es necesario que el nuevo sistema permita identificar los puntos críticos de la red y sus vulnerabilidades, así como el aporte de información sobre ellos que actualmente no es posible recoger con los medios de relevamiento y análisis de tráfico disponibles.
* Almacenar eventos en una base de datos: el sistema debe almacenar la información normalizada de los eventos que procesa el SIEM, en una base de datos distribuida para posteriormente utilizar los registros en la correlación a demanda con otros eventos y proveer la capacidad de facilitar auditorías e investigaciones según sea requerido.
* Automatización de respuestas a incidentes de seguridad: una vez detectado un evento y confirmado su clasificación como hostil, es necesario responder de la manera más rápida y eficiente posible. Además, muchos ataques consisten en una multitud de eventos que si dependieran de una respuesta individual a cada uno de ellos y realizada por un operador humano, sería imposible defenderse del tipo de ataques que consisten en miles de eventos lanzados al mismo tiempo, por ejemplo un intento de DDoS. Por otro lado, existen incidentes que se repiten periódicamente, como los de reconocimiento, en los cuales se puede aprovechar su naturaleza altamente repetitiva para desarrollar una respuesta automatizada y no distraer la atención de los operadores del sistema, permitiendo que estos se centren en tareas de mayor valor.
* Soporte para múltiples usuarios: el SIEM y sus subsistemas podrán ser operados por múltiples usuarios y con distintos niveles de privilegios, se trate de personal del SIEM, de otras áreas del CSIRT como de miembros de otras unidades de la organización que requieran tener acceso a registros puntuales.
* Correlación de eventos: luego de detectar y eventualmente responder a incidentes, es importante contar con la capacidad de revisar las bases de datos y comparar los eventos actuales con los registros de incidentes del pasado, para establecer posibles correlaciones de firmas, tipo, objetivos u otros parámetros en tiempo real. Disponer de esta información es de gran importancia, ya que permite tener una visión y comprensión más profunda de la situación, lo que permitirá dar una respuesta mucho más efectiva.
* Normalizar eventos: el sistema debe ser capaz de filtrar los eventos provistos por los distintos sensores y normalizar la información para su posterior tratamiento, presentación y almacenamiento en las bases de datos respectivas.

## Requerimientos no funcionales

* Escalabilidad de la solución: el desarrollo propuesto para el SIEM debe tener una arquitectura tal que permita escalar o redimensionar el sistema de acuerdo a la evolución de la infraestructura sobre la cual debe estar desplegado. Para ello, debe considerarse un diseño basado en componentes, modular y en contenedores, para desplegar inmediatamente en una nube propia de ser necesario. Es necesario que la solución desarrollada admita en un primer momento, la capacidad de monitorear enlaces de 1 Gbps de ancho de banda.
* Instalación automatizada: desarrollar una secuencia de comandos que permita al sistema poder desplegarse de manera totalmente automatizada y sistemática en todos los servidores destinados por la organización. Para ello, se deben emplear lenguajes y metodologías que permitan el desarrollo de un proceso de instalación autónomo, la secuenciación de tareas para lograr una secuencia de etapas confiable, predecible y segura; transparentar el proceso de instalación y configuración inicial, minimizando el número de parámetros necesarios a configurar cuando el sistema entre en servicio. Además, el mecanismo empleado no debe contar con agentes para evitar la explotación de posibles vulnerabilidades en el proceso o requerir eventuales actualizaciones. Las técnicas y tecnologías empleadas deben permitir la orquestación de todas las configuraciones y mantenimiento que requiera el sistema y sus componentes, así como la entrega continua de mejoras y actualizaciones.
* Inventariar y clasificar los activos de la organización: es necesario investigar y realizar un relevamiento de los activos con los que cuentan las infraestructuras de red y de datos a fin de clasificarlos para poder organizar una estrategia que permita el despliegue y el uso más eficiente de los recursos del SIEM y del CSIRT en su conjunto, para garantizar el monitoreo más amplio posible, eliminar puntos oscuros o desprotegidos y obtener el proceso de respuesta más eficiente a eventos de seguridad de la información.
* Manipulación de las bases de datos resultantes para su agregación bajo demanda: es necesario adecuar la estructura de las bases de datos de manera que se encuentre de la manera más optimizada posible para recibir los logs y toda la información relacionada a los eventos. Las bases de datos deben cumplir condiciones de alta disponibilidad, particionado automático y diferentes niveles de persistencia en disco.
* Utilización de software libre: la solución propuesta debe contemplar el uso, configuración y desarrollo de software libre, cuyo código fuente pueda ser estudiado, modificado y utilizado libremente por la comunidad. Puede implementar licencias GPL, AGPL, BSD, Apache, GFDL, MIT y Creative Commons, según sea el caso de cada uno de los sistemas, subsistemas y componentes individuales.
* Uso de un sistema operativo de código abierto: es necesario que el sistema implementado se ejecute sobre un sistema operativo de código abierto, pertenecientes a las familias Linux o BSD.
* Trazado de los diagramas topológicos de la red de la organización: es necesario realizar un relevamiento de las conexiones de la infraestructura de red interna de la organización central, las de sus dependencias y la red entre las unidades geográficamente distribuidas si las hubiera. Deben incluirse la topología de las conexiones de salida a Internet.

# Iteración 1: “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

|  |
| --- |
|  |
| Figura 13: logo de Security Onion |

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 14: Security Onion: Arquitectura de alto nivel |

En la Figura 14 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 15: 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 16: 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 17: 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.

|  |
| --- |
|  |
| Figura 18: Conexión de componentes Elastic |

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 18 muestra la arquitectura de alto nivel del almacenamiento en Elasticsearch.

|  |
| --- |
|  |
| Figura 19: 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 20: 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 21: 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 21), 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 22), 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 23. 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 22: Plantilla para la creación de nuevos casos |

|  |
| --- |
|  |
| Figura 23: 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 24: Algunos de los analyzers disponibles en Cortex |

|  |
| --- |
|  |
| Figura 25: Ejemplos de responders utilizables en Cortex |

|  |
| --- |
|  |
| Figura 26: Casos de uso de gestión de usuario y configuraciones de TheHive |

|  |
| --- |
|  |
| Figura 27: Diagrama de casos de uso de alertas y casos |

|  |
| --- |
|  |
| Figura 28: Casos de uso de gestión de usuario y configuraciones de Cortex |

|  |
| --- |
|  |
| Figura 29: Casos de uso del tablero de Cortex |

### 

### 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 ciertos tipos de incidentes no radica solo en optimizar el tiempo y la capacidad de atención de los analistas humanos en tanto en cuanto un recurso escaso y valioso, sino fundamentalmente en la capacidad de responder con eficacia y eficiencia a determinados arquetipos de ataques potencialmente devastadores como (por ejemplo) un DDoS, donde miles de eventos (incidentes) se producen en simultáneo y la capacidad humana de responder eficazmente 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 30: 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. Esto permite realizar una primera integración con el hardware disponible

### 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 |
| Tabla 4: Comparación entre Snort y Suricata | | |

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 31: Arquitectura de Despliegue |

En la imagen superior (Figura 31) 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 2: “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 32: 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 33: Tráfico correspondiente a una dependencia, medido durante un día, obviando las horas donde este es casi nulo |

|  |
| --- |
|  |
| Figura 34: 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 35: Asistente de instalación de Security Onion 16.04 |

|  |
| --- |
|  |
| Figura 36: 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 37: 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 38: 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 3: “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 39: Estructura general de una regla |

Como se describió en los párrafos precedentes, como 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.

## Automatización de acciones

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 40:

|  |
| --- |
|  |
| Figura 40: Secuencia de respuesta automatizada |

# Iteración 4: “Testing y corrección para puesta a prueba”

## Test de detección de ataques varios

## Test de reportes de incidentes

## Test de acciones automáticas

# Conclusión

# Futuros trabajos

# 

# 

# 

# 

# Bibliografía

[1]CCNA Cybersecurity Operations. CSIRT Overview ,Chapter 13 Incident Response and Handling, 2018. [Consultado 6-12-2019].

[N] LACNIC: acerca de; <https://www.lacnic.net/966/1/lacnic/acerca-de-lacnic>

# Anexos