Proyecto de fin de estudios en la carrera Ingeniería Eléctrica

SensorNET

Red distribuida de sensores de Seguridad Informática

Resumen Ejecutivo

Autores: Emiliano Colina
Anselmo Hermida
Waldemar Pera

Tutores: Ing. Eduardo Cota
Ing. Alejandro Blanco

Fecha: 30/04/2012

Red distribuida de sensores de Seguridad Informática

Resumen Ejecutivo

Tabla de Contenidos

Tabla de Figuras

I. Resumen

El presente documento presenta los aspectos y conclusiones más relevantes del proyecto SensorNET incluyendo en sus anexos documentación específica respecto a sus estudios y realización.

Para su cumplimiento se recorrieron diversas etapas que abarcaron un estudio del estado del arte en sensores informáticos, como ser IDS/IPS, firewalls, honeypots, etc. Posteriormente, se realizó una comparativa y elección entre diferentes herramientas del tipo gestores de eventos de seguridad en la cual se consideraron herramientas propietarias y de código abierto. Luego un diseño teórico de una red distribuida de sensores informáticos, y finalmente una prueba de concepto desplegada en una red real.

El proyecto SensorNET tiene como objetivo brindarle al Ministerio de Defensa Nacional (MDN) un modelo de implementación de una red distribuida de sensores, como parte de la visibilidad del estado de seguridad de su infraestructura informática, así como de mejorar su capacidad de respuesta ante un ataque informático.

II. Agradecimientos

A nuestras familias y amigos, por su apoyo incondicional a lo largo del desarrollo de este proyecto.

A nuestros tutores, Ing. Eduardo Cota e Ing. Alejandro Blanco.

Al Jefe del Departamento de Sistemas de Información (DSI) del MDN Ing. Roberto Ambrosoni, así como también a los miembros del staff, señores Gabriel Baygorria (Seguridad Informática), Eduardo Gómez y Gonzalo Acuña (Administración de Servidores).

III. Introducción

El proyecto esta dividido en tres etapas. La primera consiste en un estudio de los diferentes tipos de sensores existentes para la seguridad informática y su trabajo colaborativo; la segunda es el diseño de una red distribuida de sensores informáticos para la red de la Dirección General de Secretaría de Estado (DGSE) del MDN; y la tercera es la implantación de una prueba de concepto mediante un piloto en la mencionada red.

En las secciones siguientes se tratarán la motivación, el alcance y los objetivos del presente proyecto, luego de las cuales se repasan los principales conceptos en seguridad informática. Finalmente se tratarán los temas específicos del proyecto SensorNET.

En el presente documento se hará mención a documentación que cubrió cada etapa. Aquella documentación de carácter público se anexa como parte de la documentación del proyecto. Adicionalmente, existe documentación de carácter reservado la cual se excluye en virtud del Acuerdo de Confidencialidad suscripto con el MDN.

III.1 – Motivación

En el marco del Decreto 452/009 de setiembre de 2009 y la Resolución CDH 62/010 de octubre de 2010, el MDN ha adoptado como base la “Política de Seguridad de la Información para Organismos de la Administración Pública”. En este sentido, el MDN está implementando un Sistema de Gestión de Seguridad de la Información (SGSI). Además, está trabajando en conjunto con la Agencia para el Gobierno Electrónico, la Sociedad de la Información y el Conocimiento (AGESIC), y organismos internacionales tales como el Comité Interamericano Contra el Terrorismo (CICTE) de la Organización de los Estados Americanos (OEA) en su Programa de Seguridad Cibernética

En función de lo antes expuesto, surge la necesidad por parte de la DGSE del MDN de lograr una mejor visibilidad del estado de su infraestructura informática, con el fin de poder observar la “salud” de la misma y tomar acciones en consecuencia.

Surge así el proyecto SensorNET, cuyo objetivo general es brindarle al MDN un modelo de implementación de una red distribuida de sensores, como parte de la visibilidad del estado de seguridad de su infraestructura informática.

III.2 – Alcance

La primer etapa del proyecto tiene como alcance el estudio del estado del arte, de diversas herramientas que permiten implementar una red de sensores de seguridad informática, así como la comparativa y elección de un gestor central (o por sus siglas en inglés SIEM – Security Information and Event Manager), para su aplicación en las etapas segundas y terceras del proyecto. Se incluye en esta etapa un reconocimiento

de la infraestructura informática involucrada así como de los requerimientos en seguridad del cliente.

La segunda etapa tiene como alcance el diseñar conceptualmente una red de sensores en el ámbito del MDN, y en particular para la DGSE, que cumpla con el objetivo de mejorar la visualización del estado de seguridad de su red.

La tercera etapa tiene como alcance el implementar un piloto como prueba conceptual de la herramienta seleccionada en la primer etapa. Dicho piloto deberá contar mínimamente de:

- Consola central.
- Gestor de logs/eventos.
- Aplicación con funciones de NIDS (Network Intrusion Detection System).
- Integración de eventos de una tercera fuente, como ser un firewall.

Se realiza además una evaluación de la performance del piloto, teniendo como objetivo el presentar recomendaciones respecto a la escalabilidad y usabilidad de la solución así como de lo colectado por la misma.

El presente proyecto no contempla la implementación de una red distribuida de sensores en producción.

III.3 – Objetivos específicos

En virtud del Objetivo y del Alcance propuestos para el presente proyecto, es que se definen los siguientes objetivos específicos:

- Elaborar un documento “Estado del Arte en Sensores de Seguridad Informática”, donde se estudien los diferentes tipos de sensores y herramientas existentes, sus principales usos y características, así como su viabilidad de integración a la red del MDN.
- Relevar la infraestructura informática de la Dirección de Servicios Informáticos (DSI), perteneciente a la DGSE, involucrada en el alcance del presente proyecto.
- Relevar junto con los responsables del MDN los requerimientos específicos de seguridad informática.
- Elaborar un documento “Estudio de herramientas para Redes de Sensores de Seguridad Informática” donde se comparen distintas herramientas tipo SIEM, y se seleccione la que mejor ajuste a la realidad que presenta la red del MDN.
- Diseñar una red de sensores informáticos para la red del MDN alcanzada por los objetivos del proyecto. En el mismo se considerarán los conceptos presentados en los documentos iniciales.
- El diseño y la implementación de un piloto, dentro de la red del MDN, conformado por una red de sensores más un SIEM, a fin de probar los conceptos expuestos en los documentos anteriores.
- Elaborar un documento “Reporte de Piloto”, donde se describa tanto el montaje de la red de sensores y el SIEM implementados, como las recomendaciones surgidas a partir de los eventos colectados.

III.4 – Reserva de la información

Dado el tipo de organización donde se desarrolla el proyecto, se requiere por parte del MDN el asegurar el carácter reservado de la información específica. La formalización de esta necesidad se plasma en la firma de un Acuerdo de Confidencialidad por parte de los integrantes del proyecto y sus tutores con la organización.

IV. Red de sensores de Seguridad Informática

IV.1 – Seguridad Informática

Las políticas de seguridad en cada organización estarán dadas por una combinación de requerimientos particulares, entre ellos su propio modelo del negocio y los requisitos de seguridad. La seguridad de la información tiene como objetivo la preservación de tres propiedades fundamentales sobres los sistemas de información, esto es: Confidencialidad, Integridad y Disponibilidad.

Del mismo modo, cada uno de los riesgos, amenazas y vulnerabilidades identificados suelen ser medidos o evaluados respecto a su capacidad para comprometer uno o varios de las propiedades anteriores.

A su vez las funciones que se deben asegurar en un sistema informático son:

- Reconocimiento: Cada usuario deberá identificarse al utilizar el sistema a fin de que sus acciones queden registradas con dicha identificación. Con esto se apunta a que toda manipulación de datos quede registrada.
- Integridad: Decimos que un sistema es integro cuando funciona correctamente en su totalidad.
- Aislamiento: Se refiere a que los datos utilizados por un usuario dado deben ser totalmente independientes de los de otro usuario.
- Auditable: Refiere a la generación de evidencia que pueda ser utilizada en procedimientos tales como auditorías, tests, demostraciones, o comprobaciones del sistema. Dichas comprobaciones no sólo deben brindar datos precisos sino que deben aportar confianza a sus administradores.
- Gestionable: Refiere a la capacidad de poseer elementos que permitan tener control permanente no sólo del sistema principal, sino de todos sus subsistemas.
- Recuperabilidad: Debe existir la forma de recuperar todos los recursos perdidos o dañados, luego de un desastre.
- Administración: Consiste en la vigilancia permanente de todos los sucesos ocurridos, a fin de poder hacer un seguimiento posterior de cualquier hecho.

Por otro lado, también podemos catalogar a las amenazas según su “momento”: antes, durante o después de un ataque. Así, y en base a estas, se definirán las contramedidas como las políticas para cada “momento” del ataque, encontrando:

- La Prevención: Son los mecanismos que aumentan la seguridad de un sistema (antes del ataque) durante su funcionamiento normal. Ej.: cifrado de archivos.
- La Detección: Son los mecanismos orientados a detectar violaciones a la seguridad durante el ataque. Ej.: Software o Hardware orientado a la auditoría.

- La Recuperación: Son los mecanismos orientados a retomar el control y establecer el normal funcionamiento del sistema luego de un ataque. Ej.: Copias de seguridad de archivos.

En base a estas políticas podemos hablar de la Fiabilidad que presenta un sistema informático y se la define como “la probabilidad de que un sistema se comporte tal y como se espera de él”.

Es importante aclarar que ningún conjunto de medidas adoptadas llegaran a ser 100% fiables. Sin embargo el comprender y conocer el sistema y su seguridad, ayudará a evaluar correctamente los riesgos, las vulnerabilidades, las amenazas y las contramedidas así como a decidir las tácticas adecuadas a las necesidades de seguridad. [SNEA11]

IV.2 – Tipos de sensores de seguridad

IV.2.1 – Introducción

En adelante, nuestro alcance se verá reducido y orientado a un ámbito menor de la seguridad de la información: la seguridad informática.

Es necesario manejar en forma sistémica y sistemática a la seguridad informática para cumplir sus objetivos. Desde este punto de vista, es necesario conocer el estado en que los activos se encuentran respecto a sus dimensiones básicas: confidencialidad, integridad y disponibilidad. Dada la complejidad de los sistemas computacionales y sus interacciones, esto no es posible sin la asistencia de herramientas adecuadas que permitan dar una respuesta medible y gestionable del estado de los mismos.

Una red de sensores de seguridad informática tiene, entre otros objetivos, el recolectar la mayor cantidad posible de datos permitiendo conocer, lo más cercano posible a tiempo real, cómo se están comportando los activos. Dado que es posible conocer de ante mano el nivel de riesgo de cada uno de ellos, la elección de los sensores deberá estar condicionada a dicho nivel. Cuanto mayor sea el nivel de riesgo del activo, mayores serán los controles a fin de salvaguardar el mismo.

Si bien los sensores de seguridad informática no necesariamente cumplen una función directa en la implementación de controles, pueden ser de valor fundamental a fin de demostrar el cumplimiento de los controles necesarios en algunos rubros específicos de la economía o gestión de la información. Asimismo, permiten operar en forma vigilante, dotando al equipo encargado de la seguridad de una organización de capacidades proactivas y reactivas ante la exposición necesaria y aceptada de los activos a los riesgos existentes.

En lo que resta de esta sección, presentaremos una definición de los que se entiende por una red de sensores. Establecida la definición bajo la cual actuaremos, procederemos a introducir los principales tipos de sensores así como una herramienta llamada SIEM (Security Information and Event Manager). Esta última nos permite consolidar la información de los diferentes sensores, no sólo facilitando una consola única para la visualización de la información presentada por ellos, sino permitiendo además la capacidad de correlacionar esta información y alcanzar una visibilidad general, aspecto que no es posible si se utilizara cada sensor en forma separada. [SNEA11]

IV.2.2 – Definición y Objetivos de una red de sensores

Como una primera forma de aproximarnos a una definición de red de sensores de seguridad informática comenzaremos definiendo mínimamente lo que es un entorno computacional interconectado: es un conjunto de entidades y dispositivos informáticos que comparten información.

Es interesante observar que podemos distinguir un conjunto más pequeño conformado de dispositivos específicos y que tiene como finalidad montar una red de vigilancia para una red mayor. Entonces, cuando nos referimos a una red de sensores de seguridad informática identificamos este conjunto de entidades y dispositivos informáticos cuya finalidad es la "vigilancia" de los componentes de la red.

Los dispositivos que integrarían esta red, en un principio, generarían y reportarían eventos; entendiendo por evento a toda información sobre sí mismo que el dispositivo "desee" compartir, que pueda ser transmitida de alguna manera, y que tenga una relevancia para la seguridad informática. Como ejemplos de información reportada podemos mencionar desde "capacidad disponible para cumplir con la tarea", "situaciones no esperadas" y hasta un simple pero tan relevante "encendido/apagado".

En virtud de lo presentado, encontramos una categoría general de elementos los cuales, básicamente e independientemente de su naturaleza, aplican como candidatos para integrar una red de sensores. Esta categoría general estaría compuesta de elementos activos de una red, tales como impresoras, centrales telefónicas, switches equipamiento de ruteo y de borde entre otros, sistemas operativos, aplicaciones y servicios de red, por ejemplo.

Pero existen otras categorías de dispositivos o componentes que debido a su naturaleza tienen una funcionalidad específica y orientada a la seguridad informática de un entorno computacional interconectado.[SNEA11]

IV.2.3 – Firewalls

IV.2.3.1 – Introducción

Los firewalls pueden proteger tanto computadores como redes corporativas de intrusiones hostiles desde Internet u otros orígenes, teniendo en cuenta que la ubicación geográfica del atacante no es parámetro restrictivo para la mayoría de los ataques.

Con esto en mente, los firewalls son una de las principales herramientas disponibles para mejorar la seguridad de una red. El término firewall engloba muchos equipos con distintas características y funciones. En los siguientes párrafos veremos sus funciones básicas y distintas arquitecturas, al menos en una aproximación básica.

IV.2.3.2 – ¿Qué es un firewall?

Un firewall protege computadores conectadas en una red de los ataques perpetrados por elementos hostiles que, mediante los mismos, intentan acceder a información propia de la organización que protege. Estos ataques pueden tener como objetivo el robo de información confidencial, la corrupción de la misma o la interrupción

del servicio suministrado por la organización en la forma de una denegación de servicio ya sea de única fuente o distribuido (DoS o DDoS).

Un firewall puede verse como un elemento que separara zonas con diferente nivel de confianza, realizando un arbitraje del acceso entre las mismas.

Desde un punto de vista macro, como función de la seguridad informática, el firewall es el componente de tecnología que implementa una política de control de tráfico expresada en su configuración.

El firewall puede formularse en base a un servidor corriendo un sistema operativo de uso general con la adición de algún componente de software para la función de firewall o a través de algún hardware especializado ejecutando software orientado a la función de seguridad.

En su implementación más básica, un firewall consta de dos interfaces: una hacia la red que desea proteger y otra hacia el "mundo exterior" (por ejemplo: Internet). En el caso de un firewall basado en hardware, el mismo se puede basar en un router que interconecte ambas redes y que utilice alguna funcionalidad de filtrado para controlar el tránsito de paquetes entre las mismas. Aquí, la protección impide que el daño devenido de un ataque informático pueda propagarse de un subred a otra.

Como indicáramos previamente, otra posible implementación se basa en componentes de software sobre un sistema operativo de uso general. La distinción entre ésta y la solución basada en hardware, generalmente se halla en la definición clara de los límites operativos del firewall. En el caso de las soluciones basadas en hardware, el fabricante del mismo especifica los límites de operación de la misma en base a la capacidad máxima de tráfico, las conexiones por segundo o los paquetes por segundo soportados. En el caso de las soluciones basadas en software, estos números pueden no ser tan fácilmente especificados.

Otras diferencias se basan en su forma de gestión. En general, los firewalls basados en hardware constan de una interfaz única para la gestión y la configuración del equipamiento. En el caso de las soluciones basadas en software, muchas veces los componentes poseen archivos de configuración diferentes, por lo que pueden presentarse algunas dificultades para la gestión de los cambios en los mismos. En el caso de muchas soluciones OSS, se ha trabajado para la creación de un "front-end" de configuración a fin de emular las facilidades de las soluciones basadas en hardware.

IV.2.3.3 – ¿Qué puede hacer un firewall?

Un firewall examina el tráfico entre dos subredes a fin de observar si el mismo cumple con cierto criterio y, en consecuencia, tomar una decisión con respecto a dicho tráfico. Esta decisión puede ser permitir que el tráfico lo atraviese y llegue a su destino, o puede optar por descartarlo (dropped). Esta decisión se efectúa tanto para el tráfico saliente como entrante.

Dado que el firewall esta "in line" con el tráfico a controlar, una de las facilidades que presenta es la capacidad de registrar los diferentes flujos que controla. De esta forma, las diferentes ocurrencias ya sea de tráfico permitido o bloqueado, según sus reglas de operación, quedará registrada en logs.

IV.2.3.4 – ¿Qué no puede hacer un firewall?

Un firewall no puede proteger el tráfico que no pasa a través de él. Esto puede ser porque el usuario utilice un módem directamente conectado a su PC (ya sea un módem analógico o un módem 3G), o que simplemente se encuentran dentro de la misma zona de seguridad. Muchas veces las organizaciones implementan firewalls también en su red interna, aunque generalmente el mismo protege los recursos de cómputo centrales (Server Farm o Data Center), pero no así los segmentos de usuarios. En este caso, si algún malware se propaga entre aquellos segmentos que no están protegidos por el firewall, como los segmentos de usuarios finales, éste nada podrá hacer.

Un firewall tampoco podrá proteger contra ataques a través de tráfico permitido. Si la política de la organización permite el acceso a un sitio Web particular y el mismo ha sido comprometido de forma tal que los usuarios, al visitarlo, son infectados por algún tipo de malware, la protección dependerá si el firewall posee o no capacidades de poder analizar el tráfico entre el sitio Web y los clientes que acceden al mismo a fin de bloquear el ataque.

Otro caso es si existe un mal manejo o uso demasiado laxo de contraseñas y usuarios, poco nada puede hacer el firewall contra este tipo de hechos. Los mismos deberán ser procesados por otro tipo de herramientas de seguridad informática y, especialmente, por una cultura de seguridad que deberá alcanzar a todos los integrantes de la organización.

Aunque nuestro firewall posea capacidad de análisis de detección de malware, tampoco es posible que detenga todos los tipos de malware existentes o por crearse. Si bien muchas de las herramientas clásicas para la detección de malware, como ser los antivirus de hosts o aquellos que pueden integrarse a alguna solución de firewall pueden alcanzar altos valores de detección, ninguna puede alcanzar una efectividad del 100 %, sin importar la alta calidad del producto utilizado. Nuevamente, los firewalls constituyen un elemento más y no el único a la hora de desplegar herramientas dentro de una organización con la finalidad de implementar controles sobre la seguridad informática.

IV.2.3.5 – Filtrado de paquetes

La función más básica que realiza cualquier tipo de firewall es el filtrado de paquetes. Sea cual sea la tecnología o paradigma utilizado en el firewall, la primera idea que viene a la mente cuando hablamos de control del tráfico en una red, es la capacidad de filtrar en función de algunas reglas definidas.

En este sentido los llamados firewall “stateful” incorporan inteligencia de capas superiores a capa 3, capa de red. Dicha inteligencia puede ser extendida a fin de no sólo ser un soporte para estos casos, sino para permitir también la aplicación de políticas de seguridad o protecciones a protocolos a nivel de la capa de aplicación.

IV.2.3.6 – Capacidades de log

Una de las funciones esenciales en la seguridad informática, es la capacidad de que los sistemas puedan generar trazas durante su operación. Los firewalls no pueden ser ajenos a este requerimiento, por lo que es necesario que los mismos dispongan de dicha capacidad.

Este conocimiento debe ser confiable en cuanto a su marca de tiempo, máxime cuando no solamente los eventos del firewall son los registrados, sino también los de otros recursos o "sensores" en el contexto de aplicabilidad de nuestro proyecto. Esto significa que deberá existir una coordinación de la base de tiempo, siendo la forma más habitual de la misma, la implantación de un servidor de NTP.

Dado que tanto el propio firewall puede ser el objetivo de un ataque, o por la mera previsión de que el mismo sea comprometido, la recomendación es que los registros de eventos sean almacenados fuera del mismo. Para esto deberán existir mecanismos y protocolos que permitan realizar el envío de logs. Un ejemplo corresponde al protocolo Syslog. [SNEA11]

IV.2.4 – IDS-IPS

IV.2.4.1 – Introducción

La necesidad de mecanismos adicionales de seguridad como ser los sistemas de detección y prevención de intrusos se basan en la idea de que los atacantes son capaces de violar nuestra seguridad perimetral.

Así, una forma de mejorar la seguridad es la de instalar sensores dentro de nuestras zonas de confianza. Estos tendrán como objetivo avisar y/o actuar al momento de detectar una intrusión o un ataque a la seguridad.

De esta forma distinguimos tres tipos de atacantes:

a) los que "desde fuera" logran violar nuestra seguridad perimetral;

b) los usuarios malintencionados que atacan "desde dentro" nuestra red;

c) una combinación de las anteriores donde usuarios descuidados colaboran sin saberlo con un atacante externo.

Partimos entonces del supuesto de que los atacantes son capaces de atacar y llegar a comprometer parcial o totalmente los recursos de nuestra red.

Para esto los mecanismos para la detección de ataques e intrusiones tratan de encontrar y reportar la actividad maliciosa, pudiendo llegar a reaccionar adecuadamente ante un ataque, ya sea por sus propios medios o en colaboración con otros equipos. [GarPer04]

Dentro de los elementos existentes para la detección de ataques e intrusiones se destacan los Sistemas de Detección/Prevención de Intrusos.

Éstos a su ves podemos separarlos en:

- Sensores basados en equipos, llamados IDS de Host o HIDS; los cuales procesan información relacionada exclusivamente con el host

- Sensores basados en red, llamados Net IDS o NIDS, los cuales recogen y analizan información de eventos a nivel de tráfico de todo un sector de red.

IV.2.4.2 – Sistemas de detección de intrusos (IDS)

Comenzaremos definiendo lo que entendemos por Intrusión y por Detección de intrusiones, para luego comentar como logran estas entidades el cumplir con su objetivo.

[35][17] Intrusión: La entenderemos como la cadena de acciones realizadas ya sea por un usuario o proceso cuya finalidad sea el

provocar un acceso no autorizado con cualquier fin, a un equipo o sistema.

35 17 Detección de intrusiones: Proceso mediante el cual el sistema de detección identifica y responde ante una acción no autorizada contra uno o varios recursos de la red [GarPer04].

Resumidamente podemos decir que un Sistema de Detección de Intrusos es un dispositivo o pieza de software utilizado para detectar accesos no autorizados a un computador o a la red. Está integrado por sensores, los cuales monitorean y analizan el tráfico de la red con el objetivo de detectar posibles ataques o intrusiones no debidas. De esta forma, un IDS no solo analiza el tráfico, sino su comportamiento y contenido.

IV.2.4.3 – Arquitectura general de un sistema de detección de intrusos

La arquitectura básica de un IDS se puede resumir en los siguientes cuatro bloques:

1) Recolectores de información: tienen como función el colectar la información relevante a la seguridad de los equipos monitorizados. Dicha información deberán traducirla a un formato adecuado (como una secuencia de eventos) entendible por los procesadores de eventos. La información almacenada en estos eventos será la base de decisión para la detección del IDS.

2) Procesadores de eventos: conforman la inteligencia del sistema de detección, ya que serán quienes decidan, en base a la información recogida por los sensores, si se está frente a un ataque. Los mecanismos de detección más implementados son:

- modelo de detección de anomalías.(mediante los eventos colectados observa el comportamiento de un usuario, proceso o servicio y lo compara contra un comportamiento de perfil predefinido)
- modelo de detección de usos indebidos (comparan los eventos enviados por los sensores contra firmas de ataques conocidos).

3) Unidades de respuesta. Típicamente se atribuyen a los IDS las respuestas pasivas y a los IPS las activas. Las respuestas pasivas tienen como objetivo el generar una alerta sin tomar ninguna acción tendiente a evitar o mitigar la causa de la alerta, mientras que las respuestas activas tienen por fin el neutralizar las acciones que causan la alerta.

4) Elementos de almacenamiento. El volumen de información colectada por los sensores de detección es tal que se hace necesario el uso de sistemas de almacenamiento.

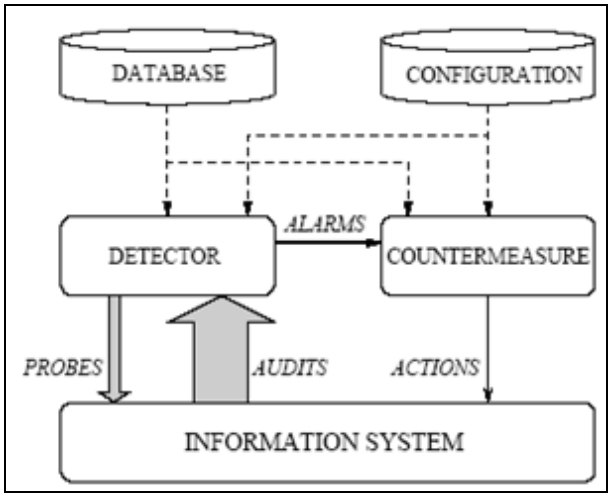


Figura 1 - Arquitectura básica de un IDS [CoLaOrPu05]

En la Figura 1, el calibre de la flecha representa la cantidad de información que fluye desde un componente hasta el otro.

Así, para detectar intrusiones en un sistema, los IDS pueden llegar a utilizar hasta tres "tipos" de información: 1) Información colectada previamente de ataques previos, 2) Información sobre la configuración actual del sistema, y 3) Información que describe el estado actual en términos de comunicación y procesos. [CoLaOrPu05]

IV.2.4.4 – Sistemas de prevención de intrusos (IPS)

Mientras los sistemas de detección de intrusos son en general sistemas pasivos, es decir, esperan a que tenga lugar un ataque para emitir una alerta, los sistemas de prevención de intrusos son sistemas con la capacidad (o al menos eso se espera de ellos) de detener un ataque o intrusión antes de que este pueda causar daños. Los sistemas de prevención de intrusos (IPS) son el resultado de unir las capacidades de bloqueo, como ser la de los firewall, con las capacidades de análisis y monitorización de los sistemas de detección de intrusos. [GarPer04]. Como puntos en común, encontramos que ambos sistemas, IDS e IPS, comparten la misma metodología básica de detección. Incluso un IDS puede llegar a ser un sistema de seguridad activo al trabajar en conjunto con un firewall.

Los IPS disponen de una unidad de respuesta capaz de actuar ante los ataques detectados, encargándose de bloquear los paquetes sospechosos tan pronto como son identificados. Así, todos los paquetes que pertenezcan a una misma sesión sospechosa serán eliminados o modificados de forma tal que sean inocuos. Lo mismo aplicará para los protocolos si el IPS es capaz de detectar anomalías en los mismos, como ser paquetes manipulados maliciosamente.

Aquí también encontramos sistemas basados en host (HIPS) y en red (NIPS).

Otra forma de clasificación para los IPS análoga a la de los IDS es según su forma de detectar el tráfico malicioso, o sea basado en usos indebidos o en anomalías.

Finalmente, es importante destacar que si bien los IPS no son capaces de "estudiar" el contenido en comunicaciones cifradas, esto no bloquea necesariamente sus capacidades para detectar o detener intrusiones sobre las mismas. En este

sentido la tarea consistirá en entrenar al dispositivo para que alerte en caso de detectar comunicaciones cifradas no autorizadas. [SNEA11]

Los modelos más relevantes de sistemas de prevención de intrusos son tratados en [SNEA11].

IV.2.5 – Honeypot

IV.2.5.1 – Definición

Un honeypot es un recurso de seguridad cuyo valor recae en ser escaneado, atacado o comprometido. [Spit02]

Un honeypot es un recurso que simula ser un objetivo real. De un honeypot se espera que sea atacado o comprometido. Las principales metas son la distracción de un atacante y la obtención de información sobre un ataque y el atacante. [BauPla02]

Las citadas definiciones, que en sí aparentan ser muy generales, dan dos rasgos característicos de un honeypot. El primero de ellos es su función como "carnada" o herramienta de distracción para los atacantes donde se espera que sean escaneados y atacados. El otro rasgo característico es su flexibilidad; como vemos, ambas definiciones coinciden en que un honeypot es un recurso, entendiéndose por tal a una aplicación, una máquina o un sistema implementado en una red como señuelo para atacantes y que pretende ser un sistema legítimo.

Típicamente un honeypot emula sistemas reales que corren servicios y cuyo fin son el atraer la atención de los atacantes, de forma que éstos insuman tiempo y recursos tratando de comprometerlo, mientras que esta actividad es monitoreada y registrada.

De esta forma, el honeypot capturará información altamente valiosa de un ataque, entre otras: su origen, mecánica, taxonomía, vulnerabilidades explotadas y herramientas usadas, tan sólo por mencionar algunas.

De la información colectada y su análisis posterior devendrán mejoras a implementar en los sistemas reales para su protección e incluso evidencias para un futuro proceso judicial.

IV.2.5.2 – Clasificación

Un honeypot puede ofrecer diversas funcionalidades y es en el nivel de interacción que ofrezca a los atacantes donde se distinguen, esperándose lograr una escala granular con la cual medir y comparar los honeypots. Cuanto más un honeypot pueda hacer y cuanto más un atacante pueda interactuar con este, mayor será la información que se pueda derivar de esta interacción. En consecuencia, y por la misma razón, mayor será el daño potencial que un atacante puede hacer.

En este sentido se exponen a continuación los tres niveles de interacción[1].

[1]Si bien los honeypots pueden ser clasificados según su objetivo o uso, la clasificación que entendemos más acertada es según su nivel de interacción.

IV.2.5.2.1 – Honeypots de Baja Interacción

Un honeypot de baja interacción proporciona ciertos servicios falsos, como se explica en [Baum02]. En una forma básica, estos servicios sólo pueden ser implementados mediante la instalación de una entidad a la escucha en un puerto específico, como se ilustra en la Figura 2, limitándose los servicios ofrecidos a dicho puerto.

En este tipo de honeypots, de baja interacción, no hay sistema operativo como objetivo real con el que un atacante pueda interactuar. Son utilizados para generar registros o alertas ni bien se tienen paquetes entrantes al sistema, ya que por definición de honeypot, toda interacción con éste se considerará como actividad maliciosa. Si bien se minimizará el riesgo de manera significativa, ya que la complejidad de un sistema operativo es eliminada, esta característica también es una desventaja, ya que no es posible ver un atacante interactuar con el sistema operativo. Según explica Baumman, un honeypot de baja interacción es como una conexión en una sola vía, puesto que está sólo escuchando, jugando un rol pasivo y no alterando el tráfico.

IV.2.5.2.2 – Honeypots de Media Interacción

Un honeypot de media interacción ofrece capacidades de interacción mayores que los de baja interacción, pero no proporciona ningún objetivo real subyacente al sistema operativo, como se muestra en la Figura 3. Los servicios falsos –y sus daemons respectivos- son más sofisticados teniendo un conocimiento más profundo acerca de los servicios específicos que ofrecen. En general, el atacante obtiene la ilusión de que existe un sistema operativo real con el que tendrá más posibilidades de interactuar y probar el sistema.

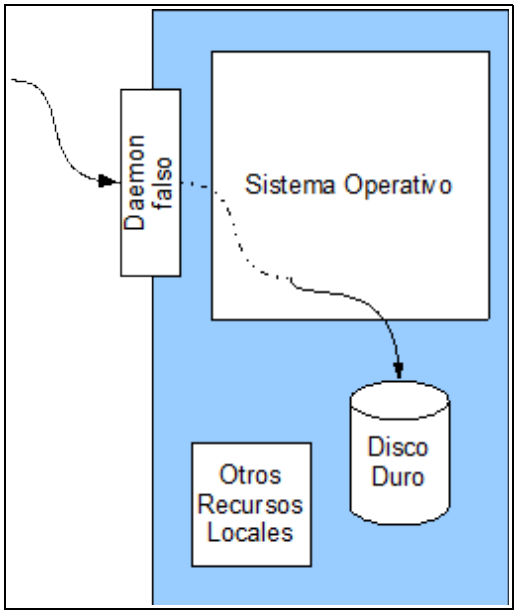


Figura 2 - Honeypot de baja interacción

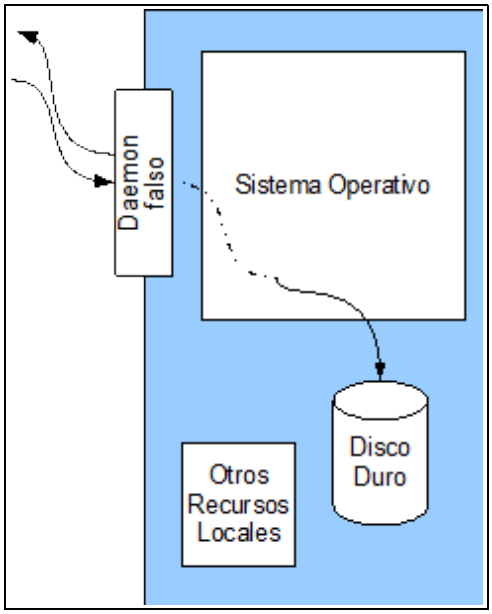


Figura 3 - Honeypot de media interacción

Este tipo de honeypots deberá ser asegurado y se deberá tener especial cuidado en los controles de seguridad implementados puesto que, si bien todos sus servicios son emulados mediante scripts, dichas soluciones presentan algunos riesgos y podrían llegar a ser víctimas de algún ataque, como ser buffer-overflow o denegación de servicio.

IV.2.5.2.3 – Honeypots de Alta Interacción

Un honeypot de alta interacción tiene un verdadero sistema operativo subyacente para ofrecer al atacante, como se ilustra en la Figura 4. Esto conduce a un riesgo mucho mayor a medida que aumenta la complejidad. Por otro lado, las posibilidades de recoger información de los posibles ataques y su atractivo como objetivo de ataque aumentan considerablemente. El objetivo del atacante será intentar obtener la mayor cantidad posible de privilegios en dicha máquina objetivo; por lo que, al proporcionar un sistema operativo completo, se le ofrece al atacante la posibilidad de levantar e instalar nuevos servicios y aplicaciones. Esto implica que el sistema debe estar bajo vigilancia permanentemente. Todas las acciones pueden, y deben ser registradas y analizadas para obtener más información acerca del ataque y los atacantes. En [BauPla02] se hace referencia a que es el principal objetivo de un honeypot de alta interacción y su despliegue en un entorno computacional hace legítimo un riesgo mayor.

Lance Spitzner explica en [Spit02] que "la percepción detrás de baja interacción vs alta interacción es la intención". En un honeypot de baja interacción la intención que motiva su instalación es la de limitar al atacante sólo a los servicios emulados que le son presentados, mientras que al implementar honeypots de alta interacción la intención es ofrecerle al atacante acceso al sistema operativo completo. En ambos casos se requiere de un sistema operativo real, sin embargo, en el caso de los honeypots de baja interacción el objetivo es limitar el atacante a la interacción sólo con los servicios emulados y no se le permitirá acceder al sistema operativo.

Cuanto mayores y complejos sean los scripts diseñados, más alto será el nivel de interacción que el honeypot deberá ofrecer. Los honeypots de alta interacción por excelencia son lo que conocemos como honeynets, los cuales definiremos en la próxima sección.

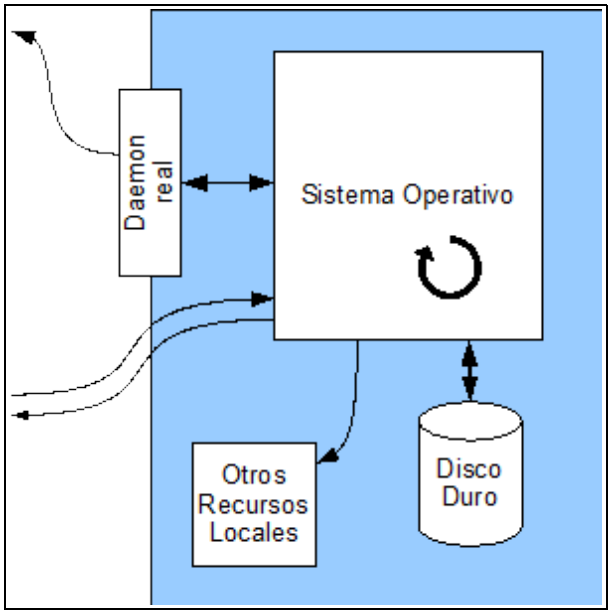


Figura 4 - Esquema de Honeypot de alta interacción

IV.2.5.3 – Honeynets

Tradicionalmente, la seguridad de la información ha sido principalmente defensiva. Firewalls, sistemas de detección de intrusos, encriptación, y otros mecanismos los cuales se usan defensivamente para proteger a los propios recursos conforman este enfoque defensivo. La estrategia es: *la defensa* de la propia organización de la mejor manera posible y *el detectar* posibles fallos en esta defensa, para luego *reaccionar* frente a estos fallos. El problema con este enfoque es que el enemigo tiene la iniciativa. Los Honeynets es un intento para cambiar esto.

IV.2.5.3.1 – Definición

Una honeynet es una herramienta de investigación que consiste de una red diseñada específicamente con el propósito de ser comprometida, con mecanismos de control que previenen a esta red ser usada como base para lanzar ataques contra otras redes. [HoeSteMon03]

Una Honeynet es una arquitectura. Esta arquitectura crea una red altamente controlada, en la cual se puede controlar y monitorear toda la actividad que ocurre dentro de ella. [KYE06]

En concreto, se trata de un honeypot de alta interacción diseñado para capturar información detallada sobre las amenazas. Esta “red trampa” proporciona una red con sistemas reales, aplicaciones y servicios para que los atacantes interactúen con estos, a diferencia de los honeypots de baja interacción que prestan únicamente servicios emulados. Es a través de esta alta interacción que se obtiene información sobre las amenazas, tanto externas como internas a una organización.

Estos sistemas víctima (honeypots en la honeynet) pueden ser cualquier tipo de sistema, servicio o información que se desee ofrecer, y pueden ir desde bases de

datos, portales web, servicios de transferencia de archivos, servidores de correo electrónico e incluso equipamiento de networking. Es por ello que su verdadero poder radica en su flexibilidad.

IV.2.5.3.2 – Arquitectura de una Honeynet

Para llevar adelante una implementación correcta de un honeynet se debe tener en claro su arquitectura, la cual tiene como elemento clave lo que conocemos como Honeywall. Se trata de un dispositivo que oficia de puerta de enlace o pasarela (gateway), separando la red de honeypots del resto del mundo y en el que todo el tráfico que va hacia o desde los honeypots debe pasar por éste. En la Figura 5 se presenta un diagrama de esta arquitectura donde el Honeywall tiene tres interfaces. Las primeras dos interfaces (eth0 y eth1) son los que separan los honeypots de todo lo demás, estas interfaces ofician de puente o bridge y no tienen IP. La tercera interfaz (eth2, que es opcional) tiene una IP que permite la administración remota.

Como mencionábamos, la pieza clave en una honeynet es el Honeywall el cual deberá cumplir las tareas fundamentales de control, la captura, el análisis y la recopilación de datos.

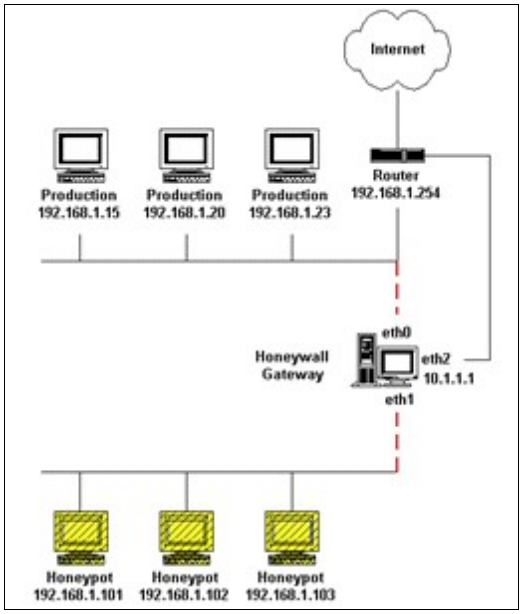


Figura 5 - Esquema de una Honeynet

El *control de datos* implica las funciones de contención y control de la actividad que realiza el atacante, sin que éste lo sepa. Esta tarea es la más importante ya que es lo que mitiga el riesgo.

La *captura de datos* es la supervisión y el registro de todas las actividades del atacante o del código malicioso dentro de la honeynet. Esta es la información capturada y que luego será analizada para aprender sobre las motivaciones, las herramientas y las tácticas empleadas por los atacantes.

Análisis de datos es el tercer requisito. Dado que la finalidad o propósito de una honeynet es la información, la misma carece de valor si no tiene la capacidad de

convertir los datos recolectados a información. Por lo tanto se deberá tener cierta capacidad para analizar los datos. Diferentes organizaciones tendrán diferentes necesidades, y requisitos para el análisis de datos.

La *recopilación de datos* sólo se aplica a organizaciones que tienen múltiples honeynets en entornos distribuidos.

IV.2.5.4 – HoneyClient o Client Honeypot

Son un tipo especial de honeypots que tienen como rasgo característico el ser activo y cuya principal funcionalidad es la búsqueda de servidores maliciosos que atacan a los clientes de un servicio. Habitualmente, su foco está en los navegadores web, simulando ser uno de éstos e interactuando con un servidor web para determinar si un ataque se ha producido; pero cualquier cliente que interactúe con un servicio brindado por un servidor puede ser parte de un honeyclient o client honeypot: por ejemplo ftp, ssh, correo electrónico, entre otros. [Danf06]

Análogamente a los honeypots tradicionales, estas soluciones se clasifican principalmente por su nivel de interacción: de alta o baja interacción, lo que denota el nivel de interacción funcional que el client honeypot tendrá con el servidor a investigar.

IV.2.5.4.1 – Arquitectura de HoneyClient

Un client honeypot se compone de tres componentes. El primer componente, es el Queuer, el cuál es el responsable de crear una lista de servidores para que el cliente visite. Esta lista se puede crear, por ejemplo, a través del crawling (rastreo). El segundo componente es el propio cliente, que es capaz de hacer peticiones a los servidores identificados por el Queuer. Después de la interacción con el servidor interviene el tercer componente, el cual es un motor de análisis y que será el responsable de determinar si un ataque ha tenido lugar contra el client honeypot.

Además de estos componentes, los client honeypots suelen estar equipados con algún tipo de estrategia de contención para evitar la propagación de ataques con éxito más allá del mismo. Esto se consigue normalmente mediante el uso de firewalls y sandboxes en la máquina virtual que lo ejecuta.

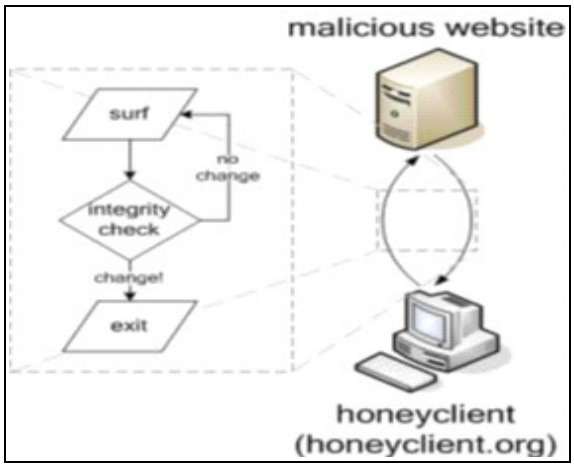


Figura 6 - HoneyClient , esquema de funcionamiento

En la Figura 6 presentamos HoneyClient, el cual es un client honeypot o honeyclient de alta interacción perteneciente al proyecto Honeyclient.org. Es una solución open-source usada para detectar malware en sitios web. La herramienta usando un crawler (rastreador) propio visita un sitio web y mientras no detecte cambio alguno en sí misma, a través de los chequeos de integridad en sus configuraciones, continúa navegando hacia otro sitio, los cuales extrae de una lista pre-cargada.

Ya en la Figura 7 presentamos HoneyMonkey, la solución Client Honeypot de Microsoft. Inicialmente el sistema navega con un nivel de parches de seguridad bajo, y cuando detecta un ataque eleva este nivel y vuelve a visitar el sitio malicioso. Esta es la forma que utiliza para la detección de vulnerabilidades de día-cero o zero-day exploits.

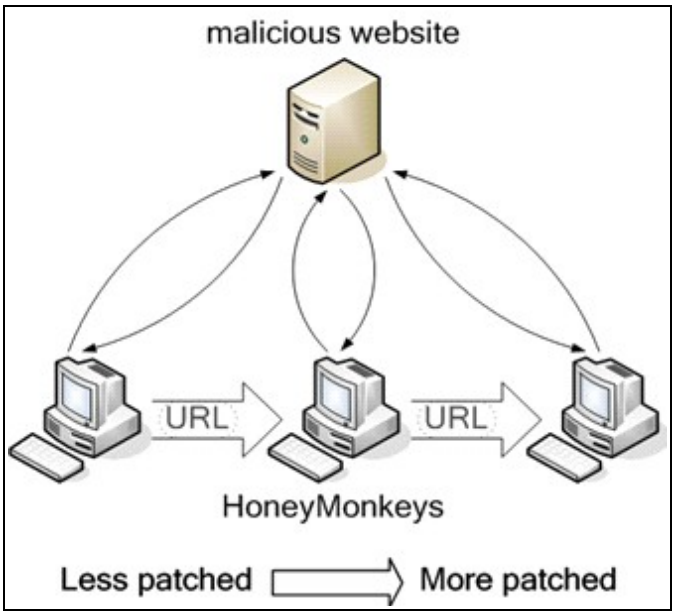


Figura 7 - HoneyMonkey, el Client Honeypot de Microsoft

IV.2.5.5 – Honeytoken

Para cerrar esta sección, presentaremos el concepto de honeytoken, el cual fuera introducido por Augusto Paes de Barros en el año 2003, y cuya idea proporciona un interesante mecanismo y de gran aporte para la seguridad de la información en las organizaciones.

IV.2.5.5.1 – Definición

Como hemos visto a lo largo de este capítulo, un honeypot no es únicamente un sistema o un equipo con el que un atacante o amenaza interactuará. Como definía Spitzner, “es un recurso del sistema de información cuyo valor reside en el uso no autorizado o ilícito de ese recurso”, y por ser un recurso y no explícitamente una computadora podríamos referirnos a una entidad digital. De esta forma, cuando es detectada una interacción con esta entidad digital es que podemos indicar que ha

habido un ataque; y es esto lo que representa un honeytoken, un tipo de honeypot que no es un dispositivo y que sirve para detectar un uso no autorizado o ilícito del mismo.

Bajando un poco más a tierra esta idea, y desarrollando el concepto de entidad digital, un honeytoken puede ser un número de tarjeta de crédito, una planilla de cálculo, un documento de texto, un registro en una base de datos, o incluso un login falso. Los honeytokens pueden ser muy diversos pero todos comparten el mismo concepto: un recurso digital del sistema de información cuyo valor reside en el uso no autorizado del mismo; nadie deberá utilizarlo o accesarlo. Esto no sólo da a los honeytokens el mismo poder y ventajas que los honeypots tradicionales, sino que se extienden más allá de las capacidades de equipos físicos.

IV.2.5.5.2 – Funcionamiento

Un honeytoken es como una trampa, una vez ubicado nadie debería interactuar con él; por lo que cualquier interacción con un honeytoken que se detecte, lo más probable es que represente una actividad no autorizada o maliciosa. Lo que se utilice como un honeytoken, y la forma de su uso, dependerá de la organización que lo implemente y qué recursos o activos se deseará monitorear.

Por ejemplo. ¿Cómo se detecta cuando se tiene un acceso no autorizado a una base de datos cuando la base de datos cuenta con miles y miles de registros así como con cientos de usuarios autorizados? El mantenimiento de quién está autorizado a acceder a qué información puede tornarse una tarea bastante compleja.

Como decíamos, los honeytokens se pueden utilizar para resolver y simplificar este problema.

Un registro médico falso llamado "John F. Kennedy" se crea y se carga en la base de datos. Esta historia clínica no tiene valor real porque no hay ningún paciente real con ese nombre. En cambio, el registro es un honeytoken, una entidad que por definición no tendrá uso autorizado. Si un empleado está buscando los datos de algún paciente interesante, este registro definitivamente se destacará. Si el empleado intenta acceder a este registro, entonces lo más probable es que se tenga a un empleado que estaría violando la privacidad del paciente. Es tan simple como eso, no hay algoritmos complejos, ni firmas que actualizar, ni reglas que configurar; alcanza con cargar la información, monitorear, y si alguien accede a este registro lo más probable es que se esté frente a una violación de una política de uso del sistema.

Otro ejemplo es el uso de números de tarjetas de crédito falsos como honeytokens. Entonces, insertando honeytokens –en este caso números de tarjetas de crédito falsos- en la base de datos se simplifica el problema. Si dichos números están siendo accedidos y si se está monitorizando la base de datos, se sabrá que alguien está violando la seguridad del sistema. Esto se logra mediante un alerta del propio motor de la base, por ejemplo, o mediante el uso de una firma para un IDS que esté “viendo” la red.

Este concepto puede extenderse más allá de bases de datos, y se puede aplicar en servidores de archivos, servidores web, o de correo electrónico, en los cuales se puede hacer uso de honeytokens. Entonces, cualquier “recurso” como definía Spitzner, que contenga datos también puede tener otros datos falsos agregados, los cuales se convertirán en nuestros honeytokens. En el caso de servidores de archivos se pueden tener archivos falsos tales como documentos de texto, hojas de cálculo o archivos PDF, por ejemplo. Estos archivos pueden tener nombres únicos, etiquetas únicas o

metadatos únicos, y nuevamente, mediante el uso de IDS con firmas personalizadas que monitoreen las redes en busca de estos registros únicos se podrá saber cuándo alguien los esté accediendo remotamente. [SNEA11]

IV.2.6 – Antimalware

IV.2.6.1 – Introducción

A la hora de hablar de código malicioso, la primera denominación que viene a la mente es la de virus. Si bien es cierto que los virus continúan siendo de los especímenes más conocidos de código malicioso, la "fauna" actual es más diversa por lo que se opta incluir a todo el código malicioso en una categoría más amplia: el malware.

De la misma forma que la contramedida primaria para la protección contra un virus es el antivirus, hoy debemos considerar ante un espectro más amplio de amenazas, herramientas que permitan protegernos ante las mismos. Estas herramientas se denominan antimalware e incluyen los antivirus clásicos como uno de sus componentes, tal como el virus es un tipo particular de malware.

Malware (del inglés "**mal**icious soft**ware**"), también llamado "badware", código maligno o malicioso, software malicioso o software malintencionado, es un tipo de software o piezas de código que tiene como objetivo infiltrarse o dañar una computadora sin el consentimiento de su propietario. Dicha infiltración puede deberse a la búsqueda de producir simplemente algún tipo de daño, pérdida de información o mal funcionamiento, o a la búsqueda de algún beneficio económico o robo de información.

El software es considerado malware basándose en los efectos que cause en un computador, pensados por el autor a la hora de crearlo. El término malware incluye, según indicáramos, a los virus, así como también a gusanos, troyanos, la mayoría de los rootkits, bots, spyware, adware intrusivo, crimeware y otros softwares maliciosos e indeseables. [SNEA11]

IV.2.6.2 – Propósitos del malware

El software diseñado para causar daños o pérdida de datos suele estar relacionado con actos de vandalismo. Muchos virus son diseñados para destruir archivos en discos duros o para corromper el sistema de archivos escribiendo datos inválidos. Algunos gusanos son diseñados para vandalizar páginas web dejando escrito el alias del autor o del grupo, por todos los sitios por donde pasan. Estos gusanos pueden parecer el equivalente en línea al graffiti.

Sin embargo, debido al aumento de usuarios de Internet, el software malicioso ha llegado a ser diseñado para sacar beneficio de él, ya sea legal o ilegalmente. Desde 2003 la mayoría de los virus y gusanos han sido diseñados para tomar control de computadoras para su explotación en el mercado negro. Estas computadoras infectadas mediante bots ("computadoras zombie") son usadas para el envío masivo de SPAM por email, para alojar datos ilegales como pornografía infantil (PC World, "Zombies PCs: Silent, Growing Threat", Jul 9, 2004.[ZPSGT04]), o para unirse en ataques DDoS como forma de extorsión, entre otras cosas.

Existen más tipos de malware producido con ánimo de lucro, por ejemplo el spyware, el adware intrusivo y los hijacker que tratan de mostrar publicidad no deseada o redireccionar visitas hacia publicidad para beneficio del creador. Estos tipos de malware no se propagan como los virus o gusanos, y generalmente son instalados aprovechándose de vulnerabilidades o junto con software legítimo como aplicaciones P2P.

De acuerdo a Panda Security, en el día 16-03-2011, la distribución por el tipo de malware detectado es como se muestra en la Figura 8.

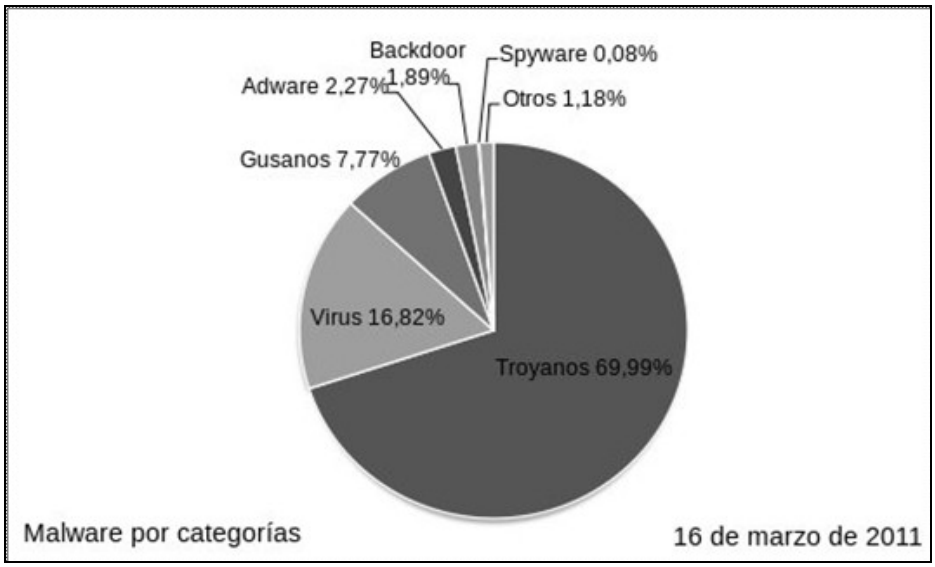


Figura 8 - Estadística de Malware el día 16-03-2011 (Panda Security).

IV.2.6.3 – Vulnerabilidades usadas por el malware

Existen varios factores que hacen a un sistema más vulnerable al malware: homogeneidad, errores de software, código sin confirmar, sobreprivilegios de usuario y sobreprivilegios de código.

Una causa de la vulnerabilidad en las redes, es la homogeneidad del software multiusuario. Por ejemplo, cuando todos los ordenadores de una red funcionan con el mismo sistema operativo, si se puede comprometer ese sistema, se podría afectar a cualquier ordenador que lo use. En particular, Microsoft Windows tiene la mayoría del mercado de los sistemas operativos de escritorio. Esto permite a los creadores de malware infectar una gran cantidad de computadoras sin tener que adaptar el software malicioso a diferentes sistemas operativos.

La mayoría del software y de los sistemas operativos contienen bugs o errores de programación que pueden ser aprovechados por el malware. Un ejemplo típico, el cual aún prevalece, son los desbordamiento de buffer (“buffer overflow”), en los cuales la estructura diseñada para almacenar datos en un área determinada de la memoria permite que sea ocupada por más datos de los que le caben, sobreescribiendo otras partes de la memoria. Esto puede ser utilizado por el malware para forzar al sistema a ejecutar su código malicioso.

En algunos sistemas, los usuarios no administradores tienen sobreprivilegios por diseño, en el sentido de que se les permite modificar las estructuras internas del sistema, porque se les ha concedido privilegios inadecuados de administrador o equivalente. Esta es una decisión de la configuración por defecto: en los sistemas de Microsoft Windows la configuración por defecto es dotar al usuario de privilegios elevados. Esta situación es debida a decisiones tomadas por Microsoft para priorizar la compatibilidad con viejos sistemas sobre la seguridad y porque las aplicaciones típicas fueron desarrolladas sin tener en cuenta a los usuarios no privilegiados. Como los exploits para escalar privilegios han aumentado, esta prioridad ha cambiado desde el lanzamiento de Microsoft Windows Vista. Como resultado, muchas aplicaciones existentes que requieren excesos de privilegios pueden tener problemas de compatibilidad con Windows Vista. Sin embargo, el Control de cuentas de usuario (UAC, "User Account Control" en inglés) de Windows Vista, intenta solucionar los problemas que tienen las aplicaciones no diseñadas para usuarios no privilegiados a través de la virtualización, actuando como apoyo para resolver el problema del acceso privilegiado inherente a las aplicaciones heredadas[2].

IV.2.6.4 – Programas antimalware

Como los ataques con malware son cada vez más frecuentes, el interés ha empezado a cambiar de protección frente a virus y spyware, a protección frente al malware, y los programas han sido específicamente desarrollados para combatirlos.

Los programas anti-malware pueden combatir el malware de dos formas:

1. Proporcionando protección en tiempo real ("real-time protection") contra la instalación de malware en una computadora. El software anti-malware escanea todos los datos procedentes de la red en busca de malware y bloquea todo lo que suponga una amenaza.
2. Detectando y eliminando malware que ya ha sido instalado en una computadora. Este tipo de protección frente al malware es normalmente mucho más fácil de usar. Este tipo de programas anti-malware escanean el contenido del registro de Windows, los archivos del sistema operativo, la memoria y los programas instalados en la computadora. Al terminar el escaneo muestran al usuario una lista con todas las amenazas encontradas y permiten escoger cuales eliminar.

La protección en tiempo real funciona idénticamente a la protección de los antivirus: el software escanea los archivos tanto al ser descargados de Internet o como al ser accedidos desde un medio removible y bloquea la actividad de los componentes identificados como malware. En algunos casos, también pueden interceptar intentos de ejecutarse automáticamente al arrancar el sistema o modificaciones en el navegador web. Debido a que muchas veces el malware es instalado como resultado de exploits para un navegador web o errores del usuario, usar un software de seguridad para proteger el navegador web puede ser una ayuda efectiva para restringir los daños que el malware puede causar.

[2] Igualmente, sin la correcta configuración de privilegios por parte del administrador, UAC no pasa de ser una simple innovación.

IV.2.6.5 – Gestión centralizada

Como ya indicáramos previamente, muchos de los productos que antiguamente ofrecían protección contra virus (antivirus), han evolucionado para brindar una protección más completa que permita el reducir la probabilidad de infección. Si bien esta modalidad es la operante en la mayoría de las versiones comerciales, muchas veces las mismas soluciones comerciales son ofrecidas con funcionalidades más reducidas en modalidad "free" orientada para usuarios domésticos.

Aquí es que surge la necesidad de plantearse: ¿son las mismas necesidades que tiene un usuario final con respecto a su herramienta antimalware, las requeridas por una organización?

En el caso del usuario final, la interacción del antimalware con el mismo se realiza en forma interactiva, desplegando avisos hacia el mismo de forma de notificarlo de alguna acción realizada o de la necesidad de una resolución con respecto a tomar o no alguna acción.

En el caso del usuario corporativo, la base es tener en cuenta que ahora el cliente final deja de ser el usuario que utiliza el recurso informático, aunque también se brinda un servicio al mismo. Nuestro cliente es la organización, a la cual nuestra herramienta de antimalware busca proteger.

Así es que muchos de los productos diferencian dos líneas de producto: las orientadas a un usuario final aislado, y las orientadas a un usuario integrado dentro de una organización. Para este último caso, generalmente se entrega la herramienta con una consola desde donde realizar la gestión de todos los antimalware instalados dentro de los recursos informáticos bajo la órbita administrativa de la organización. Estas consolas permiten desplegar en forma uniforme, las configuraciones de la herramienta ajustadas a las necesidades de la organización, como también centralizar todos los logs que cada una de las instancias ejecutadas en cada equipo pudiera generar. De esta manera se alcanza una visibilidad mayor del estado del malware dentro de la misma y permite tomar decisiones más efectivas contra el mismo.

En el contexto de buscar la construcción de sensores, sería deseable que el reporte de eventos se realizara desde la consola central, a fin de optimizar los recursos de la plataforma responsable de la centralización de la red de sensores.

IV.3 – Gestión y Análisis de los eventos generados por los sensores

IV.3.1 – Introducción

A fin de establecer un marco de referencia, presentaremos en los siguientes párrafos la descripción de un SIEM genérico, sus módulos y funciones generales. [SNEA11]

El concepto de SIEM (Security Information and Event Manager, Gestor de Seguridad de la Información y Eventos) es relativamente nuevo en el mundo de IT (Information Technology, Tecnología de la Información). Los primeros componentes de los sistemas SIEM comenzaron a emerger de varias formas hace unos 10 a 20 años, pero sólo recientemente es que se han integrado en forma adecuada y han encontrado su lugar dentro de las organizaciones.

IV.3.2 – ¿Qué es un SIEM?

Un SIEM es una compleja colección de tecnologías diseñadas para proveer visión y claridad en la organización del sistema de IT como un todo, beneficiando así también a analistas de seguridad y administradores de IT.[3]
Es generalmente pensado como el proveedor de los siguientes servicios de colección:

IV.3.2.1 – Gestión de logs

La gestión de logs[4] en un SIEM comienza con la configuración de los nodos en un sistema de IT, particularmente los más importantes o críticos, a fin de enviar los eventos relevantes de sistemas y/o aplicaciones a una base de datos centralizada manejada por el SIEM. La base de datos del SIEM primero analiza gramaticalmente[5] y normaliza los datos enviados por numerosos y heterogéneos tipos de fuentes. Luego, el SIEM típicamente provee servicios de almacenamiento de logs, de organización, de recuperación y archivo, a fin de satisfacer los requerimientos en materia de gestión de logs que una organización pueda llegar a tener. Esos servicios son habitualmente requeridos a fin de cumplir con normas regulatorias de IT. Este suministro de datos dentro del componente de gestión de logs del SIEM, presta en sí mismo el uso como herramienta de análisis casi en tiempo real y el "data mining" del estado de salud y seguridad de todos los sistemas de IT que alimentan al SIEM. Cuantos más nodos alimenten el SIEM, más completa y exacta será la visión de IT como un todo.

[3] SIEM son también conocidos como SIM (Security Information Manager, Gestor de Seguridad de la Información), SEM (Security Event Manager, Gestor de Eventos de Seguridad), y SEIM (Security Event and Information Manager, Gestor de Eventos y Seguridad de la Información).

[4] En literatura de origen español, se refiere a "gestión de bitácora". Preferimos mantener el término log, por ser de uso más habitual en nuestro entorno.

[5] "Parse" es el término utilizado en habla inglesa.

IV.3.2.2 – Cumplimiento de regulaciones de IT

Ahora que todos los eventos desde sistemas críticos y relevantes están siendo recibidos, es posible la construcción de filtros o reglas con "timers" para auditar y verificar el cumplimiento de los requerimientos impuestos por la organización. Las reglas de chequeo contra los logs que alimentan el sistema incluyen diversos controles. Como ser: monitoreo de cambios de las contraseñas, identificación del sistema operativo o aplicaciones en los cuales falla un "patch" o actualización, frecuencia con la que se ejecuta el "scanning" de un antimalware, entre tantos otros. Mientras que es posible construir nuestra propia colección de filtros o reglas para verificar el cumplimiento según los requerimientos, la mayoría de los fabricantes de SIEM incluyen un conjunto de reglas predefinidas a fin de cubrir las diferentes leyes y regulaciones que apliquen al dominio de la organización. A veces, esos paquetes son incluidos como "add-on" desde los propios vendedores a través de socios del mismo que los ofrecen mediante un pago adicional.

IV.3.2.3 – Correlación de eventos

Si los servicios brindados por un SIEM sólo se limitaran a los expuestos anteriormente, la plataforma cumplirá con la uniformización del sistema de logs de la organización; y carecería de valor si no fuera capaz de, en función de toda la información que posee, realizar el descubrimiento de la "aguja en el pajar".

Los SIEM disponen de servicios de correlación de eventos, dotándolos de un valor agregado que no podría ser posible si los servicios detallados anteriormente no se encontraran. Desde este punto de vista, los servicios de correlación son los que brindan la inteligencia de alto nivel a la ecuación permitiendo observar los eventos en su conjunto, analizar y entender qué está ocurriendo. La opción de reaccionar o no dependerá del resultado de este análisis.

Muchos SIEM generan ante la correlación de diferentes eventos y sesiones, los denominados incidentes. Estos son la forma tangible de la inteligencia detrás de la correlación.

Muchas veces, las reglas preconfiguradas en los SIEM disparan alarmas ante un conjunto de eventos que, en el contexto de la organización donde se encuentren operando, son normales. Cuando se realiza el ajuste del SIEM a estas circunstancias, se está haciendo un entrenamiento del mismo. Muchas veces se dice que el SIEM aprende las "reglas del negocio" del entorno donde ha sido desplegado.

IV.3.2.4 – Respuesta activa

Ahora que poseemos todos los sistemas de interés alimentando el SIEM, que tenemos reglas y filtros definidos en el mismo, y la correlación está activa, surge la pregunta: ¿queremos tomar acciones a fin de verificar todos los eventos recibidos y realizar una respuesta manual al incidente, o queremos que el SIEM responda automáticamente en función del tipo de eventos correlacionados?

IV.3.2.5 – Seguridad del punto final

La mayoría de los SIEM pueden monitorear la seguridad de los puntos finales, a fin de validar de forma centralizada el "estado de salud" de un sistema. Muchos SIEM pueden monitorear si el firewall en un PC está siendo ejecutado, identificar cuándo fue la última actualización del antivirus, o cuándo el nodo ha sido infectado con un "spyware". Algunos SIEM, inclusive, pueden realizar ajustes y mejoras en la seguridad del nodo en forma remota, tales como configurar el firewall y actualizar las soluciones antimalware presentes en el nodo. Más aún, algunos SIEM pueden instalar una ACL a demanda en un nodo puntual. [SNEA11]

IV.3.3 – Anatomía de un SIEM

Un SIEM puede ser comparado con una compleja máquina en la cual existen varias partes en movimiento, donde cada una de ellas realiza una función específica, y las que necesitan trabajar conjuntamente y en forma apropiada o el sistema entero fallará. Existen variaciones en lo que se entiende como un SIEM, algunas con la adición de partes específicas; pero en una forma más simple un SIEM puede dividirse en seis piezas o procesos separados. Esas piezas individuales son: el dispositivo fuente, el colector de logs, el analizador gramatical (parser) o normalizador de logs, el motor de reglas o motor de correlación, el almacenador de logs, y el monitor y recuperador de eventos. Cada una de las partes puede trabajar en forma independiente de las demás, pero si todas no trabajan juntas el SIEM como un todo no operará apropiadamente.

IV.3.3.1 – Dispositivo fuente

La mayoría de la gente no es consciente del volumen de logs que son generados normalmente en su actividad diaria. La sola acción de abrir un navegador y chequear su correo electrónico, genera una multitud de registros en diferentes dispositivos: logs en la computadora del usuario, además de varios en los switches, routers y firewalls que el usuario tiene que atravesar para alcanzar el sitio buscado, a los que se le suma los que el propio sitio web genera por el correspondiente acceso del usuario a su casilla de correo. Dependiendo de lo que se esté buscando, alguna de esta información será de utilidad.

Como indicáramos previamente, el primer componente de un SIEM es el dispositivo fuente. Definimos al mismo como una entidad (dispositivo, aplicación, proceso, etc), de la cual deseamos recibir información para ser procesada por el SIEM. Si bien los dispositivos fuentes no son parte en sí mismos del SIEM, son una pieza de vital importancia en todo el proceso. Son ellos los encargados de, como vimos, alimentar de información al SIEM por lo que en el caso de no contar con ellos, el SIEM carecería de utilidad.

Algunas veces, los dispositivos fuente pueden poseer información que no está disponible para ser enviada hacia el exterior, en nuestro caso hacia el SIEM. En estos casos, son de utilidad aquellas piezas de software agrupadas genéricamente bajo el

nombre de agentes, los cuales tienen como función recolectar la información contenida dentro del dispositivo y, previo procesamiento local o no, enviarla a nuestro SIEM.

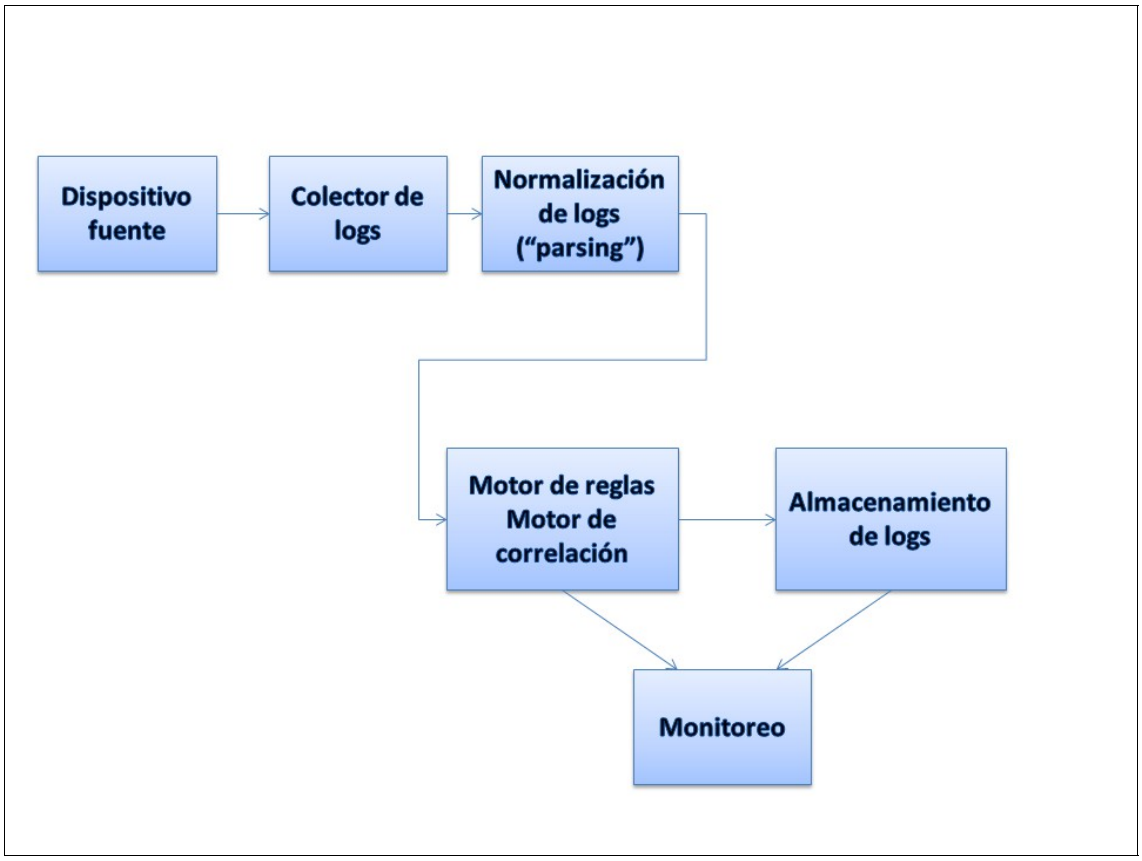


Figura 9 - Diagrama esquemático de un SIEM.

IV.3.3.2 – Colector de logs

El siguiente paso en el flujo de nuestro SIEM genérico es como son enviados todos los diferentes logs generados por los dispositivos fuente, cada uno en su forma nativa, hacia el SIEM. El mecanismo por el cual el SIEM obtiene dichos logs, varía en función del SIEM que se esté utilizando, pero en su forma más básica, puede ser diferenciado en dos métodos fundamentales: el dispositivo fuente envía sus logs al SIEM ("push"), o el SIEM recupera los logs del dispositivo ("pull"). Cada unos de estos métodos tienen sus pro y contras, dependiendo del entorno, pero son efectivos en el hecho de que el SIEM obtenga la información de los dispositivos fuente.

IV.3.3.3 – Normalización de logs ("parsing")

En este punto, los logs de los diferentes dispositivos fuente se encuentran aún en su formato nativo por lo que sólo hemos desarrollado, hasta el momento, un repositorio único para ellos. A fin de que los mismos puedan ser utilizados por el SIEM es necesario procesar los mismos y convertirlos en un formato único. El acto de

modificar los distintos formatos a uno sólo se denomina normalización. Para cada tipo de SIEM este formato único seguramente diferirá, dado que se espera que la consistencia del mismo sea sólo necesaria entre productos del mismo fabricante o proyecto de desarrollo. Este es otro punto a considerar en el caso de desear que la interacción entre el SIEM de la organización y otros interesados, ej. un CERT nacional. Aquí también deberían considerarse temas de confidencialidad o procesos de "sanitización" de la información compartida con terceros.

En general, el formato de los logs dependerá del equipo encargado del mismo. A fin de lograr que sean convertidos en el formato único soportado por el SIEM, es necesario poseer un conocimiento detallado de los mismos de manera de lograr su conversión. En general, los SIEM proporcionan varios "conversores" para los dispositivos fuente más habituales, aunque es necesario recordar que existen situaciones en las cuales el propio desarrollador o administrador del SIEM deberá programar esta "conversión". Nuevamente, esa capacidad es un diferenciador entre distintas alternativas de SIEM.

La normalización de los eventos no sólo hace más sencilla la lectura de los logs, también permiten generar reglas como las de correlación, en una forma estándar.

IV.3.3.4 – Motor de reglas/Motor de correlación

El motor de reglas se expande por encima de la normalización de eventos de los diversos dispositivos fuente, de manera que puedan dispararse alertas dentro del SIEM debido a condiciones específicas dentro de los logs. El método de escribir reglas en el SIEM usualmente comienza en forma moderadamente simple, pero puede tornarse extremadamente complejo. Típicamente, son reglas como expresiones de lógica booleana, donde se examina ciertas campos de información en búsqueda de patrones predefinidos. El hecho de que previamente a esta búsqueda se realice la normalización de los diferentes logs, permite que dicha búsqueda sea uniforme y fácilmente trasladable a todos los eventos que son procesados por el SIEM.

Ahora bien, este motor de correlación no es simplemente las reglas indicadas anteriormente. Si fuera tan simple como eso, el desarrollo de dichos motores de correlación, su flexibilidad, efectividad y eficiencia, no serían propiedades que diferencian unos de otros, sin importar si los mismos se basan en soluciones comerciales u Open Source Software (OSS). A la hora de realizar la acción de correlación, además de las expresiones booleanas, es necesario darle contextualización a las mismas. En este sentido, lo que se busca es que se logre una comprensión más acabada de cómo se originan los eventos, cuál dispositivo, sistema operativo o aplicación los origina, la ubicación dentro de la infraestructura informática, y la existencia o no de otros dispositivos fuente que pudieran estar relacionados.

La idea detrás de un motor de correlación es permitir que la inteligencia humana sea llamada a trabajar sólo después de que un proceso consolide, de manera inteligente, a la mayor cantidad posible de los eventos recibidos. El especialista sólo deberá actuar cuando la discrecionalidad de la inteligencia del algoritmo no sea suficiente para poder determinar si se está o no frente a un incidente.

IV.3.3.5 – Almacenamiento de logs

A fin de poder trabajar con grandes volúmenes de información, los cuales convencionalmente son la realidad involucrada de un SIEM, es necesario desarrollar una estrategia de persistencia ya sea por propósitos históricos, o por mandato de cumplimiento reglamentario. La estrategia de persistencia se puede basar en tres paradigmas: una base de datos, archivos de texto plano o archivos binarios.

La utilización de una base de datos, es el método más habitual para el almacenamiento de logs en las soluciones SIEM. Los motores de base de datos habitualmente utilizados son soluciones de escala "enterprise. La primera ventaja en la utilización de una base de datos es que existe un método para, en forma estructurada, almacenar la información. En nuestro caso: los logs o eventos recibidos y procesados por el SIEM, contando con un entorno para la manipulación de dicha información a través de un lenguaje como el SQL. Consideramos que, casi invariablemente, los requerimientos computacionales asociados al almacenamiento y procesamiento de los logs son exigentes, por lo que ha de considerarse cuidadosamente el motor y sus especificaciones a fin de que la solución de SIEM implementada no falle en este punto. Normalmente el acceso a la información en este formato es bueno, aunque a veces se requiera de la optimización del mismo a fin de ser utilizada en forma más adecuada por el SIEM.

Respecto a las bases en texto plano, estas son utilizadas en la mayoría de los servidores para el almacenamiento de eventos. Además, un servidor para el registro de eventos más que implementar la persistencia de los mismos, se utiliza para su centralización o consolidación, preservando así la integridad de estos registros.

Como contrapartida, la primera desventaja es un problema de capacidad. El mismo está dada por el significativo aumento del requerimiento del espacio de almacenamiento con respecto a una solución que utilice una base de datos. También en este aspecto, la cantidad o volumen de los archivos puede tornarse muy grande. En este caso nos encontraremos con las limitantes propias del sistema de archivos que use el sistema operativo, o que la información no se encuentre con alguna estructura jerárquica. Esto último conspira también contra la velocidad de procesamiento ante búsquedas de alguna información particular.

Finalmente, los archivos en formato binario son utilizados en algunas ocasiones como un formato propietario, siendo sólo accesibles desde el SIEM para el cual han sido diseñados.

IV.3.3.6 – Monitoreo

El proceso final de nuestro SIEM, es el método en el cual se interactúa con los logs almacenados por el SIEM. Una vez que todos los logs en el SIEM y los eventos han sido procesados, es necesario el uso de la información útil; de lo contrario los logs únicamente existirían en el SIEM sólo con el propósito de almacenamiento. El SIEM tendrá alguna consola a fin de proporcionar una forma de interacción con el mismo, la cual generalmente es alguna aplicación Web, aunque también podría ser algún tipo de cliente particular que pueda ejecutarse en una estación de trabajo, para alguna modalidad cliente/servidor. Sea cual fuere la interfaz utilizada, la misma permitirá la interacción con la información almacenada por el SIEM y con los resultados del motor de correlación. Dependiendo de la implementación en particular, la misma consola puede también permitir la administración del propio SIEM.

Otra de la funciones de la consola del SIEM es el brindar un entorno que permitirá la generación de nuevas configuraciones para el motor de reglas o el motor de correlación. A su vez se permitiría ajustar los métodos por los cuales se extrae información desde los dispositivos fuente, principalmente en el caso de que dicha información se obtenga mediante algún proceso de "pooling" programado. También ofrecerá trabajar con la información almacenada por el SIEM, permitiendo la generación de reportes o la exportación de la misma con fines forenses o para ser compartida con otro centro de seguridad. [SNEA11]

IV.4 – Protocolos de intercambio de información

A continuación se describirán algunos de los protocolos más utilizados para el intercambio de información entre los dispositivos de una red informática. Si bien no se verá el funcionamiento de los mismos en forma detallada, si se presentarán sus principales problemas en seguridad y la forma de prevenirlos.

IV.4.1 – Introducción

Cada plataforma cuenta, o debería contar, con al menos un sistema para el envío de sus mensajes de registro. Por ejemplo en el caso de sistemas Unix contamos con Syslogd y Syslog-ng, entre otros; en el caso de Microsoft Windows tenemos WinSyslog y Snare, por citar algunos. El cometido de todos es básicamente el mismo, brindar un protocolo para el envío y transporte de mensajes a través de la red con el objetivo de centralizar los servicios de log. Como veremos más adelante, esto permite contar con un registro de los mensajes enviados por los diferentes servicios y dispositivos a fin de tenerlos disponibles para un análisis posterior.

En general un mensaje de registro suele tener información sobre la seguridad del sistema, aunque puede contener cualquier información del mismo. Lo que sí es común para todos los mensajes es la fecha y hora del envío.

Los datos que en general se estilan registrar son del tipo:

- Un intento de acceso con contraseña equivocada
- Un acceso correcto al sistema
- Anomalías como ser variaciones en el funcionamiento normal del sistema
- Alertas por ocurrencia de alguna condición especial
- Información sobre las actividades del sistema operativo y aplicaciones
- Errores de hardware o de software

IV.4.2 – Principios de funcionamiento

La filosofía de funcionamiento de estos sistemas es básicamente la misma. Existe un host el cual actuará como servidor, y uno o varios dispositivos (clientes) que serán quienes reporten sus mensajes al primero. El servidor será el encargado de recolectar los mensajes enviados por sus clientes y archivarlos según su procedencia o severidad.

IV.4.3 – Syslog

El protocolo Syslog en su versión original transmite sus mensajes vía UDP, por el puerto 514, en formato de texto plano. Algunas implementaciones más recientes, como Syslog-ng, permiten usar TCP, y ofrecen la posibilidad de cifrar los datos mediante el uso de SSL/TLS.

En la arquitectura Syslog podemos identificar tres tipos de actores:

- Fuente: Son aquellos que generan los mensajes. (Cualquier dispositivo)
- Relay: Su función es la de reenviar los mensajes desde dispositivos fuente o desde otros relays hacia los collectors
- Collector: Son los encargados de colectar y almacenar los mensajes.

Algunas de las formas en que es posible interconectar estos elementos se muestra en la Figura 10 [Ger08]

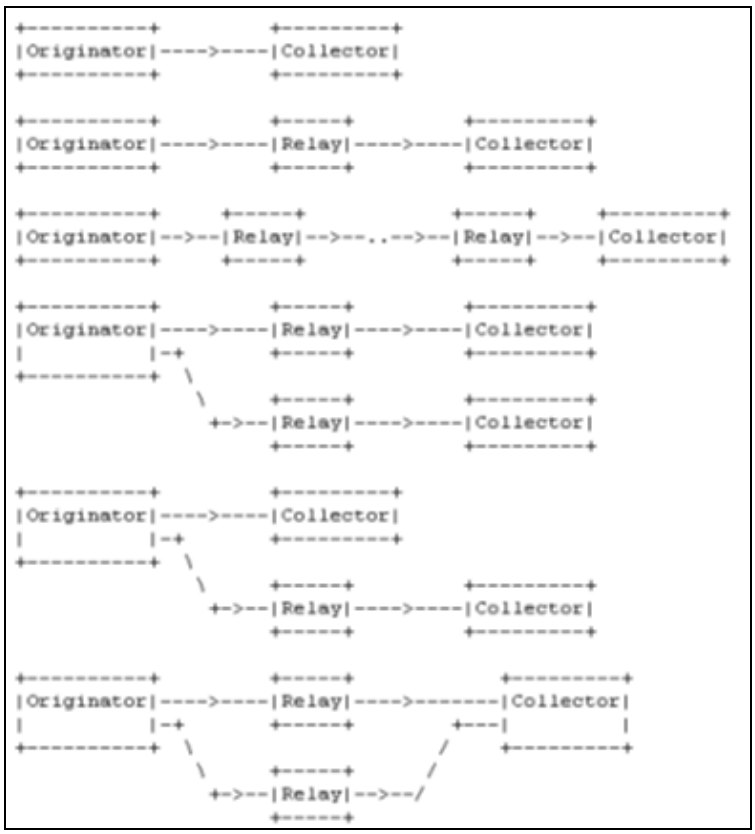


Figura 10 - Interconexiones posibles de agentes Syslog [Ger08].

IV.4.3.1 – Análisis de seguridad de Syslog

A continuación describiremos someramente las carencias y algunas recomendaciones para conseguir mediante Syslog un entorno y registro de eventos seguro y confiable. En estos pasos incluiremos: la creación de un repositorio seguro para los log de eventos, la utilización de un mecanismo confiable para el transporte de los datos y el tener la capacidad de verificar y mantener la integridad de los sistemas de envío y recepción de los logs. [Ken03]

IV.4.3.1.1 – Confidencialidad

En el caso de Linux, los datos encaminados por el demonio syslogd a un proceso syslogd remoto son enviados en texto claro, y por tanto son fácilmente visibles utilizando herramientas de análisis de red. Esto permite a un intruso ver los eventos informados por routers, switches o herramientas tales como Snort.

Una opción para mejorar este aspecto es la utilización de transmisiones cifradas, por ejemplo mediante el uso de SSH. Esto evita la posibilidad de recibir ataques mediante "sniffers". Otras alternativas son el uso de aplicaciones tales como STunnel para envío de los datos sobre el protocolo SSL, o el uso de encriptación mediante IPSec.

También es recomendable cifrar los archivos que contengan los datos de log generados por Syslog y que residan en host y/o en el servidor de registros de logs. Esto permite mantener la confidencialidad, incluso si se obtiene acceso no autorizado a estos archivos. [Ken03]

IV.4.3.1.2 – Integridad

Los datos almacenados por el proceso de Syslog quedan en formato de texto sin cifrar o también llamado texto claro. Incluso es fácil dar con éstos si se guardan en las ubicaciones por defecto. Por ejemplo los datos de syslogd en un sistema UNIX son guardados en el directorio /var/log.

Adicionalmente, cuando se utiliza un servidor de registro central con la versión original de Syslog, el administrador no será capaz de determinar si los datos del registro han sido modificados, ya sea cuando son generados o mientras atraviesan la red hasta el host de registro. Esto se debe a que Syslog utiliza paquetes UDP para el envío de sus datos lo cual no proporciona ninguna garantía de entrega. Esto es un resultado directo del protocolo elegido en la implementación original de Syslog para el transporte de sus datos.

Así, una vez más se debe considerar la sustitución del Syslog por una de sus versiones más recientes. El Nsyslog, Syslog-ng y el SCSC Secure Syslog abordan este tema mediante la sustitución de los paquetes UDP por paquetes TCP. En este sentido Modular Syslog por ejemplo utiliza un mecanismo de firma electrónica (usando una función hash). Esto permite a los administradores verificar que los datos no han sido modificados durante su traslado o almacenamiento. [Ken03]

En los casos en que no sea posible ninguna de las alternativas anteriores, podría ser utilizado IPsec para el transporte de la mensajería Syslog.

IV.4.3.1.3 – Autenticidad

Como ya comentamos, el protocolo Syslog no proporciona ninguna herramienta para permitir la autenticación de los datos recibidos. Esto brinda la posibilidad de, mediante la utilización de herramientas como "netcat", crear mensajes falsos con direcciones de origen inválidas (spoofing). Claramente esto también compromete la integridad del registro de eventos.

En este aspecto, los certificados de autenticación de un host con Secure Syslog son suficientes para cumplir con este requisito durante el pasaje de los datos por la red. Otra opción es utilizar la autenticación que brinda el uso de SSH.

Desafortunadamente, si un sistema o dispositivo no es compatible con Secure Syslog o SSH, será difícil determinar la autenticidad de los datos recibidos por este. [Ken03]

IV.4.3.2 – Intentos de mejora para Syslog

IV.4.3.2.1 – Modular Syslog

El objetivo de este remplazo de Syslog es el de proveer al administrador una versión con la cual le sea posible verificar la integridad de los logs de eventos que fueron archivados. Para esto se adicionaron en el demonio syslogd módulos de entrada y salida. Esto le permite al administrador determinar si los datos archivados fueron modificados (verificación de integridad) durante la transferencia a través de la red o en el almacenamiento.

Modular Syslog también cuenta con otros módulos los cuales proveen la habilidad de almacenamiento en bases de datos del tipo MySQL o POSTGREsql.

Finalmente Modular Syslog brinda un módulo con el cual el administrador puede aplicar expresiones regulares para filtrar las bases de datos de logs. [Ken03]

IV.4.3.2.2 – Nsyslog

Nsyslog soluciona el problema de integridad de datos presente en las versiones actuales de syslog sustituyendo el uso de paquetes UDP para la transferencia de los datos por el uso de TCP.

Como sabemos, TCP establece la entrega fiable de los datos ya que requiere que todos los paquetes que se envían a la red sean reconocidos.

La confidencialidad de los datos durante la transferencia por la red también es posible con Nsyslog, ya que permite el encriptado de paquetes usando el protocolo SSL (Secure Socket Layer). [Ken03]

Finalmente cabe destacar que Nsyslog es compatible con una amplia gama de aplicaciones que el actual Syslog también soporta, e incluso puede ser configurado para escuchar los paquetes UDP en el puerto 514.

IV.4.3.2.3 – Syslog-ng

El objetivo de Syslog-ng es crear un nuevo set de herramientas de Syslog las cuales extienden el protocolo actual para satisfacer las necesidades que el entorno de una red actual plantea. El Syslog-ng es un sistema de logueo flexible y altamente escalable, ideal para crear soluciones de logueo centralizadas. [Ken03]

Algunas de las mejoras son:

- Transferencia de logs de manera fiable desde el host al servidor o servidores remotos, enviados mediante el protocolo TCP.
- Logueo seguro mediante SSL/TLS: Utilizando TLS no solo para encriptar la comunicación sino paran permitir la autenticación mutua entre el cliente y el servidor.
- Buffering de mensajes en disco en caso de que la conexión de red no se encuentra disponible durante el envío de mensajes.
- Acceso directo a base de datos. Esto que permite realizar búsquedas y consultas de los mensajes e interactuar con bases de tipo MySQL, Oracle, PostgreSQL y SQLite.
- Syslog-ng permite recoger registros de forma masiva en entornos heterogéneos utilizando diferentes sistemas operativos y plataformas de hardware, incluyendo Linux, Unix, BSD, Sun Solaris, HP-UX y AIX. Además cuenta con un agente para la transferencia de registros de Microsoft Windows hacia un servidor Syslog-ng.
- Puede ordenar los mensajes de registro basados en su contenido con diversos parámetros, como host de origen, aplicación y prioridad. Permite además filtrados complejos utilizando expresiones regulares y operadores booleanos, ofreciendo la posibilidad de realizar envíos de mensajes importantes a destinos específicos.
- Soporte de IPv4 e Ipv6, estando preparado para trabajar en ambos entornos de red. [Sar08]

IV.4.3.3 – Syslog para Windows

En sistemas operativos de Microsoft, existen algunas alternativas como WinSysLog, Syslog Watcher, SyslogIT, etc.

IV.4.3.4 – Resumen de funcionalidades y ventajas de Syslog

- Provee funcionalidades de logueo a aplicaciones y dispositivos.
- Proporciona a los administradores controles sobre los logs.
- Permite clasificar y priorizar mensajes.
- Los mensajes pueden ser enviados a múltiples destinos:
  - archivos de log;
  - otros hosts para centralizar logs.
- La centralización de los mensajes permite a su vez:
  - un control centralizado de logs;
  - monitoreo centralizado con un único punto de acceso;
  - facilidades para data minning;
  - los mensajes de distintos sistemas pueden ser ordenados en un único sistema;
  - mensajes relacionados desde distintos sistemas pueden ser correlacionados y consultados en un solo lugar;
  - acceso a un único host de logueo en vez de a múltiples sistemas;

- agregar información para facilitar la búsqueda;
- patrones sospechosos pueden ser reconocidos más fácilmente.

35 17 Disponible nativamente en múltiples plataformas que pueden operar como sensores.

IV.4.4 – SNMP

IV.4.4.1 – Introducción

SNMP (Simple Network Management Protocol) es un protocolo utilizado para intercambiar información entre los dispositivos de una red y un gestor con el fin básico de monitorizar y gestionar a los primeros.

En esta sección nos centraremos en analizar este protocolo en lo que refiere a la seguridad que presentan sus distintas versiones.

IV.4.4.2 – Arquitectura y principio de funcionamiento de SNMP

El modelo SNMP consta de cuatro componentes:

- Nodos administrativos:
- Estaciones administradoras.
- Información de administración.
- Un protocolo de administración.

Por nodo administrativo entenderemos a cualquier dispositivo capaz de comunicar información al mundo exterior, como ser hosts, routers, impresoras, etcétera. Para ser administrado directamente por SNMP estos nodos deben ser capaces de ejecutar el agente (proceso de administración) SNMP. Para los dispositivos que no cuenten con la capacidad de ejecutar dicho agente se deberá definir un agente apoderado SNMP el cual supervisará uno o más dispositivos no SNMP, y se comunicará con la estación administradora en nombre de ellos.

Las estaciones administradoras son aquellos sistemas encargados de ejecutar el software de administración; generalmente computadoras. Las mismas corren el proceso de comunicación con los agentes. Algunas de estas estaciones cuentan con una interfaz gráfica para facilitar la visualización de la red al administrador.

La Información de administración se refiere a la información que podemos supervisar. Cada nodo administrativo mantiene una o más variables que describen su estado actual (también llamadas objetos) y almacena los estados pasados. El conjunto de todas estas variables esta descrito en una estructura de datos llamada MIB (Management Information Base). La MIB es justamente la base datos a través de la cual tendremos acceso a la información gestionable del equipo.

Finalmente, el protocolo de administración, o sea el utilizado entre la estación administradora (o gestor) y los nodos administrativos, es justamente SNMP. Este brinda el mecanismo para acceder a los objetos en la MIB a fin de poder consultarlos o incluso modificarlos, así como permitir a los nodos que envíen mensajes no solicitados

reportando su estado. Vale decir entonces que la comunicación se puede producir de dos maneras:

1. El gestor puede preguntar al agente acerca del valor de alguna variable
2. El agente puede informar al gestor acerca de algún hecho importante.

Para estos mensajes SNMP utiliza UDP como mecanismo de transporte, por lo cual no se garantiza la entrega de la información. Queda en manos del receptor de los mensajes el solicitar el reenvío de la información en caso de falla.

Respecto a la autenticación, SNMP utiliza para el intercambio de información un string community el cual actúa similar a una contraseña. Dicho string establece una relación lógica entre los dispositivos administrados y el gestor a fin de validar los miembros que la utilizan. Así, los dispositivos administrados solo responderán a los mensajes de un gestor con un string community válido[6]. La idea con esto es evitar que un gestor no autorizado pueda enviar o recibir los mensajes de intercambio. Sin embargo los string community se transmiten en texto claro, lo cual representa una conocida debilidad de SNMPv1 y SNMPv2. [Ron03] SNMPv3 busca eliminar estas limitaciones, incorporando tanto funciones de integridad como de confidencialidad.

A continuación se presentan estas distintas versiones.

IV.4.4.3 – SNMPv1/SNMPv2c y su seguridad

El hecho que SNMPv1 se utilice más para el monitoreo de los nodos que para el control de los mismos es razonable debido a sus carencias en seguridad. En este sentido entendemos como falencias de esta versión a las siguientes:

- Los string community se transmiten en texto claro. Por lo que si un atacante obtiene acceso a estos, tendrá acceso sobre los dispositivos, siendo limitado solo por los privilegios de dicha comunidad.
- El transporte mediante UDP está sujeto a la suplantación de direcciones IP de origen. Esto permite que un usuario no autorizado pueda asumir la identidad del gestor para realizar operaciones maliciosas.
- La recepción de mensajes SNMP mal formados puede provocar la denegación de servicios en los dispositivos administrados.

En resumen, las amenazas que estas vulnerabilidades representan son:

- Modificación de la información de gestión mientras ésta está en tránsito.
- Suplantación de identidad de un usuario autorizado.
- Replicación de mensajes de gestión.
- Divulgación de la información de los mensajes, como ser la comunidad.
- Envío de paquetes mal formados y sus posibles consecuencias.
- Análisis de tráfico por parte de usuarios no autorizados.

[6] Todos los miembros de una comunidad tienen el mismo privilegio de acceso, ya sea de solo lectura o de lectura-escritura

Respecto a SNMPv2c, el mismo no alcanzó a cumplir con los requisitos originales para una mejor seguridad y administración. El salto cualitativo en seguridad se observó en SNMPv3. [Ron03]

IV.4.4.4 – SNMPv3 y seguridad

SNMPv3 fue diseñado para sumar seguridad y administración con respecto a sus versiones anteriores mediante el agregado de algoritmos de autenticación y encriptado. Sin embargo desde su definición se estableció que este protocolo no estaba diseñado para prevenir que existan excesivos intercambios de mensajes entre los nodos y el gestor (que puedan provocar una denegación de servicio), ni evitar que un atacante pueda analizar el tráfico entre nodo y gestor.

El marco para la gestión de SNMPv3 sigue la misma arquitectura que la vista para las versiones 1 y 2. Esta versión incluye mejoras importantes en seguridad, manteniendo la compatibilidad con las versiones anteriores.

En esta versión, para identificar una entidad SNMP, se introduce el concepto de EngineID. Un EngineID es un identificador único configurado por un administrador en el agente SNMP de cada dispositivo a gestionar. Las versiones anteriores de SNMP se basaban en la dirección IP o en el nombre del dominio de un dispositivo gestionado, datos que pueden ser obtenidos por extraños y conducir a riesgos de seguridad. El EngineID añade otra capa de seguridad y es utilizado para crear la clave de autenticación y para identificar el origen y el destino del mensaje.

El mecanismo de seguridad de SNMPv3 está basado en dos modelos: USM (User-based Security Model) para la implementación de la autenticación y el cifrado, y VACM (View-based Access Control) para el control de acceso. Mientras que la autenticación y el cifrado se utilizan principalmente para la seguridad a nivel de mensaje, el control de acceso se utiliza para restringir el acceso a las MIB.

El proceso de autenticación asegura que el mensaje proviene de un originador válido, que los datos no se modificaron durante su transmisión y que el mensaje fue entregado de manera oportuna. La autenticación en SNMPv3 se realiza mediante una clave compartida entre la entidad emisora y la receptora. Dicho código se construye utilizando MD5 (Message Digest 5) o SHA (Secure Hash Algoritm) en base a la fecha y hora, la clave secreta, el EngineID y el contenido del mensaje.

El segundo control de autenticación esta dado por el tiempo de entrega del mensaje. Quien recibe el mensaje compara el time-stamp (indicador de tiempo) del mensaje recibido con el valor de su reloj interno. Si la diferencia de tiempos es mayor al umbral especificado se rechaza el mensaje. [Ron03]

Para brindar flexibilidad a los entornos, la configuración de seguridad en SNMPv3 admite tres niveles:

1- Sin autenticación ni cifrado: En este nivel no se tiene ninguna disposición para la seguridad ni para la privacidad. Se utiliza

generalmente en etapas de depuración de la red o durante el desarrollo de aplicaciones.

2- Con autenticación pero sin cifrado: En este nivel se requiere que las entidades SNMP se autentiquen pero no se cifra el contenido del mensaje.

3- Con autenticación y cifrado: Aquí el mensaje es cifrado y se realiza el proceso de autenticación entre las entidades.

El cifrado del mensaje SNMP se realiza mediante DES (Data Encryption Standard) de modo de evitar las escuchas y asegurar la privacidad de la información. Cuando el cifrado está habilitado se emiten dos tipos de contraseñas, una se utiliza para la autenticación y la segunda para el cifrado. La clave privada se crea utilizando el mismo algoritmo hash que en el proceso de autenticación. Al igual que en el proceso de autenticación, la entidad que envía y la que recibe utilizan la misma clave privada para cifrar y descifrar el mensaje. Esto evita que personas no autorizadas obtengan la información mientras la misma se encuentra en tránsito.

Respecto al control de acceso, encontramos en esta versión una mejora significativa respecto a las versiones anteriores. Aquí los agentes SNMP se configuran para proporcionar diferentes niveles de acceso a su MIB. Con esto, diferentes usuarios tendrán diferentes privilegios según las políticas de control definidas en los agentes SNMP. Para ello, cada agente mantendrá una tabla de control de acceso (ACL) con dichas políticas. [Ron03]

IV.4.4.5 – Buenas prácticas en seguridad al utilizar SNMPv1/v2

- Deshabilitar los servicios que no se utilizan. Esto implica deshabilitar el tráfico SNMP en los routers de borde, salvo en el caso que estemos gestionando equipos remotos en otras redes.
- Cambiar la configuración por defecto de las strings community. Se evitará así el acceso no autorizado por la simple utilización de los string “private” (lectura-escritura) y “public” (solo lectura).
- Restringir el uso de strings community con privilegios elevados. Para esto es recomendable proporcionar los niveles adecuados de autorización y acceso a cada usuario, así como también utilizar strings diferentes para las consultas y para las notificaciones. [Ron03]
- Mantener un backup de la configuración del sistema de gestión en otro punto de la red.

IV.4.5 – IDMEF

IV.4.5.1 – Introducción

El Intrusion Detection Message Exchange Format, o por sus siglas en Inglés IDMEF, desarrollado por el grupo IDWG (Intrusion Detection Working Group), es una especificación basada en XML y fue propuesto como un formato común para el

intercambio de alertas de distintos IDS, permitiendo así la interoperabilidad entre sistemas comerciales, OSS, y los sistemas de investigación.

Según [DeCu07] tiene como propósito el definir formatos de datos y procedimientos de intercambio para compartir información de interés entre los sistemas de detección de intrusos, sistemas de respuesta y los sistemas de gestión que posiblemente tengan que interactuar con ellos.

IV.4.5.2 – Descripción - Formato de mensaje

El modelo de datos IDMEF es una representación orientada a objetos de los datos de alerta enviados a las consolas de gestión por los analizadores de detección de intrusiones; y este modelo de datos está diseñado para proveer de un estándar de representación de alertas de una manera que no de lugar a ambigüedades, permitiendo así la relación entre alertas simples y complejas a ser descriptas.

Un mensaje IDMEF se organiza de la siguiente manera. La clase de nivel superior para todo mensaje IDMEF es ***IDMEFMessage***, y cada tipo de mensaje es una subclase de esta clase de nivel superior. Actualmente, hay definidos dos tipos de mensajes: ***Alerts*** y ***Heartbeats***. Dentro de cada mensaje se utilizan subclases para proporcionar la información detallada en el mensaje.

Cada vez que un analizador detecta un evento para el que ha sido configurado escuchar envía un mensaje de alerta (Alerts) a su manager; y dependiendo del analizador, un mensaje Alert puede corresponder a un único evento detectado o a múltiples eventos, presentando una lista de atributos con sus respectivos tipos.

Así, será posible identificar el analizador que creó la alerta (*Analyser*), la marca de tiempo en la que fue creada (*CreateTime*), la marca de tiempo del evento recogido por el sensor (*DetectTime*), el origen y destino del evento, etcétera. Además, cada alerta proporcionará una clasificación que definirá de forma única el motivo por el cual la alerta ha sido creada. Esta clasificación será conocida por el resto de los componentes del sistema, de forma que cualquiera de ellos será capaz de instanciarla en su base de conocimiento, haciendo posible un proceso de correlación de información a través de estas alertas.

Los mensajes Heartbeat son empleados por los analizadores para reportar su estado actual al manager.

Las relaciones entre los componentes principales del modelo de datos se muestran en la Figura 11, con sus mensajes Alert y Heartbeat, así como sus respectivas subclases:

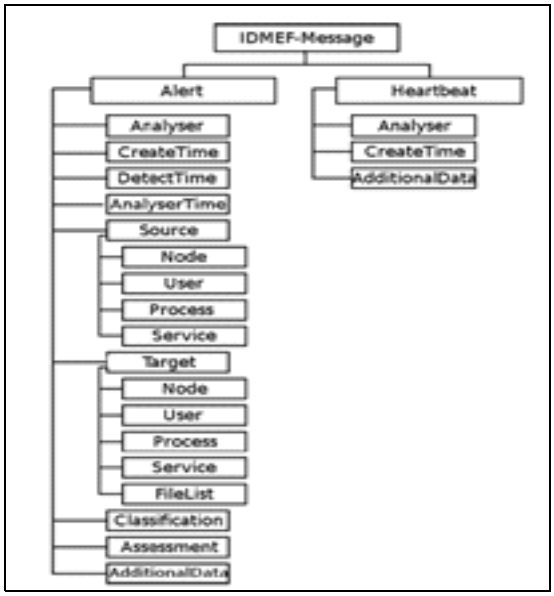


Figura 11 - Diagrama de los componentes de un mensaje IDMEF

IV.4.5.3 – Mejoras a IDMEF

Diversos autores e investigadores han propuesto mejoras a IDMEF, como [HaDeWe10].

Los fundamentos a tales mejoras radican en que el modelo de datos actual no especifica cómo una alerta debe ser clasificada o identificada. Asimismo, en un sistema distribuido y cooperativo de detección de intrusiones, el intercambio de datos de alertas aumenta la complejidad de la comunicación así como los requerimientos de ancho de banda para tal fin. Entonces, se podría dar el caso en que determinados tipos de actividad maliciosa no sea notificada o se dificulte la función de correlación en los managers, limitando sus capacidades proactivas y de respuesta.

Es de resaltar que el formato de mensaje IDMEF es independiente del protocolo de comunicación. Si bien el IDWG ha propuesto también IDXP (Intrusion Detection Exchange Protocol, RFC4767) como el protocolo de comunicación para IDMEF, el mismo se encuentra en una etapa experimental y no es un requisito obligatorio. [SNEA11]

V. Comparativa de SIEM

V.1 – Introducción

Expuestas ya las principales características de una red de sensores, sus componentes, sistemas de comunicación y las características de un SIEM genérico, pasaremos a presentar el estudio de los SIEM seleccionados. Si bien la taxonomía previamente desarrollada permite comprender y establecer una base de comparación entre diferentes productos, claramente no todos se ajustarán en forma estricta a la misma.

En esta sección nos dedicaremos al estudio en forma genérica de cuatro SIEM seleccionados, mediante los cuales es posible el despliegue de una red distribuida de sensores.

Como criterio de selección, se buscó que dos de ellas fueran de “Código Abierto“ (Open Source Software - OSS) y dos del tipo “Código Cerrado”. Estas categorías no son excluyentes, pues como es habitual en varios proyectos OSS, también poseen una versión de código propietario, al menos en alguno de sus componentes. En el caso que aplique, se presentará el producto OSS y se describirá la diferencia con su equivalente propietario.

Dado que el número productos SIEM es bastante amplio, se tomó como criterio de elección estudios realizados por el Grupo Gartner conocidos como “Magic Quadrant”, en este caso para SIEM [GMQ11]. Este estudio se realiza sólo a productos con soporte comercial, los cuales generalmente son soluciones de código propietario. En el caso de las soluciones OSS, se estudió el tiempo de trabajo de la misma, su nivel de desarrollo, su integración con otras soluciones, sus funcionalidades, y si en el presente el proyecto continúa en actividad.

En base a los criterios expuestos, las herramientas estudiadas son:

- HP ArcSight ESM (Propietario)
- Q1 Labs QRadar (Propietario)
- AlienVault OSSIM (OSS y Propietario)
- Prelude (OSS y Propietario)

V.2 – Aspectos Considerados

Para dicha elección, se considerará tanto las herramientas estudiadas, la realidad de la red a proteger, así como las posibilidades presentes y futuras de dicha elección. Esta última consideración busca que aún cuando puede elegirse una de las herramientas para el desarrollo de la solución buscada en el presente proyecto, que dicha elección no hipoteque la posibilidad a futuro del cambio a una nueva herramienta.

V.3 – HP-ArcSight ESM

V.3.1 – Introducción

ArcSight cuenta con varios productos de seguridad adaptables a una amplia gama de empresas de diferentes tamaños. En particular su producto SIEM es utilizado tanto en organizaciones corporativas como en ambientes educativos y gubernamentales; ya sea para detectar amenazas a la seguridad de sus redes como para cumplir con normativas de seguridad.

En la Figura 12 y Figura 13 se muestran los productos ArcSight que serán desarrollados en esta sección, indicando en cada uno la capa de la arquitectura ArcSight en que trabajan.

Figura 12 - Productos modulares de ArcSight [ASL10]

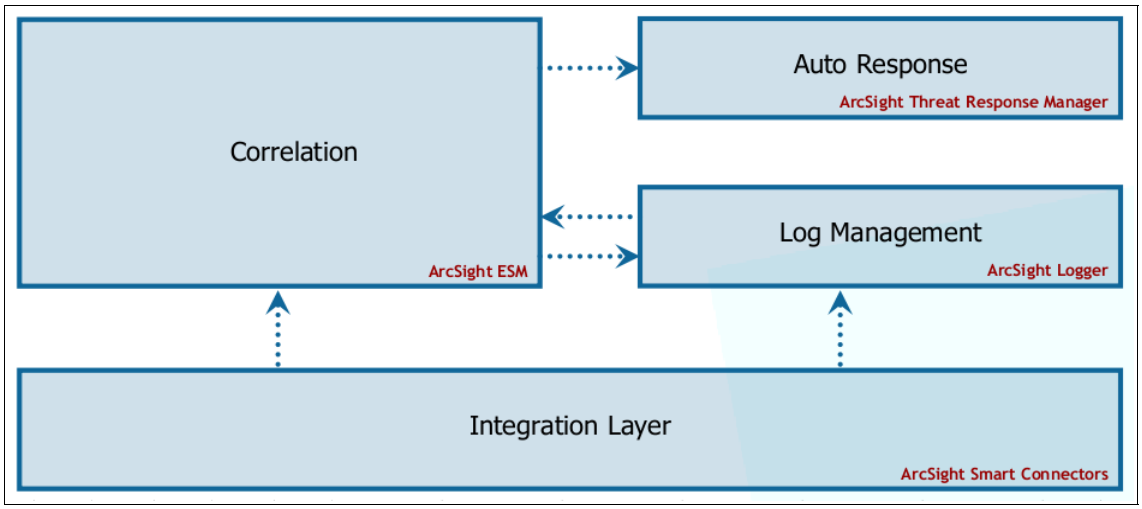


Figura 13 - Comunicación e Interacción entre componentes de ArcSight [ASL10]

En general los productos ArcSight cuentan con paquetes opcionales que habilitan a sus equipos para cumplir con reglamentaciones para el manejo de

información como ser: SOX[7] (utilizado en entornos bancarios y financieros), PCI DSS[8] (utilizado por agentes que participan en la cadena de manipulación de información asociada a tarjetas de crédito), HIPAA[9] (utilizado para la gestión de registros médicos) entre otros.

Estos dispositivos están disponibles para ser implantados en dos opciones; como hardware independiente o como software para ser instalado en un equipo ya existente. [MiHaVa11]

Dado que ArcSight utiliza terminología específica para sus productos, y a fin de conocerlos y referenciarlos posteriormente, haremos una breve descripción de los mismos:

- **ArcSight ESM Manager:** Actúa como el procesador central del ArcSight ESM. Se encarga de descifrar y descomprimir los eventos que llegan desde el Smart Connector, procesa los logs según las reglas predefinidas, los almacena en la base de datos y realiza las consultas históricas sobre la misma.
- **ArcSight ESM Database:** Se trata de la base de datos Oracle que es utilizada para el almacenamiento de los eventos recibidos desde otros dispositivos, la configuración del sistema y el conjunto de reglas.
- **ArcSight Partition Archiver:** Es el encargado de gestionar y actualizar la base de datos.
- **ArcSight ESM Console:** Se trata de una consola Java que es la interfaz del ArcSight ESM.
- **ArcSight SmartConnector:** Es una pieza de software (o hardware) encargada de recolectar en sus formatos nativos los logs provenientes de los dispositivos fuente. Sus funciones incluyen la normalización de los logs a un formato común (CEF, Common Event Format) en la cual el ArcSight ESM trabajará, y el envío de los mismos hasta el ArcSight Manager de una forma segura.
- **ArcSight ESM Web:** Se trata de una interfaz web que actúa como una consola más “liviana”, permitiendo un acceso rápido a la información específica de incidentes de seguridad.
- **Common Event Format (CEF):** Es el formato en el que ArcSight ESM analizará y almacenará los registros generados por los dispositivos fuente.
- **Evento Base:** Un evento base es un evento generado por un dispositivo fuente, por ejemplo un switch o un router.
- **Evento correlacionado:** Un evento correlacionado es aquel generado a partir de otros eventos y del cumplimiento de una o más reglas, o del cruce de algún umbral/norma preestablecida. [MiHaVa11]

A la hora de pensar en un despliegue de sensores parcial o totalmente distribuido, cabe destacar que los componentes antes mencionados de ArcSight soportan sistemas operativos tanto Linux como Windows, en 32 o 64 bits. [SNEH11].

Por otro lado, el ArcSight SmartConnector cuenta con capacidad para colectar logs de los sistemas y equipos mencionados en el link referenciado como ([ASP11]).

[7] http://es.wikipedia.org/wiki/Ley_Sarbanes-Oxley
[8] https://www.pcisecuritystandards.org/security_standards/
[9] http://www.hhs.gov/ocr/privacy/

V.3.2 – ArcSight SmartConnectors

ArcSight SmartConnector es la aplicación encargada de colectar los logs desde los dispositivos de red, normalizarlos y enviarlos al ArcSight ESM. Desde este punto de vista, el se alinea a dos de los componentes anteriormente descriptos en nuestro SIEM genérico: el módulo de recepción de logs y el módulo de normalización.

Con su preconfiguración inicial, ArcSight SmartConnector es capaz de recibir y realizar el "parse" de los logs de más de 250 dispositivos, cubriendo entre otras, las categorías: Antivirus, bases de datos, Honeypot, HIDS, NIDS, Firewalls, Switch, Wireless. [SNEH11]

La normalización de los logs colectados al formato común (CEF) permitirá luego al ArcSight ESM procesarlos de forma más rápida y eficiente, dado que la normalización se hace a nivel del propio conector.

Al igual que otros módulos (como ser el ESM) el SmartConnector existe en dos versiones: hardware independiente (appliance) o software para ser instalado en un equipo existente. Pudiendo lograr con los primeros procesamientos de hasta 5000 EPS. En el caso de utilizar la solución en software, no existe un dimensionamiento específico dado que se depende de factores a nivel de hardware y de software que variarán según el caso.

Un punto importante a considerar, es el caso de requerirse recolectar de dispositivos fuente que no poseen un conector nativo. En esta situación, se incluyen módulos que permiten extender la compatibilidad de colección a dispositivos no compatibles inicialmente. Esto se conoce como "*Flex connectors*" y consiste en la posibilidad de crear "connectors" que se adapten a los procesos de logging de casi cualquier dispositivo.

En el caso de organizaciones con recursos limitados, SmartConnector posee además filtros opcionales de salida, transmitiendo así únicamente aquellos datos que se consideren indispensables para el análisis. La idea en este sentido es economizar en ancho de banda y en espacio de almacenamiento.

Ahora bien, ¿cómo trabaja el SmartConnector? Las funciones primarias del ArcSight Connector pueden separarse en cuatro capas: Extracción de eventos, Normalización, Categorización, y Envío, tal como se muestra en la Figura 14.

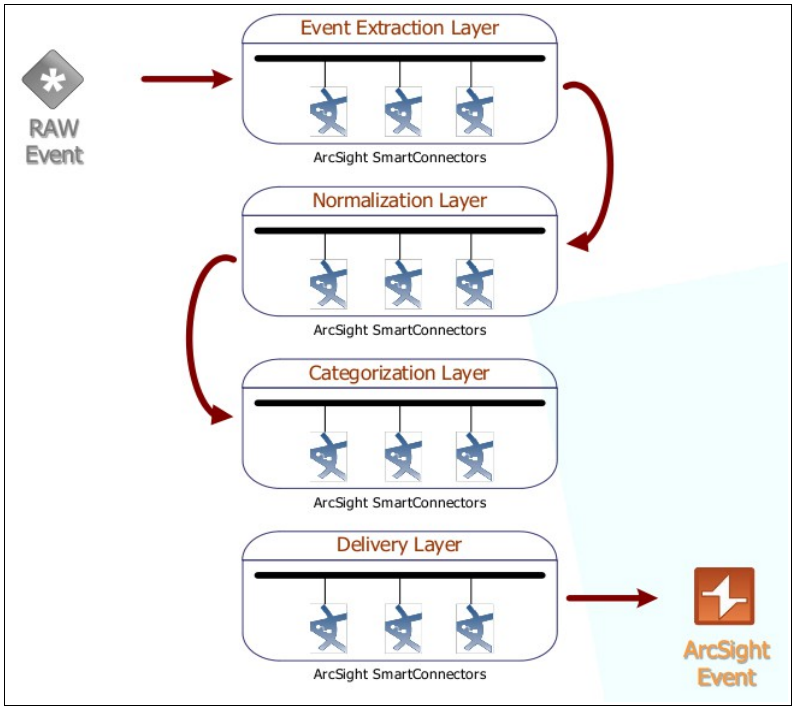


Figura 14 - Funciones del SmartConnector de ArcSight [ASL]

Capa de Extracción

Es la encargada de recepcionar la información proveniente de los dispositivos fuente en su protocolo nativo de reporte, pudiendo interpretar notificaciones en Syslog o Traps SNMP, así como buscar en campos de archivos XML, o mediante el uso de expresiones regulares en archivos de texto plano (RegExp). Cuenta también con la capacidad de buscar en bases de datos ODBC (Open Data Base Connectivity), o en otros formatos mediante el desarrollo de Connectors a medida (Flex API).[ASL10].

Existen dos opciones de despliegue para realizar el trabajo que involucra esta capa, las cuales se muestran en la Figura 15. La primera opción consiste en instalar agentes en los dispositivos a sensar, mientras que la segunda es permitiendo que ArcSight se encargue de la colección.

La opción sin agente (agentless) implica una implementación más fácil al no estar condicionada por el tipo de sensor del que se desea colectar información. La opción con agente en el dispositivo sensor está condicionada a la existencia del agente para la plataforma en cuestión, siendo el OS una de las mayores limitantes.

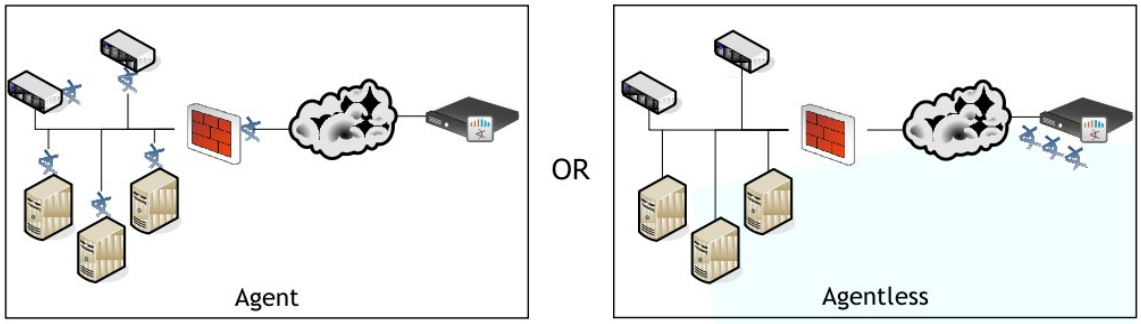


Figura 15 - Opciones de despliegue para la capa de extracción [ASL10]

Capa de Normalización

Su función es la de traducir el formato nativo de los eventos a un formato común utilizado por ArcSight (CEF). A su vez se encarga de presentar los eventos de forma resumida destacando aquellos aspectos relevantes. Un ejemplo de normalización se muestra en la Figura 16.

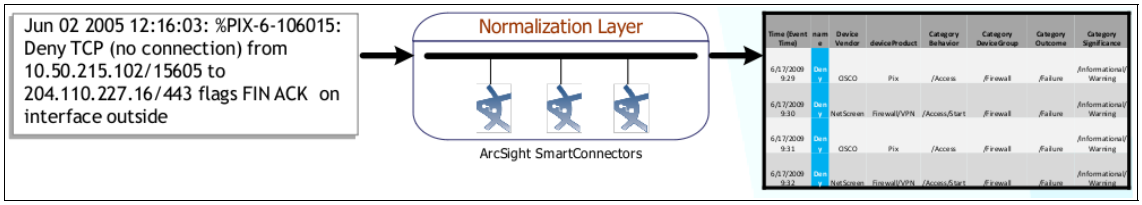


Figura 16 - Proceso de Normalización de ArcSight [ASL10]

Capa de Categorización

El fin de esta capa es agrupar los logs normalizados según su categoría, a fin de lograr una visualización clara y resumida. Para esto los eventos provenientes de distintos dispositivos fuente, y cuya información sea un evento en común, serán catalogados de idéntica forma. Esto permite desvincular al evento de su fuente y por tanto el resumen de la información

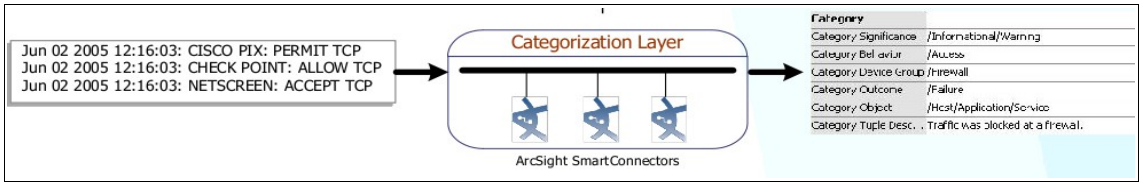


Figura 17 - Proceso de Categorización de ArcSight [ASL10]

Capa de Entrega

Es la encargada de entregar la información generada por los eventos a los equipos que corresponda ya sea el ESM y/o uno o varios ArcSight Loggers. Como se puede observar en la Figura 18, ArcSight Connector admite varios escenarios para la entrega de la información.

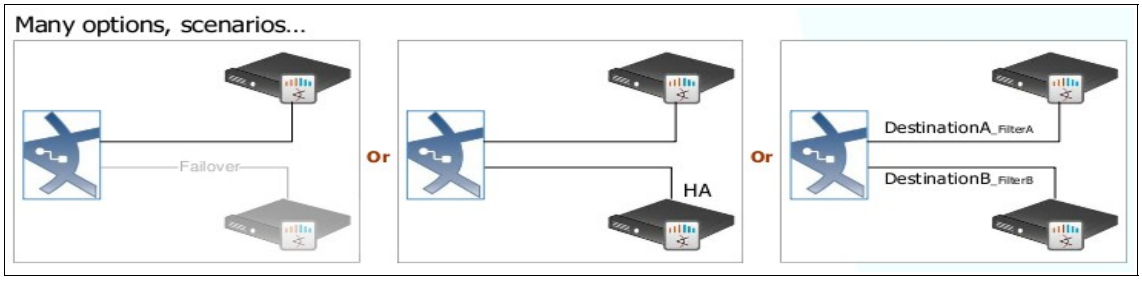


Figura 18 - Arquitecturas para el Delivery de los logs en ArcSight [ASL10]

Al igual que en soluciones de otros fabricantes, una vez que los logs llegan al primer equipo propietario, la comunicación entre éste y sus pares será cifrada y en un formato propietario. La capa en cuestión también se encarga de este aspecto, cifrando y comprimiendo los mensajes enviados. [SNEH11]

V.3.3 – ArcSight Logger

Un problema común al momento de implementar un SIEM es la gran cantidad de logs que son necesarios para el trabajo así como la compatibilidad de sus formatos. ArcSight Logger se presenta como una solución "universal" para la gestión de logs la cual unifica la búsqueda y permite además la presentación de informes, alertas y análisis a través de cualquier tipo de datos de logs. Esto lo hace apto para recopilar, analizar y almacenar grandes cantidades de datos generados por las redes actuales. Es compatible con múltiples opciones de despliegue y puede ser implementado como un hardware independiente o como software instalado en un equipo.

Podríamos tener la inclinación de alinear este módulo, en nuestro modelo de SIEM genérico, como el módulo destinado al almacenamiento de logs. Sin embargo, en el caso de ArcSight Logger, su función es más particular. Esto es claro cuando este módulo es opcional, dado que los eventos son almacenados en una base de datos destinada para ese fin, la cual, en nuestro estudio consideramos como el correspondiente al almacenamiento. Por tanto, ArcSight Logger no tiene un correspondiente en nuestro SIEM genérico.

En este sentido, este módulo se definiría como un "Log Manager", representando un complemento a una solución SIEM.

En la arquitectura de componentes de ArcSight, no es obligatorio la inclusión del Logger. Sin embargo, el ESM se caracteriza por compactar la información y enfocarse en la correlación de la misma. Por otra parte, el Logger se orienta a una larga retención de logs y facilidades de búsqueda sobre los mismos. Desde este punto de vista, es un complemento a la solución del SIEM por lo que su implementación agrega una capa de gestión de logs que mejora el manejo del entorno.

Podemos decir entonces que ArcSight Logger combina las tareas de gestión de logs, la supervisión del funcionamiento de los dispositivos y el control de la seguridad.

Una vez que los logs se encuentran dentro del ArcSight Logger, podemos separar aquellos que son necesarios para el control de la seguridad del resto de los logs. Para ello, se utiliza un motor de búsqueda con el objeto de obtener la información específica necesaria. [SNEH11]

Una idea de la flexibilidad en el almacenamiento que presenta este equipo puede observarse en la Figura 19. En ella se muestra como en base a los dispositivos sensados es posible definir grupos de dispositivos, existiendo incluso la posibilidad de que un dispositivo pueda pertenecer a más de un grupo. A su vez, cada grupo de dispositivos tendrá definido políticas diferentes de retención, especificando el número de días que los eventos serán retenidos y el tamaño máximo de almacenamiento.

Otro escenario posible es el manejo de eventos en base a la dirección IP de origen. De esta forma también pueden ser direccionados a grupos particulares de almacenamiento permitiendo por ejemplo priorizar los equipos más críticos de la organización con mayores capacidades de almacenamiento. [SNEH11]

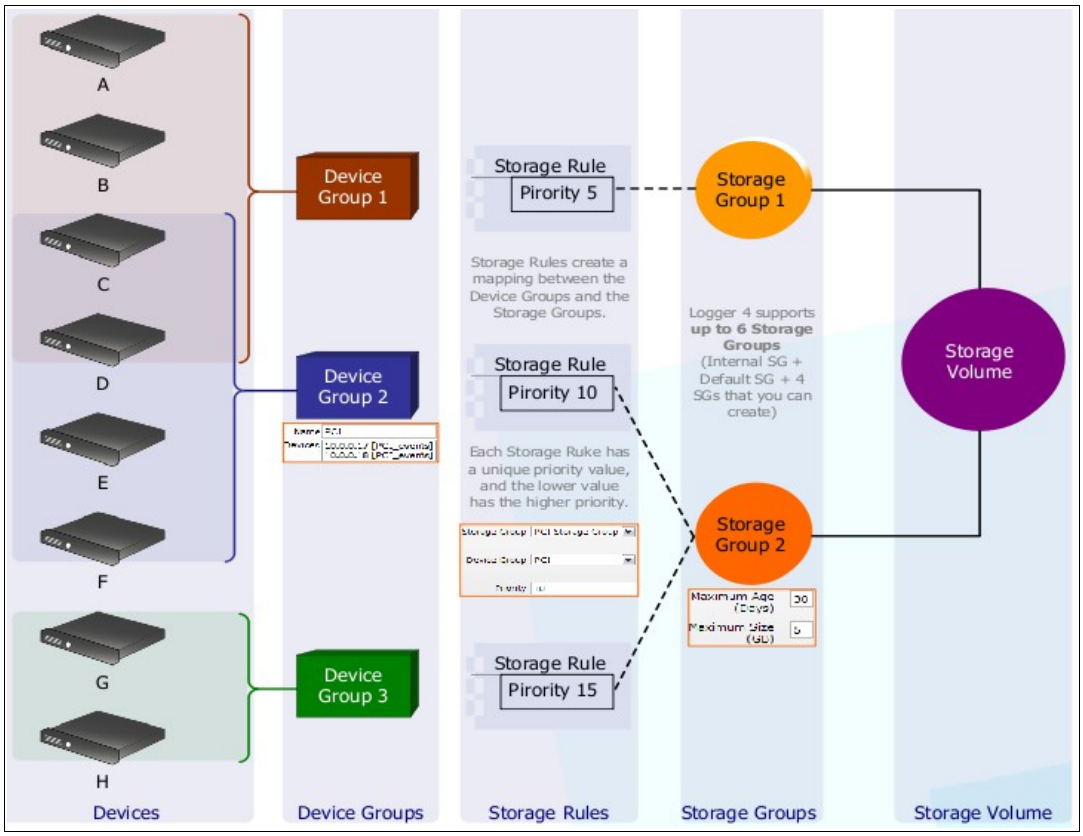


Figura 19 - Posibilidades de storage para el ArcSight Logger [ASL10]

V.3.4 – Comunicaciones en la arquitectura ArcSight

La Figura 20 resume los puertos y protocolos utilizados por la arquitectura ArcSight para el intercambio de información entre sus principales componentes. Puede observarse cómo los datos transferidos son cifrados (SSL) y el protocolo de transmisión utilizado garantiza la entrega de la información (TCP).

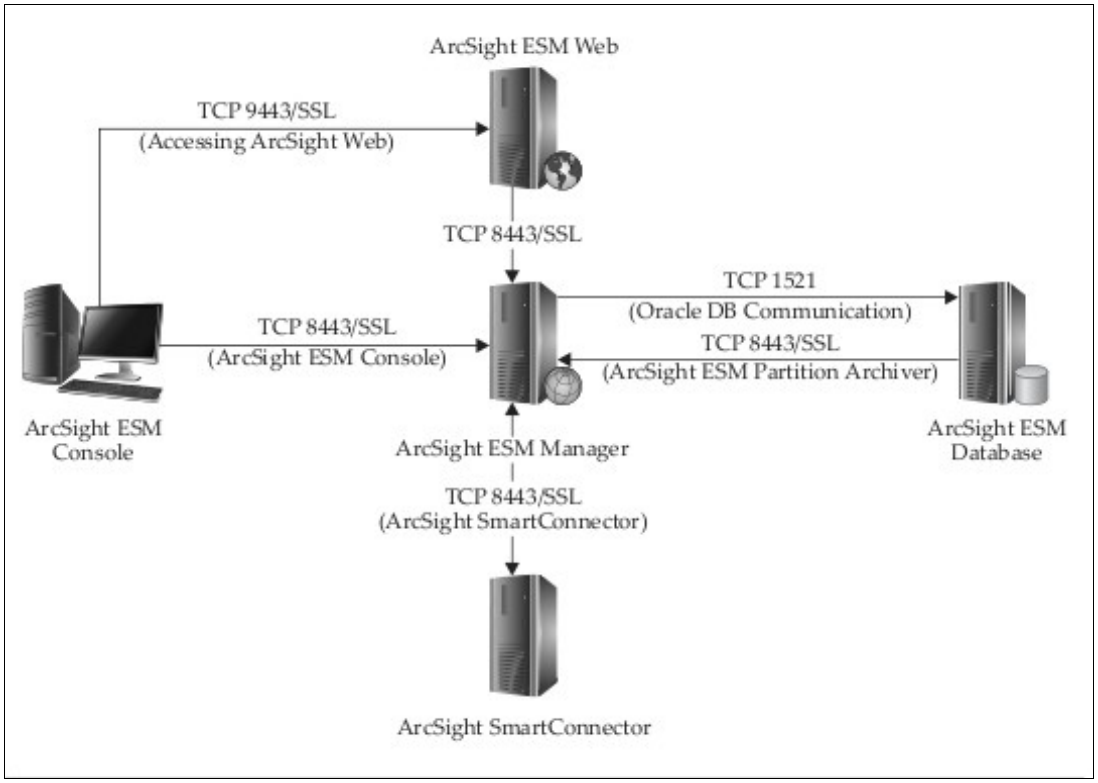


Figura 20 - Diagrama de comunicación entre componentes de ArcSight [MiHaVa11]

A continuación se describen las acciones involucradas en el envío de logs, desde un dispositivo fuente hasta su almacenamiento en la base de datos, incluyendo su proceso en el SmartConnector. Los puertos y protocolos utilizados en cada caso se detallan en la Figura 20, Figura 21, y Figura 22.

1) **Recepción de logs:** Los logs provenientes desde los dispositivos fuente pueden colectarse por métodos de push o pull según se cuente o no con agentes en los primeros.
   En caso de utilizar Syslog, estos pueden ser enviados directamente desde el dispositivo al SmartConnector quien quedará a la espera de los mismos.
   Por el contrario, en caso de utilizar Windows Event logs, será necesario configurar al SmartConnector con la información del servidor y sus credenciales de usuario a fin de que le sea posible establecer una conexión con el servidor y colectar sus logs.
   En la Figura 21 y Figura 22 se muestra cómo la comunicación entre los componentes propietarios (SmartConnector y Manager) es asegurada mediante el uso de SSL sobre TCP.

2) **Procesamiento de logs:** Sin importar el formato nativo con el que arriben los logs, el SmartConnector los traducirá al formato único para el ESM, CEF.

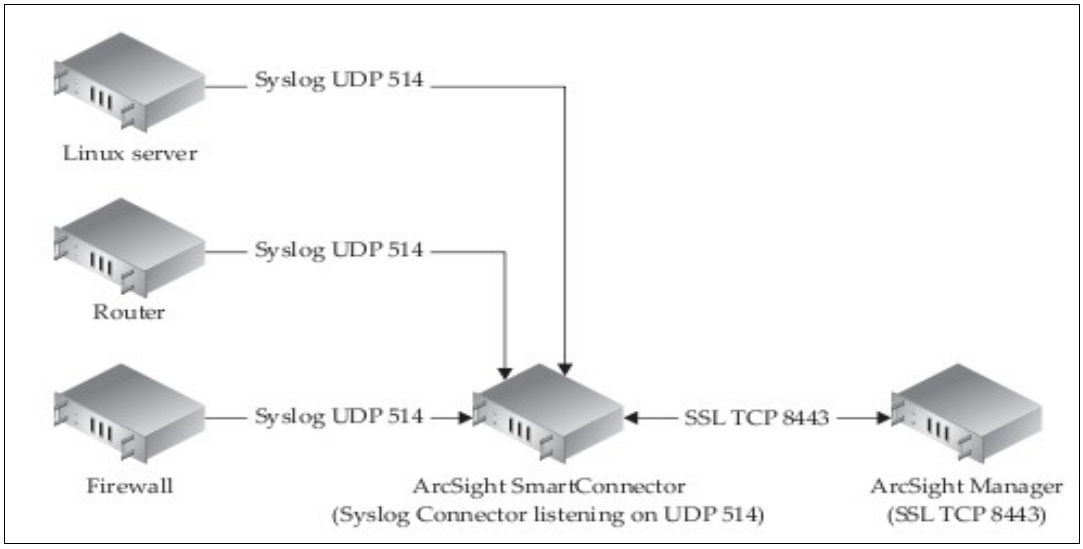


Figura 21 - ArcSight SmartConnector push [MiHaVa11]

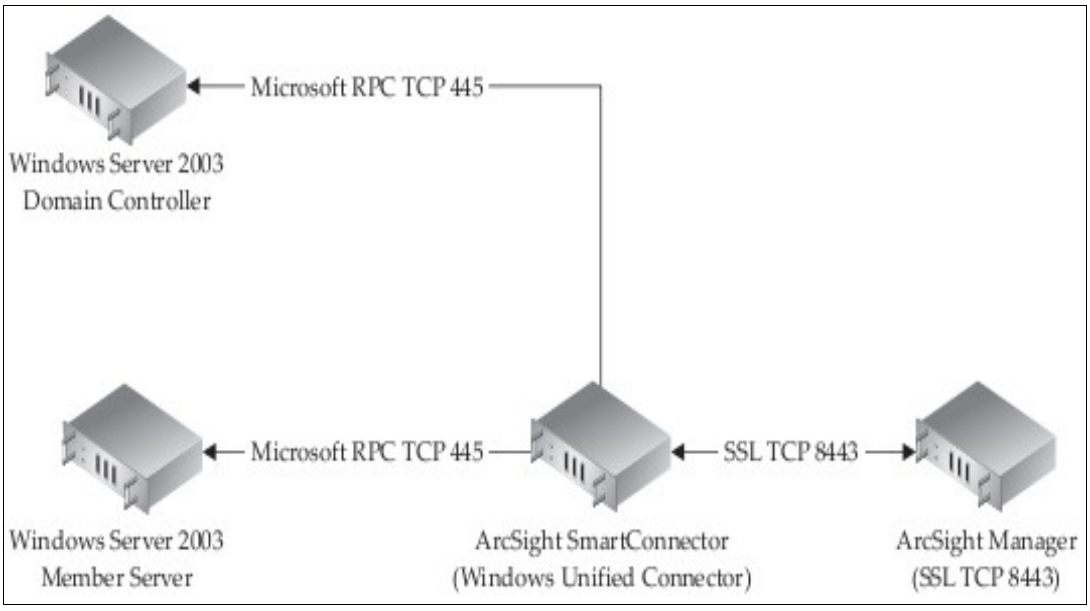


Figura 22 - ArcSight SmartConnector pull [MiHaVa11]

3) **Compresión y cifrado de logs:** A fin de garantizar una transmisión segura, los logs son enviados sobre SSL por el puerto 8443 al ArcSight Manager; y codificados utilizando un certificado generado por el ESM.
Dichos logs también serán comprimidos a fin de reducir la utilización de los enlaces de la red. En este sentido, es posible configurar al ArcSight tanto para que transmita los logs en tiempo real como para que estos sean transmitidos exclusivamente en horarios de bajo tráfico.

4) Recepción de logs por el Manager: El ArcSight Manager autenticará todas las conexiones de entrada a fin de segurizar la recepción de

información. Luego de finalizada la conexión SSL entre el Manager y el Connector se procede a la descompresión y descifrado de los logs.

5) **Procesamiento de logs por el Manager:** El ArcSight Manager comenzará por procesar los logs, aplicándoles las reglas y combinándolos con otros eventos a fin de poder correlacionarlos.

6) **Alertas generadas por el Manager:** Cuando ciertos eventos ocurran o cuando se haya sobrepasado umbrales predefinidos, es posible configurar al Manager para que envíe alertas, ya sea por email, o bien desplegándolas y presentándolas en la consola.

7) **Envío de logs a la base de datos:** El ArcSight Manager se comunicará con el servidor ArcSight Database, donde reside la base de datos, a fin de enviarle los nuevos logs.

8) **Almacenamiento de logs:** Una vez que la base de datos ha recibido los logs desde el Manager procederá a indexarlos a fin de facilitar su futura búsqueda.

Como se puede observar en esta sección, la mayoría de los intercambios de información en los procesos ArcSight son seguros. Sin embargo hay dos instancias que no utilizan cifrado [MiHaVa11]. La primera se da cuando el ArcSight Manager envía los eventos a la base de datos Oracle (utilizando el puerto 1521). La segunda instancia depende del SmartConnector que se esté utilizando. Si se está empleando Syslog, los logs estarán llegando al SmartConnector a través de UDP de forma no segura y sin garantía de entrega. Si por el contrario se utilizan otros SmartConnector más seguros como ser el Windows Unified Connector, éste utiliza RPC para la transferencia de los logs.

V.3.5 – ArcSight ESM Manager

Hasta este momento, hemos descripto algunos de los componentes que constituyen la arquitectura de ArcSight. Hemos presentado los SmartConnectors que cumplen la función de recepción de los eventos desde los dispositivos fuente y su normalización. También hemos presentado al Logger como una herramienta adicional a la arquitectura, a fin de obtener información que perdería su característica “nativa” al ser procesada por el ESM.

El siguiente punto en nuestro estudio de ArcSight sería el correspondiente al Motor de Reglas/Motor de Correlación, según la arquitectura de nuestro SIEM genérico. En este punto, debemos identificar al ArcSight ESM como dicho módulo.

Es natural tratar de realizar una descripción detallada de los procesos que internamente hacen al ESM como este módulo de análisis automático. Sin embargo, y dada la naturaleza de esta solución como software de tipo privativo, no se posee los detalles de dicha funcionalidad.

La única aproximación es que existe algún mecanismo basado en reglas. Esta suposición se basa en que es posible la construcción de las mismas, a fin de

incorporar las “reglas del negocio”. Sin embargo, y dado que otros mecanismos internos son utilizados, esto escapa a la información disponible en forma pública.

De la información disponible, sabemos que el ESM efectúa correlación basada en reglas, posee la capacidad de realizar perfiles según el contexto, y que es capaz de aplicar filtros y prioridades a los eventos procesados permitiendo realizar la correlación. El detalle de estos algoritmos y aplicaciones no es posible analizarlos por las razones antes citadas.

Como detalle adicional, la implementación de ESM es basada en Java, siendo su ejecución determinada por la JVM del servidor donde se alojará. Los recursos que se asigne a dicha JVM, en consecuencia, determinarán los recursos disponibles para el ESM, por lo que será necesario seguir las recomendaciones orientadas a su ajuste en función del tamaño del despliegue considerado, siendo el “memory heap size” uno de los parámetros fundamentales.

V.3.6 – Base de datos Oracle

La base de datos sobre la cual ArcSight se soporta es Oracle. Como aplicación modular, dicha base está pensada para operar en un servidor diferente al utilizado para el ESM. Esto no es necesariamente la regla para entornos reducidos, por lo que el propio instalador de ESM ofrece instalación de la base de datos Oracle en el mismo servidor.

Dicha instalación también automatiza todos los procesos destinados a que dicha base esté preparada para su utilización por el ESM, como ser la generación de instancias, creación de tablas, esquemas, recursos, entre otros.

V.3.7 – ArcSight Console

Si bien hemos indicado la existencia de un cliente o consola más “liviana” (ArcSight ESM Web) para el acceso al ESM, sólo el ArcSight Console tiene a disposición todas las posibilidades de gestión y configuración de los recursos presentes en el ESM.

A diferencia de los componentes anteriores, los cuales se consideran parte del core de la solución, el Console se instalará en la estación del usuario final quien operará la solución. Es, por tanto, el componente “client” de la solución.

La diferencia fundamental con la consola “liviana”, es que desde el Console es posible también configurar y acceder al resto de los componentes como los SmartConnectors y la base de datos.

Considerando los componentes de nuestro SIEM genérico, el ArcSight Console junto al ArcSight ESM constituyen el componente de “Monitoreo”.

V.3.8 – ArcSight Web

Este componente permite el acceso a la operación del ESM sin utilizar un cliente específico. La idea es acceder desde cualquier dispositivo con el único requerimiento de poseer un navegador web.

Como se expresara con los anteriores componentes, el mismo es independiente y puede o no estar instalado en el mismo servidor donde se realizó el despliegue del ESM. Muchas veces, se opta por instalar ArcSight Web en un segmento aislado, siendo accesible desde Internet mediante algún recurso privilegiado (ej. VPN).

V.3.9 – ArcSight Web Client

Si bien lo indicamos como un componente más, en realidad no existe como tal. El cliente para el acceso al ArcSight Web es simplemente un browser, siendo soportado tanto en versiones de Microsoft Internet Explorer como Mozilla Firefox.

Nuevamente, alineando los componentes a nuestro SIEM genérico, el ArcSight Web y un browser constituyen, en este caso en una forma alternativa a la ya vista, el componente de “Monitoreo y Presentación”. [SNEH11]

V.4 – Q1 Labs QRadar

V.4.1 – Introducción

Al igual que ArcSight los productos de Q1 Labs están disponibles tanto en hardware como en software. Su producto principal en seguridad se denomina QRadar SIEM. Este sistema incluye todo lo necesario para la recolección de información, la gestión de eventos de seguridad, el almacenamiento de logs, el monitoreo del tráfico, y la generación de reportes.

En particular QRadar incluye:

- Monitor de eventos de seguridad.
- Monitor para el tráfico de red.
- Scanner de vulnerabilidades integrado.
- Inventario de activos y generación de perfiles.
- Capacidad de análisis y correlación de datos.
- Detección heurística de amenazas y priorización.
- Generación de reportes.

Cabe destacar que junto con ArcSight, QRadar de Q1 Labs está entre los SIEM mejor ranqueados en los últimos informes de Gartner. [MiHaVa11]

V.4.2 – Arquitectura de QRadar

Comencemos por el appliance QRadar SIEM. Este sistema será quien reciba y almacene los reportes desde los dispositivos sensados de la red o de otros sistemas

de seguridad, ya sea mediante eventos de Syslog o de Windows Event Log. Así mismo QRadar SIEM Appliance puede recibir flujos de datos desde otras aplicaciones como ser NetFlow de Cisco, J-flow de Juniper, Nessus o QualysGuard.

Al observar la descripción precedente, queda claro que este appliance constituye una solución de tipo “All-in-one”, ya que puede recibir los eventos de los dispositivos fuente directamente, siendo todo el posterior procesamiento realizado en forma interna. Desde este punto de vista, este appliance abarca todos los componentes de nuestra solución genérica.

Además, este sistema cuenta con puertos de monitoreo para el “sniffing” del tráfico sensado, completando así un análisis del tráfico hasta la capa aplicación.

Una vez recibidos los datos desde los diferentes sensores, procederá a normalizarlos, clasificarlos, aplicarles las reglas de filtrado y matching correspondientes, almacenarlos y finalmente analizarlos y correlacionarlos.

En la Figura 23 se muestra un esquema de cómo el sistema SIEM QRadar procesa los diferentes tipos de eventos así como aquellos módulos involucrados en dicho proceso.

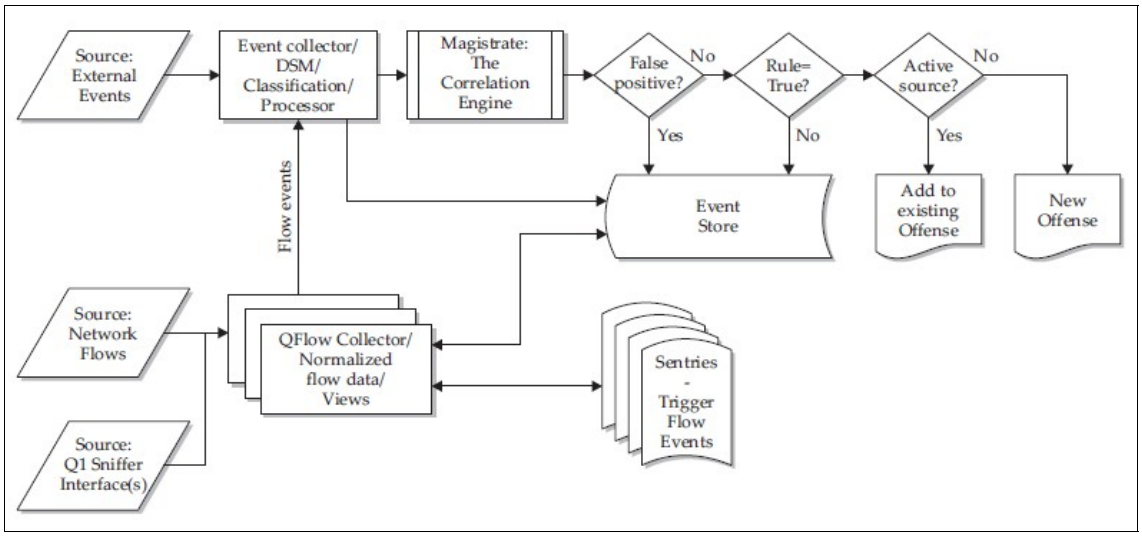


Figura 23 - Arquitectura básica de QRadar [MiHaVa11]

El proceso se inicia luego de que un evento o flujo ha sido recibido por la unidad QRadar, la cual filtrará este evento o flujo utilizando objetos conocidos como BB: Building Blocks. Los datos convertidos, denominados Views, serán pasados a los Sentries o Centinelas. Estos últimos generarán eventos internos que serán enviados al motor de procesamiento de reglas el cual se denomina Magistrate.

En esta descripción, el Event Collector y Qflow Collector, constituyen el Colector y Normalización de logs de nuestro modelo genérico, mientras que el Magistrate lo es con respecto al Motor de reglas/Motor de correlación.

Lo interesante de esta arquitectura es la existencia de un módulo especializado en la recepción de flujos, sea en la forma de NetFlow, sFlow u otros, o de capturas en modo “sniffer”. Esto constituye una forma de incorporación de la detección en base a anomalías de tráfico, aceptando que su operación no es tan parametrizable como en el caso del procesamiento de eventos.

En este punto, el Magistrate utiliza un sistema de ponderación en base a varios atributos configurables para determinar si la situación es lo suficientemente hostil como para implicar una ataque. En caso afirmativo se disparará un aviso al profesional de seguridad para que analice la situación a fin de mitigar los daños y solucionar la amenaza.

Una vez remediada la amenaza es posible utilizar QRadar para elaborar informes detallados sobre las acciones específicas que provocaron las circunstancias hostiles y las medidas adoptadas para asegurar la situación.

La solución SIEM "All-in-one" puede ser ampliada con el módulo colector QFlow, el cual proporciona la capacidad de monitorear flujos en segmentos distribuidos. Un ejemplo de esta solución se muestra esquemáticamente en la Figura 24.

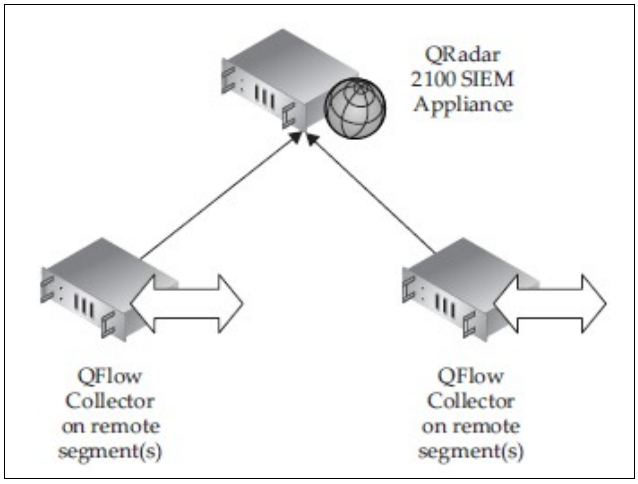


Figura 24 - Solución QRadar 2100 SIEM con módulos Qflow Collector [MiHaVa11]

Para organizaciones de mediano a gran porte los equipos QRadar permiten ser expandidos adicionándoles módulos QFlow Collector, así como módulos Event Processor (módulos dedicados al procesamiento de eventos), o Flow Processor (módulos dedicados al procesamiento de flujos). Un ejemplo de esta solución se muestra en la Figura 25.

Tal vez la diferencia con respecto a otras soluciones es que en el caso de QRadar, el formato appliance es el elegido para la distribución de la solución no considerándose la opción de componente de software a instalar. Si bien esto asegura un óptimo acoplamiento de los componentes, limita la libertad de elección y tiene la tendencia a que se pueda construir formas extremadamente propietarias de operación, limitando la así flexibilidad de agregar componentes de terceros. Solo quedaría la integración a través de los Event Processor. [SNEH11]

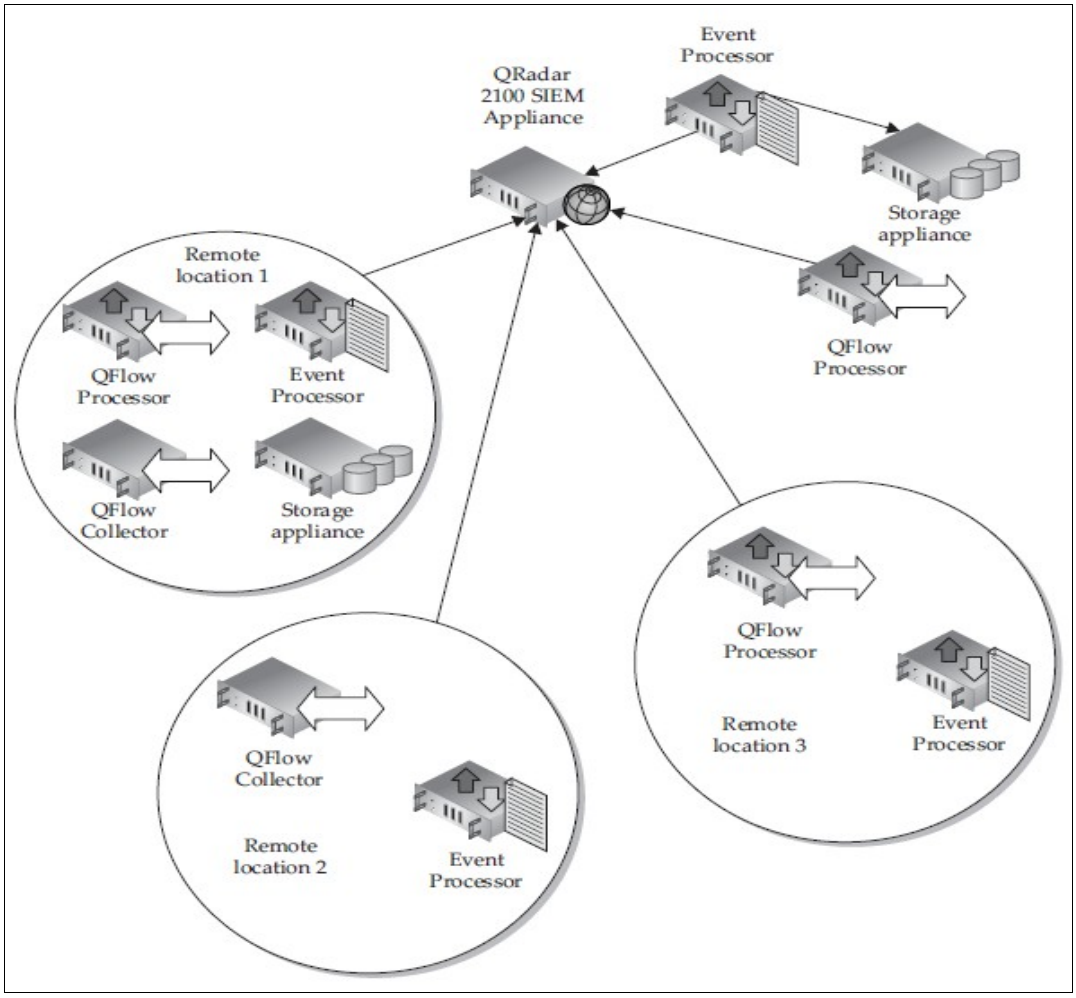


Figura 25 - Solución QRadar 3100 SIEM con módulos de expansión [MiHaVa11]

Todos los dispositivos mencionados (QFlow Collector, Event Processor, y Flow Processor) permiten ser instalados en locaciones remotas, obteniendo funciones de procesamiento, almacenamiento y recolección de logs en forma distribuida. Debido a esto QRadar debe ser capaz de trabajar con flujos de datos provenientes de diversas fuentes, incluso distribuidas. En este sentido cabe aclarar que, por defecto, las conexiones entre dispositivos de QRadar serán cifradas usando SSL. [MiHaVa11].

V.4.3 – Nomenclatura de Q1 Labs

Al igual que otros proveedores Q1 Labs tiene su propia nomenclatura para referirse a sus componentes, sistemas y servicios particulares. Algunos de estos términos se describen a continuación:

- **Adaptive Log Exporter (ALE)**: Se trata de un agente específico encargado de colectar, normalizar, y exportar logs en sistemas Windows. Incluye compatibilidad con: Microsoft IIS, Microsoft IAS, Microsoft Exchange mediante OWA o SMTP y Trend Micro InterScan.

- **Building Block (BB)**: Es un filtro reutilizable generalmente usado dentro de reglas más complejas. Un ejemplo, sería un BB que seleccione los servidores SMTP de la organización, y luego utilice dicho BB para excluir dichos servidores del muestreo de red en otra regla.
- **Device Support Module (DSM)**: Utilizado para analizar los eventos provenientes de diversos dispositivos en sus formatos nativos y traducirlos al formato estándar de QRadar; nuestro componente de normalización genérico.
- **Flow**: Conjunto de paquetes que describen la comunicación entre dos hosts y comparten algunas características comunes.
- **Log Source**: Efectúa el mapeo de la información contenida en un evento que llega en su propio formato, a un DSM para la ejecución del parsing correspondiente. Es el componente individual de un dispositivo fuente para su normalización.
- **Magistrate**: Es el encargado de proveer el núcleo de procesamiento del SIEM. Con el mismo, se analizan y procesan los eventos contra las reglas definidas a fin de determinar actividad maliciosa. Corresponde a nuestro Motor de reglas/Motor de correlación, tal cual fuera indicado anteriormente.
- **Network Behavior Anomaly Detection (NBAD)**: Se trata de una herramienta encargada de monitorear de forma continua la red en busca de comportamientos o eventos inusuales, con la función de proteger preventivamente contra ataques de día cero.
- **Offense**: Es la forma en que QRadar denomina a los incidentes percibidos como indicadores de amenazas.
- **Q1 Labs Event Identifier (QID)**: Se trata de una tabla con la lista de los dispositivos propietarios cuyos eventos son reconocidos. El propósito es asistir al sistema en la normalización de estos eventos al formato utilizado por la base de datos de QRadar. QID se encargará también de encaminar los eventos hacia el módulo Magistrate, para su análisis, clasificándolos por categorías, subcategorías, severidad y credibilidad.
- **QRadar Request Language (QRL)**: Refiere a las definiciones de los flujos de datos y los elementos de red que serán supervisados, las cuales serán guardadas en un registro.
- **Rules**: Al igual que en otros dispositivos, las reglas se refieren a las condiciones, patrones o series de pruebas a las que se someten los flujos de datos en busca de comportamientos no deseados.
- **Sentries**: Se componen de un conjunto de monitores y filtros dedicados al control de los flujos de datos. El objetivo es la generación de eventos y alertas.
- **SuperFlow**: Se trata de la compresión en un sólo registro de varios flujos de datos que presenten un patrón similar, constituyendo un proceso elemental de correlación. Las agrupaciones mencionadas se dan en las siguientes categorías:
    - una fuente a varios destinos. Tratados como escaneos de red.
    - varias fuentes a un destino. Tratados como DDoS (denegación de servicio distribuido).
    - Una fuente a un destino con cambio de puerto. Tratados como escaneos de puerto de host.
- **View**: QRadar Views son métodos de QRadar para la organización de los flujos de datos, ya sea para su presentación o para su análisis en los Sentries. Se

componen esencialmente de filtros aplicados a los flujos de datos, permitiendo obtener un dictamen rápido de la peligrosidad del flujo. Las diversas QRadar Views se organizan jerárquicamente y son fácilmente editables.

V.4.4 – Interfaz administrativa de QRadar

Accediendo a la interfaz de gestión mediante un navegador web, y sobre el protocolo HTTPS, es posible apreciar la vista inicial que ofrece QRadar; Figura 26

En ésta se distinguen las siguientes lengüetas :

- Dashboard.
- Offenses.
- Events.
- Assets.
- Network Surveillance.
- Flows.
- Reports.
- Admin [SNEH11]

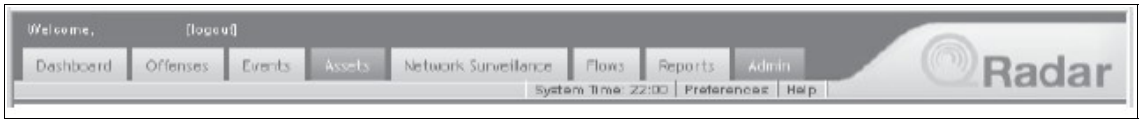


Figura 26 - Lengüetas de QRadar [MiHaVa11]

V.4.5 – Gestión de Activos

El sistema QRadar puede aprender acerca de los activos de la red a través de los datos entregados por QFlow y por los scanners de vulnerabilidades. Cada recurso identificado cuenta con un perfil donde se incluye: puertos abiertos, servicios instalados, entre otros. Estos perfiles a su vez pueden ser editados manualmente. La interfaz para gestionar dichos perfiles se muestra en la Figura 27. Con ella es posible buscar un nuevo recurso instalado o conocer el estado de los existentes. [MiHaVa11]

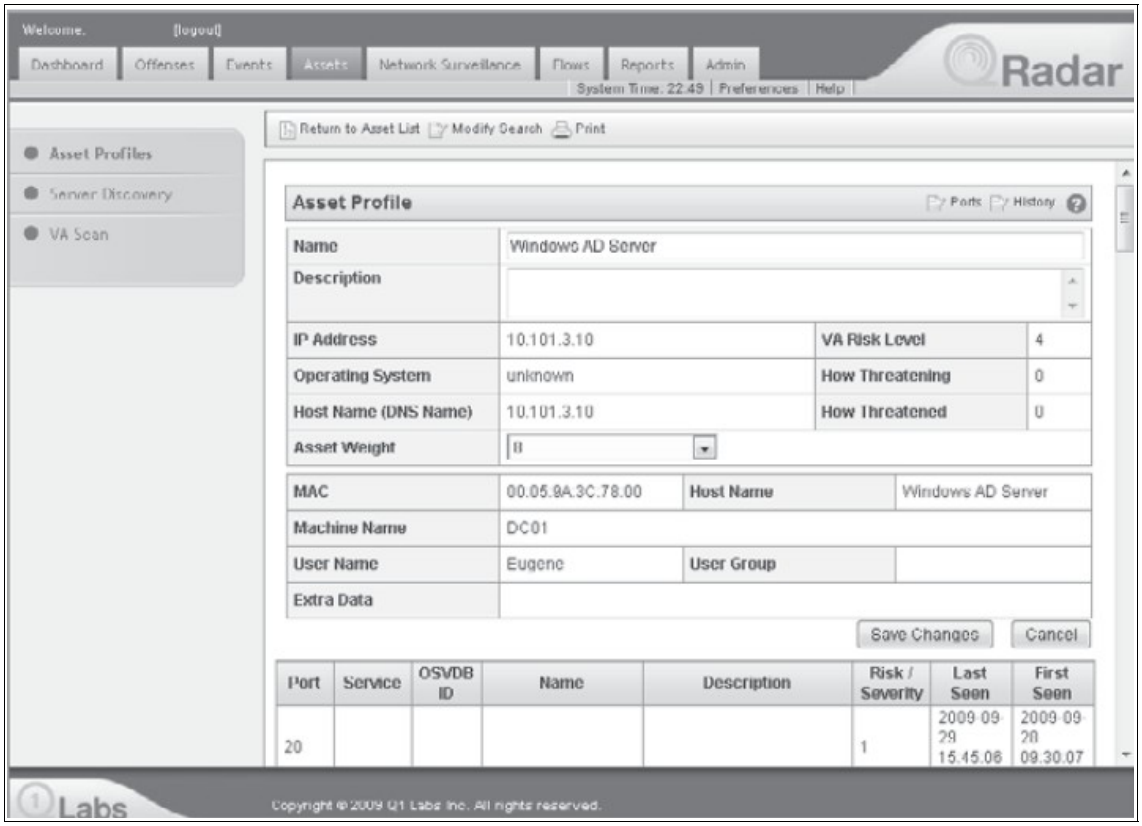


Figura 27 - Interfaz para la gestión de recursos de QRadar [MiHaVa11]

V.4.6 – Obtención de flujo de datos y eventos en QRadar

Para la tarea de obtener información fidedigna desde los dispositivos fuente, QRadar es compatible con: Syslog, SNMP, muestreo de flujos (NetFlow, JFlow, etc), Snare, y Win Agents Event Log (dispositivos Windows).

Para los sistemas no compatibles en forma nativa, QRadar cuenta con una herramienta propia para la recolección y el envío de logs. Se trata de la aplicación Adaptative Log Exporter la cual se instala en sistemas Windows, y es la encargada de colectar y exportar los logs hacia los DSM. Éstos últimos serán los encargados de realizar el parsing y procesamiento de los logs referidos. Se aprecia en este caso que si bien existe la posibilidad de extender los dispositivos fuente que reportan a QRadar, es necesario la instalación fuera del appliance de una aplicación para este fin.

Al igual que en otras soluciones, QRadar es capaz de colectar información de todo tipo de dispositivos y aplicaciones como ser: routers, firewalls, servidores, scanners de vulnerabilidades, entre otros. [MiHaVa11]

V.4.6.1 – Dispositivos generadores de eventos

Al igual que en otras soluciones SIEM, QRadar permite que los dispositivos de red puedan tanto enviar sus eventos (push) a QRadar así como éste colectarlos (pull) en los casos en que aquellos soporten alguno de los siguientes protocolos:

- Syslog
- Java Database Connector (JDBC)
- OPSEC/LEA – Check Point Log Export API
- Security Device Event Exchange (SDEE) – Cisco/SOAP based
- Juniper Network Security Manager (NSM), protocolo syslog modificado
- Simple Network Management Protocol (SNMP) v1, v2, y v3.

Luego de recibidos los datos, QRadar utiliza los DSMs para identificar el dispositivo fuente de origen y poder analizar correctamente la información. En caso de que QRadar no pueda identificar la fuente, o no cuente con un DSM adecuado, será posible construir un DSM universal definiéndose el parsing y el mapeo al formato de QRadar.

A partir de aquí los datos son analizados contra los sets de reglas que estén definidos y posteriormente almacenados. En dicho análisis se aplican comparaciones de los eventos contra firmas de ataques conocidos, detección de anomalías o variaciones de comportamiento, así como reglas personalizadas al entorno específico.

Asimismo se utiliza esta información para identificar cómo están conectados los recursos en la red y obtener información de inventario.

Si un evento es detectado como malicioso, el mismo es derivado al módulo Magistrate, el cual se encarga de su seguimiento. A su vez, si el Magistrate lo identifica como un nuevo ataque generará una nueva alerta. Por el contrario, si lo identifica con un ataque en curso, agregará dicho evento a los otros ya involucrados en la alerta previamente generada. [MiHaVa11]

Al observar con atención las funciones señaladas en el párrafo anterior, queda claro que en el caso de QRadar la función de Motor de reglas/Motor de correlación, no se encuentra en forma centralizada, sino que la misma opera en forma distribuida, lo cual también es diferente a todas las otras soluciones bajo estudio. Desde este punto de vista, aunque básico, los DSM también realizan parte de la correlación.

Una lista más detallada de los diferentes dispositivos y aplicaciones soportadas puede encontrarse en la referencia [QRPSD].

V.4.6.2 – Dispositivos generadores de flujos de datos

Ejemplos de estos dispositivos son routers y firewalls. Muchos de los fabricantes incluyen sus propios protocolos en los flujos de datos, como ser:

- QFlow de Q1 Labs
- NetFlow de Cisco Systems
- J-Flow de Juniper Networks
- sFlow de Hewlett Packard

Los flujos de datos pueden llegar a QRadar de dos formas diferentes. O bien los dispositivos reportan a una interfaz de QFlow Collector con una IP determinada, o bien el propio QRadar se dedicará a “sniffear” el segmento de red.

Por defecto QRadar SIEM entiende 136 protocolos conocidos, pudiendo ampliar esta lista a nuevos protocolos que se utilicen en el entorno.

Como se mencionara, estos flujos serán normalizados y filtrados (según los requisitos de cada organización) en las QRadar Views; y éstas serán utilizadas por los

Sentries, los cuales enviarán una alerta al QRadar Event Processor en caso de hallar coincidencias con algún perfil de amenaza establecido.

Desde dicho punto, los eventos derivados por la información de flujo serán procesados análogamente a las otras fuentes de eventos. [MiHaVa11]

Esta función también es original en QRadar, dado que se posee un módulo especializado en el manejo de datos del tipo flujo.

V.4.7 – Editor de ecuaciones

A la hora de construir nuevas Views será necesario configurar previamente el editor de ecuaciones. Este último permite definir los criterios utilizados por el sistema QRadar para el filtrado y análisis del tráfico. Mediante lógica algebraica estándar el editor de ecuaciones define el estado de las Views.

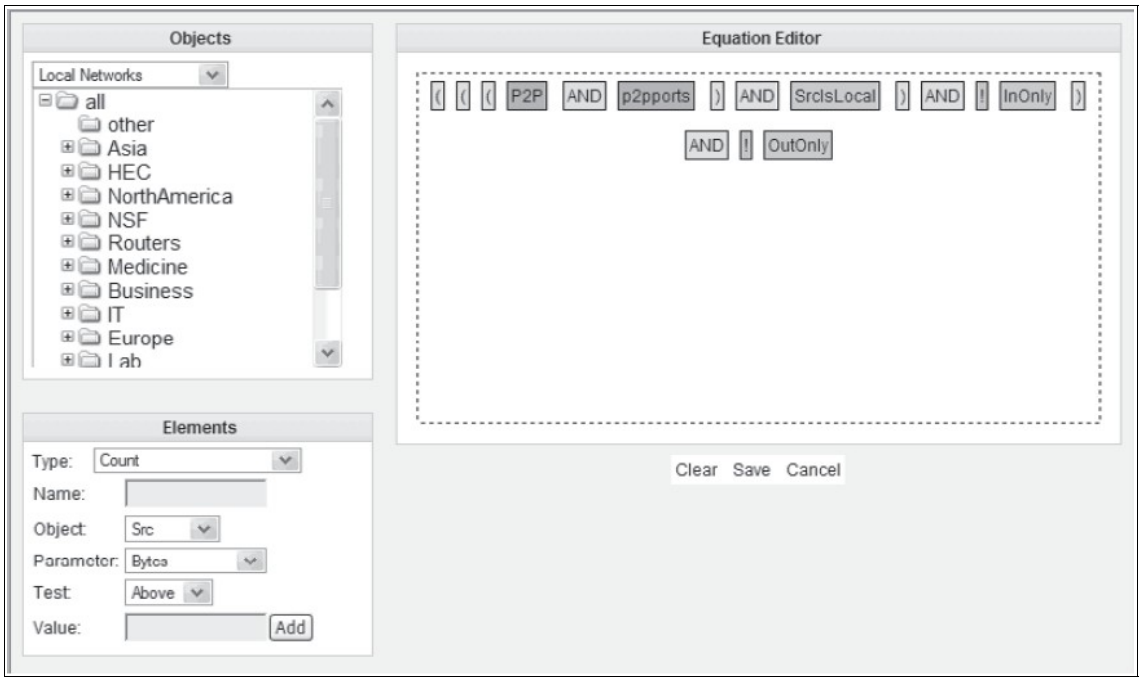


Figura 28 - Editor de ecuaciones de QRadar [MiHaVa11]

En la Figura 28 se muestra el editor de ecuaciones. Allí se destacan tres áreas:

- La sección objetos, donde se proporcionan las categorías de dispositivos fuente según su área de uso.
- La sección elementos donde se incluyen parámetros específicos del dispositivo como ser IP, puerto, tipos de flujo deseados, etc.
- La sección donde se construye propiamente la ecuación en base a funciones lógicas y los elementos previamente definidos.

Si bien las Views ofrecerán una idea de lo que está ocurriendo en la red, es necesario tomar en cuenta su impacto a nivel de procesamiento de la CPU. Por esto se recomienda desactivar las Views por defecto que no sean adecuadas al entorno de una organización determinada. Así mismo se recomienda generar los objetos que

componen las ecuaciones lo más generales posibles, a fin de incluir la mayor cantidad de elementos posibles y reducir la cantidad de entradas a los procesos.

Este último punto repercutirá negativamente en el nivel de detalle logrado por las Views. Siendo un compromiso entre la visibilidad necesaria y el consumo de recursos. [MiHaVa11] [10]

V.4.8 – Sentries de QRadar

Los Sentries o Centinelas son filtros lógicos que procesan cada flujo de datos que pasa por ellos. Son utilizados para identificar patrones de tráfico particulares y generar eventos a partir de éstos. Estos eventos a su vez son enviados al Event Processor para su posterior análisis y correlación con otros flujos y eventos.

Cada flujo de entrada a los Sentries pasará por una Sentry list donde se comparan los patrones del flujo contra lo definido en el o los Sentries. Cuando se produce una coincidencia, el Sentry producirá un evento.

Al igual que para las Views, es recomendable deshabilitar los Sentries que no apliquen a los requerimientos de la organización. [MiHaVa11]

QRadar ofrece cuatro tipos de Sentries:

- **Security/Policy**: Monitorea el tráfico en busca de violaciones de seguridad o de políticas en general como ser usos no permitidos de aplicaciones.
- **Comportamiento**: Estos Sentries monitorean su entorno en busca de cambios de comportamiento.
- **Anomalía**: Monitorea su entorno en busca de actividades anormales.
- **Umbrales**: Monitorea el entorno de la red en busca de actividad que supere los umbrales establecidos. [SNEH11]

V.4.9 – Reglas en QRadar

QRadar define como una “Rule” a la colección de condiciones que, de producirse, generan como consecuencia una o más acciones. Es posible configurar QRadar a fin de que responda a una secuencia específica de eventos o patrones. Estos eventos pueden ser enviar una notificación al módulo Offense, el cual puede disparar acciones contra sistemas externos, mediante un mensaje de Syslog, un trap de SNMP o el envío de un email. Como siempre ocurre, es posible extender las reglas existentes con aquellas definidas específicamente para el entorno donde se haya desplegado.

V.4.10 – The Offense Manager

Este módulo se licencia en forma separada, y permite una visualización más en detalle de los Offense que han sido detectados por QRadar. De esta forma, es posible navegar en detalle dentro de los eventos, a fin de obtener información más específica en contraposición a la compactada obtenida por defecto.

[10] El límite máximo para cada View es de 200 hosts. [MiHaVa11]

V.4.11 – DSMs de QRadar

Los DSM (Device Support Module) son los componentes de QRadar encargados de entenderse con los dispositivos generadores de eventos a fin de integrarlos efectivamente con QRadar. También se encargarán de analizar los eventos recibidos así como de almacenarlos de forma ordenada en la base de datos. Por tanto, se alinean con las funciones de colección y normalización de eventos de nuestro modelo genérico de SIEM.

En caso que QRadar DSM no pueda analizar o normalizar correctamente los eventos en crudo que recibe, es posible configurar manualmente el proceso de normalización marcando qué atributos de dichos eventos deben ser mapeados al formato QRadar. Utilizando la herramienta event-mapping es posible mapear los eventos de dispositivos desconocidos a categorías conocidas por QRadar de forma de poder clasificarlos y correlacionarlos adecuadamente.

V.4.12 – Análisis de Eventos en QRadar

La Figura 29 muestra una consola personalizada para la vigilancia de eventos, flujos y alertas relevantes en la red de una organización.

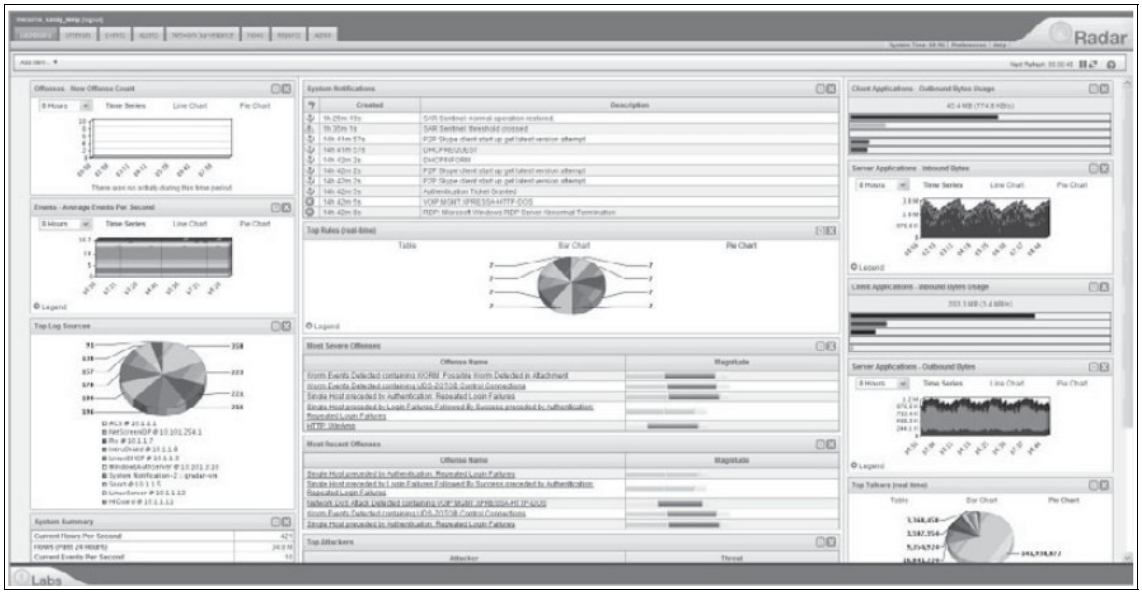


Figura 29 - Consola de eventos de QRadar [MiHaVa11]

Dentro de la misma se despliega una variedad de información referente a lo que está ocurriendo en tiempo real así como vistas históricas. Se cuenta con opciones de búsqueda, categorías y filtros para la visualización.

Es éste el componente que cumple las funciones de “Visualización y Monitoreo” para lo que es nuestro modelo de SIEM genérico.

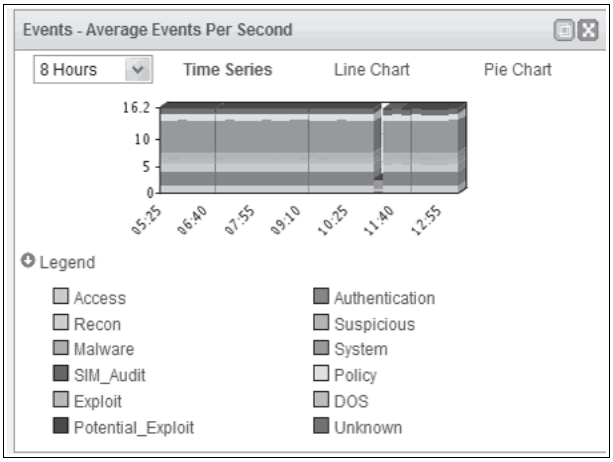


Figura 30 - Eventos por segundo promedio de QRadar [MiHaVa11]

Como podemos ver en la Figura 30, se cuenta con diversas categorías de eventos, sus cantidades y momentos en que han ocurrido en las últimas ocho horas. Incluso se puede ver aquellos catalogados como *Sospechosos* y *Desconocidos.* Para una mejor comprensión de su incidencia, la consola permite el despliegue de estos valores y porcentajes en diversos formatos.

Debido a las capacidades de QRadar para normalizar el flujo de eventos en la consola de forma rápida, se puede limitar la visualización de sólo aquellos eventos importantes relacionados con una investigación específica y en tiempo real. QRadar ofrece también el manejo de magnitudes para los eventos a fin de permitir profundizar y afinar la visión de aquellos eventos más graves dentro de una categoría.

Una vez identificada la actividad reportada como sospechosa o desconocida, y más allá de las medidas adoptadas para su contención, el administrador podrá construir reglas basándose en asistentes (wizards) para que en un futuro, ante una situación similar, el sistema se comporte según su criterio, ya sea categorizándola acorde, notificando y/o disparando una alerta.

V.5 – Alien Vault OSSIM

V.5.1 – Introducción

OSSIM es un SIEM de tipo OSS disponible para ser descargado, instalado, modificado y operado. Es desarrollado por AlienVault, empresa que, a su vez, cuenta con una versión comercial. Como es de esperar, al existir una versión comercial, la versión gratuita tiene limitantes de performance, soporte y capacidad de almacenamiento.

De hecho, debido a éste último aspecto, la propia AlienVault no considera a OSSIM como un SIEM completo. Sin embargo la versión OSS brinda un buen acercamiento al producto para aquellos interesados en instalar un SIEM en su organización.

Hemos de decir que es una solución en constante evolución y que ha avanzado considerablemente desde sus comienzos en 2003, procurando ser un proyecto muy ambicioso[11]. A pesar de esto, la documentación existente es escasa.

El concepto base detrás de OSSIM es "no re-inventar la rueda". Para esto, utiliza varias herramientas probadas y ampliamente difundidas de código abierto, de forma de aprovechar en cada una sus aspectos más destacados. A este conjunto AlienVault le suma por su parte un motor de correlación, elaboración de informes y herramientas para la gestión centralizada.

Podemos destacar, en general, la flexibilidad de configuración brindada a los usuarios para adaptar esta herramienta a sus necesidades específicas, así como la integración de diversas herramientas OSS para trabajar en conjunto.

A lo largo de las siguientes secciones presentaremos las herramientas que esta solución incorpora, veremos aspectos de su arquitectura, sus sensores y la forma en que se colecta la información, entre otros temas. Finalmente, presentaremos las diferencias entre la solución libre y la comercial.

V.5.2 – Herramientas utilizadas por OSSIM

Las ventajas en la utilización de herramientas de código abierto, además de la reducción de costos, es la de aprovechar el talento de miles de programadores que trabajan en mejorar y desarrollar aplicaciones para las mismas. OSSIM utiliza más de quince herramientas destacadas de código abierto, entre las cuales resaltamos: Snort, OpenVAS, Ntop, Nagios, PADS, P0f, OCS-NG, OSSEC, OSVDB, NFSen/NFDump e Inprotect. Entre ellas cubren las funciones de: HIDS, scanner de vulnerabilidades, monitoreo de tráfico, monitoreo de dispositivos, inventario de activos, entre otras. [SNEH11]

V.5.3 – Aspectos y tareas de OSSIM

La manera en que todas estas herramientas trabajan en conjunto, de forma de lograr un resultado que sea mejor que una suma de aportes individuales, es lo que se describe a continuación.

En la Figura 31 se detallan las tareas realizadas por OSSIM en base a la sinergia de sus componentes. Su explicación se dará en los párrafos siguientes.

Detectores: AlienVault define un detector como cualquier programa que escuche en la red, monitoree archivos o logs en busca de señales de ataques, y emita alertas en consecuencia. Básicamente tendremos detectores basados en patrones o basados en anomalías[12].

- ***Detectores basados en patrones:*** OSSIM trae incluido un detector del tipo IDS/HIDS basado en patrones (o firmas). Sin embargo el verdadero poder de OSSIM está dado por su capacidad de interactuar con una infinidad de

[11] Al momento de escribir este capítulo, el fabricante anunciaba el próximo lanzamiento de una renovada versión: AlienVault Unified Security Management Platform (AV-USM™)

[12] Estos conceptos son tratados en la sección IDS/IPS del documento "Estado del Arte en Redes de Sensores de Seguridad Informática" [SNEA11].

dispositivos externos, tanto comerciales como libres. La compatibilidad en este proceso esta dada a través de un agente para el cual es posible desarrollar plugins a medida.

- ***Detectores basados en anomalías:*** Distintos detectores basados en anomalías son incorporados en OSSIM. Un ejemplo, es la capacidad de Ntop de indicar las medias y desviaciones de tráfico en función de la historia anterior.

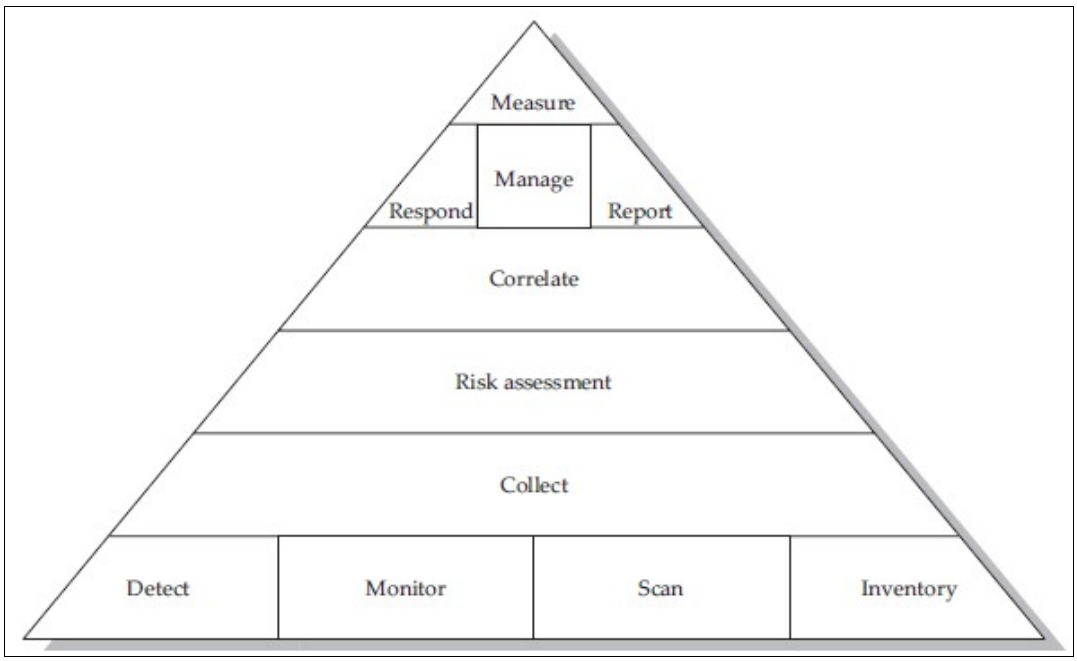


Figura 31 - Diagrama de tareas que se realizan en OSSIM [MiHaVa11]

Monitores: Los Monitores son utilizados por OSSIM para proveerse de una perspectiva del tráfico, detectar rápidamente cambios en la red y brindar información al sistema de correlación. Existen tres tipos de monitores: de red, de disponibilidad, y personalizados.

- ***Monitores de red***: Basado en la herramienta Ntop, su función es la construcción de perfiles de uso y análisis de sesiones. Se generan así tres tipos de monitoreo:
  - Información del uso de la red, como ser el número de bytes transmitidos en un período de tiempo.
  - Información de la actividad de un servicio, como ser estadísticas de servicios pop, http, smtp, ssh, etc.
  - Monitoreo de sesiones en tiempo real, los cuales proveen información actualizada sobre las sesiones activas, como ser cuáles hosts participan y qué puertos utilizan. En este sentido la herramienta Ntop provee información a través de “sniffer” o mediante la importación de información desde otros dispositivos como los routers.

- ***Monitores de disponibilidad***: Es utilizado para detectar denegaciones de servicio u otras interrupciones en la red. Basado en la herramienta Nagios, interactúa con un plugin a fin de incluir estos datos en el proceso de correlación y presentarla en caso que corresponda.

- ***Monitores personalizados***: Se refiere a los plugins realizados a medida, a fin de detectar cualquier cosa que se desee y actuar en consecuencia. Por ejemplo, disparar un escaneo de vulnerabilidades en un host que se considere comprometido y tomar medidas especiales en base al resultado.

Scanners: Los scanners tienen por función simular ataques y determinar si un dispositivo de red es vulnerable a un ataque en particular. Estas exploraciones de vulnerabilidades también colaborarán en el proceso de correlación. En particular la herramienta OpenVAS es la utilizada para esta tarea.

Inventario: OSSIM incorpora herramientas basadas en agentes, o sin ellos, para la recolección automática de información de inventario. También brinda la posibilidad de insertar manualmente datos de inventario en la base de datos.

Colector: El propósito de la infraestructura de recolección es la de capturar y normalizar toda la información de seguridad proveniente de distintos dispositivos, así como el envío de la misma al servidor para su posterior procesamiento. Para obtener dicha información desde los dispositivos fuente el colector admite los mecanismos de push y pull, permitiendo la priorización y políticas de recolección.

En particular el sistema puede colectar eventos utilizando cualquiera de los siguientes protocolos: Syslog y Syslog-ng, SNMPv2 y SNMPv3, HTTP, SQL, ODBC, WMI, FTP, SFTP, Socket Unix, Plain log, SSH, Rsync, Samba, NFS, SDEE, RDEP, OPSEC, y CPMI.

- ***Push:*** En este escenario los datos son enviados en el formato nativo del dispositivo. Es comúnmente utilizado en entornos donde se elije a Syslog o SNMP como protocolos para la transmisión de información, o cuando es complicado instalar agentes en los dispositivos a sensar.

- ***Pull:*** Los agentes instalados en los dispositivos fuente se encargan de colectar la información del dispositivo en su formato particular, y enviarla a su agente servidor.

- ***Priorización:*** Los logs que arriban al servidor son normalizados según los niveles de prioridad a la escala estándar de AlienVault, entre 0 y 5. Sin embargo, el administrador puede ajustar dichos valores por defecto a través de una tabla de normalización y políticas de priorización. Por ejemplo, puede asignarse la máxima prioridad a los eventos internos o a los externos.

- ***Políticas de recolección:*** Es posible establecer políticas de recolección como ser prioridades, así como fijar niveles de importancia al momento de procesar la información proveniente de los sensores. El objetivo es filtrar y consolidar los

eventos antes de ser enviados al servidor de OSSIM. Esta técnica permite además al administrador regular el flujo de eventos. [MiHaVa11]

Evaluación de Riesgos: Se trata de un proceso de medición del riesgo mediante el cual se procura determinar aquellos sistemas relevantes. Mediante esta tarea OSSIM se posiciona como un ayudante para la toma de decisiones. El sistema calcula un valor de riesgo para cada evento, donde dicho cálculo se basa en los siguientes tres parámetros:

- valor del activo (costo de que el activo sea comprometido);
- amenaza representada por el evento (cuánto daño puede ser hecho al activo);
- probabilidad de que el evento ocurra.

Los creadores de OSSIM utilizan la siguiente definición para describir el cálculo de riesgo: *Una medida del impacto potencial de una amenaza sobre un activo, dada la probabilidad de que éste ocurra.*

Al posicionar el SIEM en la red con una vista única de los activos, las amenazas y el tráfico de red (para ayudar a determinar la probabilidad de un ataque), será esta vista global la que permitirá al SIEM asignar en tiempo real valores de riesgo a un evento que básicamente haya ocurrido o esté ocurriendo, sea un falso positivo o tan sólo un evento menor.

Correlación: Tendiente a reducir los falsos positivos (falsas alarmas) y evitar los falsos negativos (instancias en que las intrusiones pasan desapercibidas), OSSIM junto con su motor de correlación está orientado a proporcionar un "contexto del ataque". Para ello, el sistema considera las siguientes cinco variables: las alertas, las vulnerabilidades, el inventario, las anomalías y la red.
En este sentido OSSIM realiza tres tipos de correlación:

- ***Correlación Lógica:*** se lleva a cabo a través de un conjunto de reglas predefinidas y personalizables que ejecutan lógica booleana en cualquier número de condiciones de un evento, generándose un árbol de condiciones sobre la base de éstas reglas. El evento se prueba contra este árbol de condiciones y su prioridad se incrementa a medida que se cumplan las condiciones. Eventos adicionales podrán ser generados por el motor de correlación y recursivamente ser añadidos al proceso de correlación.

- ***Correlación de Inventario:*** ocurre cuando las características propias de un activo se miden frente a una amenaza particular.

- ***Correlación Cruzada:*** realiza un control cruzado entre los datos del IDS y los datos de vulnerabilidades, permitiendo priorizar eventos en función de si un activo es vulnerable o no.

Responder: OSSIM es capaz de responder automáticamente a un evento determinado o a un conjunto de eventos. Las respuestas incluyen el envío de un correo electrónico o el envío de una directiva para cambios en la red, tales como el ajuste de un firewall o configuración en un switch.

Gestionar: Una vez que un ataque es detectado, colectado, valorado su riesgo, correlacionado y validado por el analista como algo real, se genera un ticket para realizar un seguimiento del incidente hasta su resolución. Estos tickets pueden ser generados desde varios lugares del sistema: panel de alarma, consola forense o paneles de riesgo; y cada entrada contiene información sobre el propietario del incidente, los eventos contenidos en el incidente, el estado actual de los hechos y la historia del incidente. El ticket se almacena en una base de datos permitiendo búsquedas futuras, análisis de tendencias, etc.

Reportar: OSSIM contiene un robusto motor de informes con diversos formatos para los mismos, y que además permite personalizar y crear informes con fines específicos.

Medir: Se proporcionan distintos paneles que visualmente presentan los datos de una manera cómoda al operador, ofreciendo diversos puntos de vista en los diferentes niveles de la organización.

V.5.4 – Arquitectura de OSSIM

Básicamente, la arquitectura de OSSIM puede condensarse en cuatro componentes básicos:

- Sensores
- Servidor de gestión
- Base de datos
- Interfaz de visualización

Sensores:

Los sensores son los componentes de más bajo nivel, sirviendo como interfaz entre los dispositivos fuente y el servidor de gestión.

En éstos se engloban las funcionalidades de Detección, Monitoreo, Escaneo e Inventario descritos en la sección anterior. Así, en el modelo OSSIM, los sensores combinan las capacidades de un agente colector con las de un detector y monitor. Como agente, utiliza plugins para analizar el tráfico proveniente de otros dispositivos de seguridad y enviarlo al servidor de gestión.

Sumado a esto, y como ya mencionáramos, un sensor puede servir como IDS (detectando en base a firmas o anomalías); servir como un scanner de vulnerabilidades e incluso realizar monitoreos del desempeño de la red en la que se encuentra. [MiHaVa11]

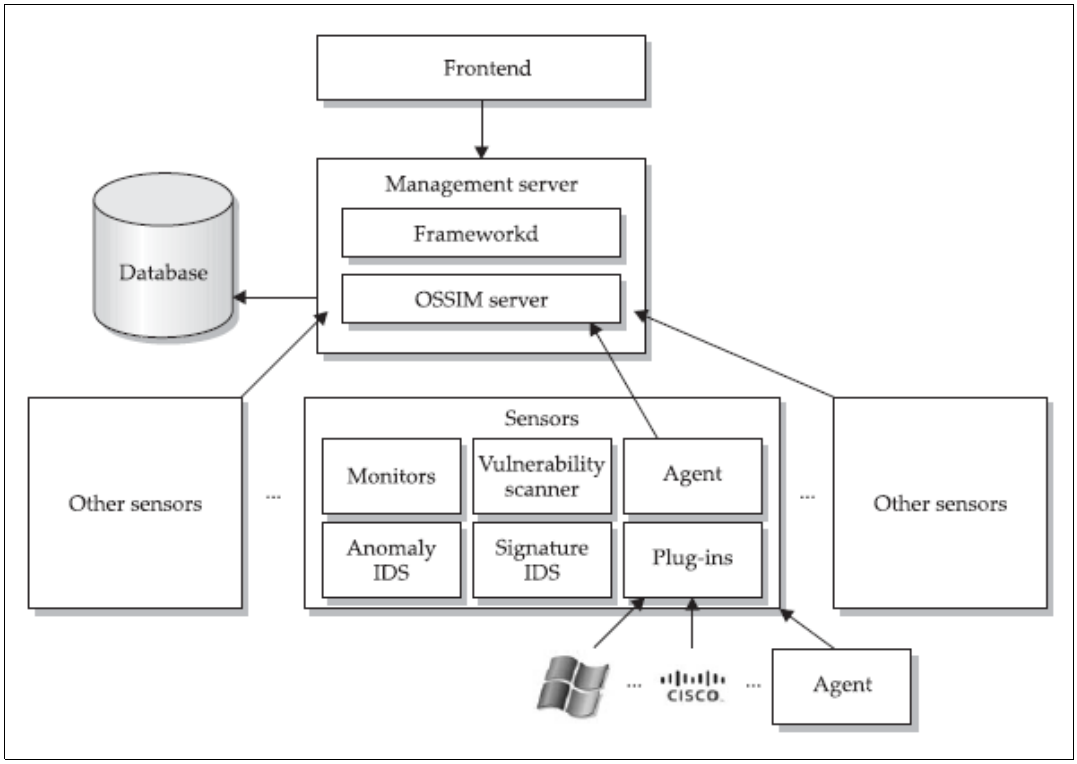


Figura 32 - Arquitectura de OSSIM [MiHaVa11]

Servidor de Gestión:

El servidor de gestión incluye los siguientes dos componentes:

- *Frameworkd*, demonio que controla otros componentes como los sensores;
- *OSSIM-server*, en esencia es quien procesa los eventos recibidos desde los sensores.

En la arquitectura OSSIM, el servidor de gestión será el responsable de la normalización, la recolección, la priorización, la correlación, y la evaluación del riesgo. Además, realiza otras funciones como ser: copias de seguridad, inventario, gestión de tickets y procesos y tareas programadas. Como vemos, centraliza en un conjunto de piezas de software la mayor parte de los aspectos del modelo de SIEM genérico que hemos presentado.

Base de datos:

La base de datos que utiliza el modelo es una de tipo SQL (MySQL) y será la encargada de almacenar toda la información necesaria para que OSSIM pueda funcionar.

Debemos mencionar que OSSIM en su versión libre sólo almacena aquellos datos necesarios para la correlación en tiempo real y el análisis forense. Para el almacenamiento a largo plazo, debe ser usada la versión comercial del producto. [MiHaVa11]

Interfaz de visualización (dashboard):

Es la "consola" en nuestro modelo de SIEM genérico y la que proporciona una interfaz de usuario al sistema. La misma es de tipo web-based, accesible mediante

cualquier navegador web, y el gran aspecto a destacar de la misma es la forma en que es presentada, integrando todas las herramientas que la solución dispone.

Figura 33 - Interfaz de visualización: distintas capturas del dashboard [ASSD10]

En esta consola se puede visualizar, además de todas las herramientas disponibles, el estado de sensores y alertas, efectuar configuraciones y ajustes de la plataforma y generar diversos reportes. Vale mencionar que las funcionalidades disponibles en la consola de la versión comercial, tanto para reportes como para gestión de la plataforma, superan ampliamente a las de la versión libre.

En la Figura 33 se presentan distintas capturas de la misma. [SNEH11]

V.5.5 – Flujo de datos en OSSIM

Para entender la integración de cada uno de los productos haremos un recorrido del flujo de datos desde la generación de un evento hasta su despliegue en la consola de mando:

1. Los eventos son procesados por los detectores. En caso de que éstos detecten un patrón de ataque o anomalía generarán una alerta.
2. Las alertas son procesadas, en caso de ser necesario, por los consolidadores antes de ser enviadas. Estos se encargarán de agrupar la información enviada a fin de ocupar el mínimo ancho de banda.
3. Las alertas son recibidas por el colector a través de diferentes protocolos abiertos de comunicación.
4. Se realiza el “parsing” de los datos recibidos, cumpliendose a su vez varias funciones:

- normalizar de los eventos;
- guardar dicha información, clasificándola por prioridad según la política de seguridad definida y los datos del sistema atacado;
- valorar el riesgo que implica la alerta y de ser necesario envía una alerta al cuadro de mandos.

5. Las alertas cualificadas son enviadas a cada uno de los procesos de correlación que actualizarán sus variables de estado y eventualmente lanzarán nuevas alertas con una información más completa o consolidada. Estas alertas son enviadas de nuevo al parser para su almacenamiento, priorización, valoración del riesgo, entre otras acciones.
6. El monitor de riesgos visualizará periódicamente la situación de cada uno de los índices de riesgo según han sido calculados por CALM[13].
7. El cuadro de mandos mostrará las alarmas recientes, actualizará el estado de cada uno de los índices, los comparará respecto de los umbrales y lanzará nuevas alarmas o realizará las acciones correspondientes en caso de ser necesario.
8. El administrador podrá desde el cuadro de mandos enlazar y visualizar a través de la consola forense todos los eventos ocurridos en el momento de la alerta. Podrá además comprobar el estado actual de los dispositivos a través de los monitores de uso, perfiles y sesiones. [KaMu03]

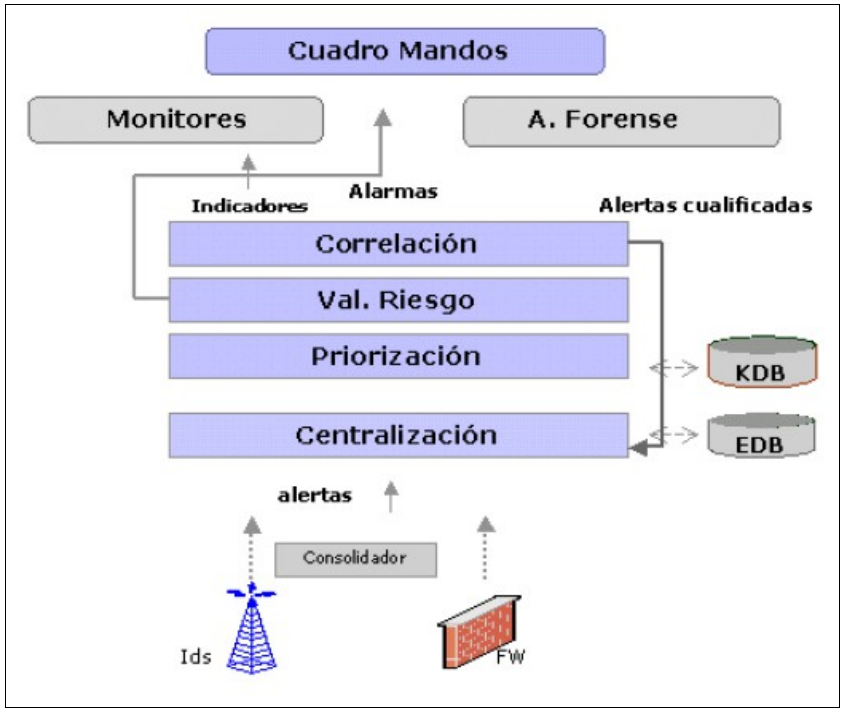


Figura 34 - Flujo de datos en OSSIM [KaMu03][14] [15]

[13] CALM (Compromise and Attack Level Monitor) es un algoritmo de valoración por acumulación de eventos con recuperación en el tiempo. Recibe como entrada un alto volumen de eventos y como salida un único indicador del estado general.

[14] KDB, la base de datos del Framework, en la cual parametrizaremos el sistema para que conozca nuestra red y definiremos nuestra política de seguridad. [KaMu03]

V.5.6 - Comparación entre AlienVault Professional SIEM y AlienVault Open Source SIEM (OSSIM)

	Open Source	Professional SIEM
Support	Community	7x24
Quality Assurance	Community	Professional Q&A
Security	Not audited	Audited
Performance	Moderate	30 x Open Source, Assured
SIEM Intelligence	Logical Correlation Simple Taxonomy	Cross Correlation Rich Taxonomy
Logger	N/A	Unlimited Forensic Storage
Reports	< 25 + Jasper	> 200 + Web Wizard
Scalability/HA	N/A	HA, Distributed ,Multitenant, Unlimited
Compliance	High Level Reports	High and Low Taxonomy-based
Updates	None	Daily rules and reports
User Management	Individual, simple controls	Templates and Granular Controls

Figura 35 - Comparativa de versiones libres y pagas de AlienVault [AVPSvOS10]

Si bien la versión libre puede ser utilizada durante el tiempo que se desee permitiendo en todo momento migrar a la versión profesional, existen algunas diferencias de base entre ambos productos. El cuadro de la Figura 35 muestra un resumen de dichas diferencias.

Básicamente las dos versiones proporcionan las funciones generales como ser: detectar amenazas dentro de la red, recoger y almacenar eventos y otras informaciones (ej inventario) de la red, correlacionar la información antes mencionada, generar informes y administrar la infraestructura de seguridad de la red. [AVPSvOS10]

Entre las diferencias mencionadas destacamos:

- **Seguridad**: Mientras que la versión Professional permite establecer comunicaciones cifradas entre sus componentes distribuidos así como con la mayoría de los dispositivos fuente; OSSIM no es compatible con tales comunicaciones.

- **Performance**: Cada componente de la versión Professional ofrece treinta veces más rendimiento que la versión libre. En particular, cada sensor (colector) de la primera permite gestionar más de 5000 eventos por segundo, mientras que la performance de OSSIM no esta asegurada.

[15] EDB, la base de datos de eventos, la más voluminosa pues alojará todos los eventos individuales recibidos de nuestros detectores. [KaMu03]

- **Correlación Cruzada**: AlienVault Professional puede llevar a cabo tanto una correlación lógica como una correlación cruzada, integrando información de inventario con la de IDS, aumentando así las capacidades de detección de ataques. OSSIM no ofrece la capacidad de correlación cruzada.

- **Logger**: Disponible sólo para la versión comercial, esta compuesto por una familia de dispositivos dedicados a proporcionar un transporte seguro y autenticado así como capacidad de almacenamiento para la información y los eventos. De esta forma los eventos pueden cifrarse, asegurando su integridad de extremo a extremo, y manteniendo su valor forense y legal.

- **Entornos distribuidos:** AlienVault Professional permite desarrollar una solución totalmente distribuida, permitiendo incluso el desarrollo de soluciones de seguridad para un MISP (Management ISP). OSSIM no cuenta con dicho soporte, permitiendo tan sólo algunos elementos y con escasa documentación.

En resumen y como hemos visto, AlienVault OSSIM cuenta con un compendio de herramientas OSS interactuando adaptada y conjuntamente. Es, a partir de esta sinergia, el bajo o nulo coste y la posibilidad de migrar a su versión comercial lo que la hace una herramienta de consideración. Sin embargo, encontramos que las diferencias entre la versión libre y la comercial son relevantes, particularmente en los aspectos de performance, escalabilidad y distribución.

V.6 – Prelude-IDS

V.6.1 – Introducción

Al igual que otras herramientas libres OSS, Prelude-IDS[16] cuenta con la contribución de múltiples profesionales en seguridad alrededor del mundo. Sumando lo anterior a sus capacidades para colectar, normalizar, clasificar y correlacionar eventos de seguridad, independientemente de la marca y licencias de los dispositivos, convierte a Prelude-IDS en un sistema SIEM universal. [Yas09]

V.6.2 – Evaluación preliminar de Prelude-IDS

Cumplimiento de Regulaciones: Prelude-IDS aporta:
- una normalización de todos los eventos del sistema a un formato común;
- la centralización en el almacenamiento de logs;
- el mantenimiento de una pista forense.

La normalización de todos los eventos se realiza al formato IDMEF. El objetivo es fijar el formato y los procedimientos para el intercambio de los datos de interés, como ser la detección de intrusiones, los referentes a sistemas de respuesta y de gestión que deban interactuar con Prelude-IDS.

[16] CS (www.c-s.fr), provee "Prelude Pro" como producto comercial y, por tanto, del soporte no comunitario.

El formato IDMEF utiliza un modelo de base de datos XML el cual define un estándar para la representación de las alertas. [SNEA11] Esto permite a su vez una mayor interoperabilidad entre diferentes dispositivos. Además, la normalización permite que los eventos de los diferentes dispositivos puedan ser almacenados en un formato estructurado único.

Esto no es distinto a lo realizado por otras soluciones, pero en este caso el formato es abierto y estándar; todas las otras soluciones lo hacen en un formato propietario y cerrado.

Compatibilidad Universal: Prelude-IDS es capaz de interoperar con casi cualquier dispositivo de red puesto que ofrece entornos de trabajo en lenguajes tales como C, C++, Python, Ruby, LUA, y Perl. Esto asegura que las aplicaciones de seguridad existentes pueden ser convertidas al formato nativo de Prelude-IDS. [Yas09]

Una vez instalado Prelude-IDS existen tres opciones para interoperar con los dispositivos existentes:

- Que los dispositivos instalados envíen sus logs al Prelude-LML, haciendo que el mismo realice la función de “parsing”.
- Establecer enlaces directos entre Prelude-IDS y los dispositivos existentes que sean compatibles en forma nativa, lo cual implica la utilización de libprelude como API de integración.
- Contactar al personal de Prelude-IDS para evaluar la compatibilidad de los dispositivos no soportados nativamente.

Un listado de las aplicaciones que son soportadas por Prelude-IDS en forma nativa pueden encontrarse en el documento referenciado como [SNEH11]. Entre ellas se encuentran: Nepenthes, OSSEC, y Snort.

En adición, Prelude-IDS puede monitorear cualquier tipo de archivo de log que proceda de diversos dispositivos y sistemas como ser: Firewalls, Routers, IDS, Antivirus, Database, Honeypots, etc. [SNEH11]

Para el caso de los dispositivos no soportados nativamente, pero que cuenten con la opción del reporte de sus eventos mediante Syslog, estos podrán ser igualmente supervisados. [Yas09] Esta integración se realiza mediante el componente Prelude-LML, el cual se encuentra por defecto configurado para integrarse a un Syslog Server.

Monitoreo de eventos en tiempo real: Prelude-IDS permite la visualización de los eventos en tiempo real mediante la interfaz web Prewikka. Dicha visualización se logra no sólo por la elección de una vista particular, sino porque la propia Prewikka permite actualizar en forma automática la misma, siendo ideal para una consola en el NOC/SOC (Network Operation Center / Security Operation Center).

Reducción en los costos de seguridad: Dado que Prelude-IDS es una herramienta de código abierto, no existen costos asociados a derechos de licencia ni límites en la cantidad de dispositivos que puede supervisar. Sin embargo, los costos asociados a implantar una solución de ésta naturaleza (OSS) se deberán a la configuración personalizada de la misma y a su mantenimiento posterior.

V.6.3 – Arquitectura de Prelude-IDS

Como ya mencionáramos, la arquitectura de Prelude-IDS centraliza el reporte de eventos de todos los dispositivos sensados. Dicho punto es el Prelude-Manager, al cual cada sensor se conectará de forma segura utilizando los protocolos SSL/TLS.

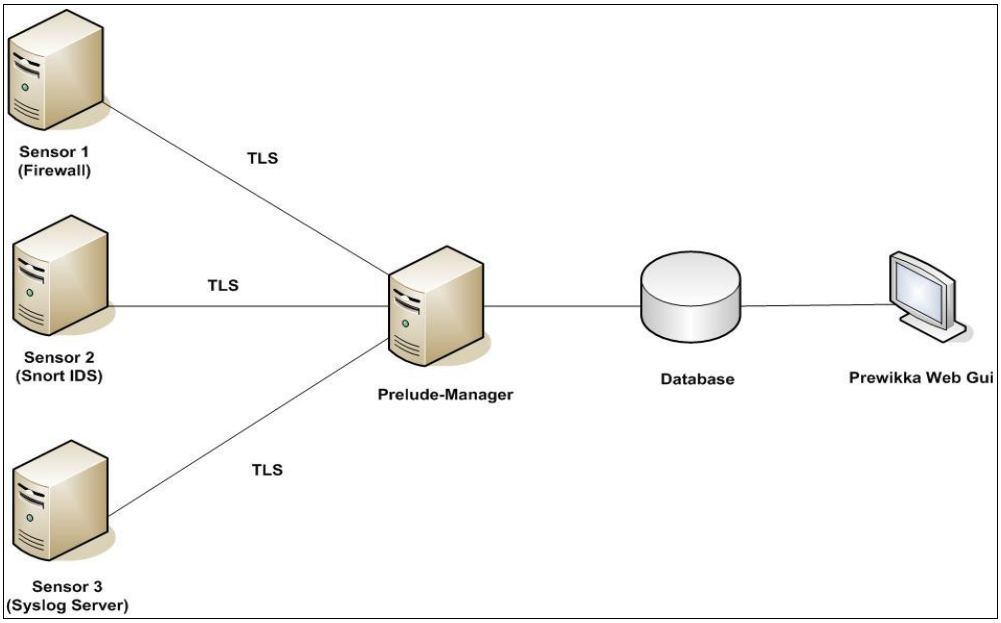


Figura 36 - Diagrama básico de la arquitectura de Prelude-IDS [Yas09]

La Figura 36 muestra un ejemplo de instalación simple de Prelude-IDS, con un único Prelude-Manager y tres sensores. En esta arquitectura, al igual que en otras ya vistas, se considera como sensor a todo dispositivo que reporte sus logs al Management Server (Prelude-Manager). Prelude-IDS considera como sensores a los IDSs, Firewalls, Syslog-Server, Hosts, entre otros.

Como mencionáramos, es posible recibir los logs en su formato nativo desde dispositivos compatibles directamente conectados al Prelude-LML o a través de la utilización de frameworks que permiten desarrollar compatibilidad con otros dispositivos no soportados nativamente. Aquí nos referimos al soporte de interpretador en el Prelude-LML, no a la utilización de libprelude como API de integración.

Otro aspecto importante es que Prelude-IDS admite arquitecturas distribuidas, incluso a través de enlaces tipo WAN. En estos casos, cada área adicional deberá contar con un Prelude-Manager, los cuales retransmitirán sus eventos a un servidor central. Si lo comparamos con OSSIM (la otra opción OSS analizada en el presente documento), esta arquitectura en Prelude-IDS no requiere licenciamiento adicional alguno. En el caso del OSSIM sólo es posible alcanzar este esquema usando la versión comercial, AlienVault Professional.

Con el fin de asegurar la redundancia de la información, cada área guardará su información en una base de datos local, además de enviarla al servidor central. Esto evita, entre otros problemas, los presentados por fallas en el sistema de comunicación. Un ejemplo de arquitectura distribuida se presenta en la Figura 37.

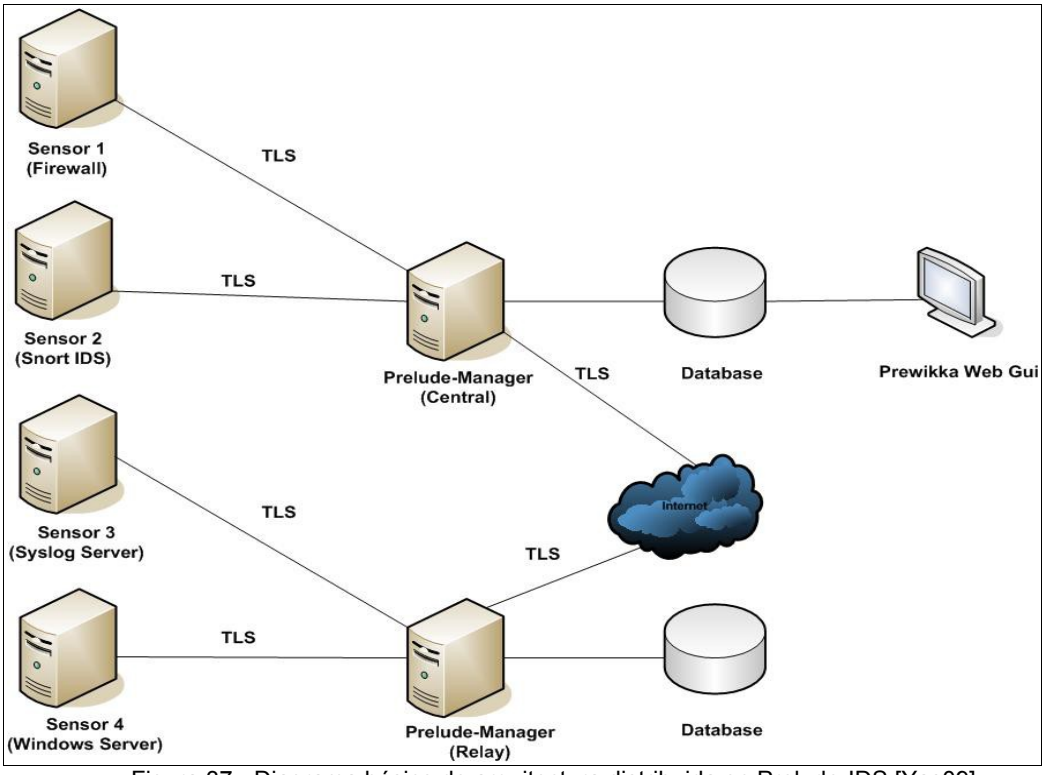


Figura 37 - Diagrama básico de arquitectura distribuida en Prelude-IDS [Yas09]

Finalmente, instalando Prewikka en los Prelude-Manager locales, se brinda la posibilidad al personal de seguridad informática de conocer los eventos actuales referentes a su área. [Yas09]

V.6.3.1 – Sistemas de Comunicación en Prelude

Es claro que uno de los puntos claves en cualquier arquitectura de intercambio de información, es la elección/definición de la mensajería utilizada.

En el caso de las herramientas comerciales, el intercambio en forma “abierta” se da en la primera etapa, en lo que se refiere a la recepción de mensajes desde los dispositivos fuentes. Sin embargo, los intercambios subsecuentes, los cuales se realizan entre los componentes de la propia solución, se efectúan mediante protocolos propietarios para cada una de las mismas.

Esto no debe confundirse con la disposición de elementos “custom” que permiten desarrollar interpretadores para mensajería no soportada en forma nativa. Lo que aquí se indica es que no existe una API (Application Programming Interface) que permita realizar la comunicación directamente con los componentes de la solución.

En el caso de Prelude-IDS, se siguió un camino diferente. Si bien existe la posibilidad de desarrollar interpretadores para el procesamiento de fuentes no soportadas en forma nativa, el carácter “abierto” al tratarse de un proyecto OSS, también impactó en la opción de la mensajería utilizada.

Se hizo la opción por un formato de mensajería abierta como es el caso del IDMEF. Este formato de mensajería es el resultado del trabajo desarrollado por el grupo de IDWG de la IETF. Esto no sólo implica que la mensajería tiene un formato no

propietario, sino que se permite su integración en forma más profunda. En particular, el proyecto Prelude-IDS desarrolló una biblioteca, libprelude, la cual constituye una API que permite la comunicación directa a los componentes de la arquitectura.

La biblioteca libprelude, no sólo resuelve la mensajería a utilizar, sino que realiza una implementación de transporte sobre SSL que garantiza la confidencialidad de la misma y los controles de acceso adecuados para los componentes. En este sentido, Prelude-IDS entrega una opción adicional que no se encuentra en las otros soluciones estudiadas.

El impacto de esta elección no es sólo positivo desde el punto de vista conceptual, sino también desde un punto de vista práctico. El proyecto en sus más de diez años de vida, ha logrado la incorporación de su librería en proyectos OSS como Snort-IDS y Nepenthes, entre otros.

Otro aspecto que también tiene referencia en la elección de una mensajería abierta, es la posibilidad de lograr la integración en forma estándar de diferentes soluciones.

V.6.4 – Componentes en la Arquitectura de Prelude-IDS

Prelude-IDS basa su arquitectura en siete componentes: Prelude-Manager, Libprelude, LibpreludeDB, Prelude-LML, Prelude-Correlator, Prewikka Interface, y Prelude-PFLogger. Sin embargo, sólo cuatro de éstos (Prelude-Manager, Libprelude, LibpreludeDB, y Prewikka) son indispensables para el funcionamiento del sistema base. Los otros componentes se limitan a brindar funcionalidades adicionales e interoperabilidad. [Yas09]

V.6.4.1 – Prelude-Manager

Prelude-Manager es el servidor central, el encargado de la gestión. Este aceptará todas las conexiones, en forma segura, ya sean provenientes de sensores o de otros managers, siendo compatible tanto con IPv4 como con IPv6. En este sentido, el Prelude-Manager tiene además a su cargo el procesamiento de dichos eventos y, si fuera necesario, su conversión al formato binario IDMEF[17].

Si intentamos alinear el Prelude-Manager con los componentes de nuestro SIEM genérico, difícilmente lo logremos.

En la arquitectura genérica no definimos un componente cuya función es controlar la conexión subyacente. Siempre hemos asumido su existencia, pero no fue explicitada en forma separada. Igualmente, observamos que un componente similar no es encontrado en los otros SIEMs bajo estudio, por lo que la pregunta queda claramente planteada: ¿por qué tenemos un componente específico para la comunicación?

Ya hemos expuesto que Prelude-IDS posee, además de la posibilidad de extender el “parsing” de datos a nuevas fuentes, una API que permite la integración de componentes de terceros. Esto no es posible con los otros SIEMs estudiados, entre otras razones por ser propietaria la comunicación interna entre los componentes. Como en el caso de Prelude-IDS no sólo la comunicación se basa en una mensajería

[17] Si bien IDMEF está definido como formato XML, PreludeDB almacena la información en formato binario. Uno de los componentes adicionales de la solución de Prelude Pro, incluye la posibilidad de exportar los datos almacenados en la PreludeDB, en IDMEF XML.

estándar, sino que además existe una API, nos lleva a pensar que los desarrolladores optaron por disponer un componente separado para el control de la comunicación, facilitando también la conexión a la API antes citada.

Sin embargo, existe otra función asociada al Prelude-Manager. Ya en nuestro SIEM genérico, indicamos la existencia de un componente responsable del almacenamiento de eventos. Si bien la mensajería IDMEF tiene un soporte en formato XML, otra decisión de diseño llevó a los desarrolladores a utilizar una base de datos para almacenar los eventos recibidos como mensajes IDMEF, por lo cual se dice que los mismos son almacenados en formato binario.

En este punto, el Prelude-Manager cumple una función adicional de control de acceso a la base de datos y, por tanto, realiza la función de almacenamiento de logs de nuestro SIEM genérico.

Para el envío de información, Prelude-IDS soporta los siguientes formatos mediante plugins correspondientes:

- Bases de datos MySQL, PostgreSQL, SQLite
- Reportes en formato XML
- Reportes en texto claro
- Relaying, permitiendo generar reportes de alerta para otros gestores
- Envíos de alertas en texto claro a través del servidor SMTP. [Yas09]

Esta capacidad de enviar diferentes tipos de alertas en base a diferentes logs lo provee de una gran flexibilidad.

Al momento de procesar los eventos Prelude-Manager está dotado de un sistema de colas y niveles de prioridad configurables. Esto permite personalizar el orden en que se traten los eventos según su procedencia o criticidad, aplicar filtros o incluso disparar acciones, según las necesidades de cada organización. [SNEH11]

V.6.4.2 – Libprelude

Libprelude (Library-Prelude) es una biblioteca que garantiza una conexión segura entre todos los sensores y el Prelude-Manager, y que además proporciona una API para la comunicación con los subsistemas de Prelude-IDS. De esta manera se tiene la funcionalidad necesaria para la generación y emisión de eventos IDMEF, la selectividad y la capacidad de retransmisión de datos en caso de interrupciones en las comunicaciones entre los componentes del sistema. Es de destacar asimismo que una de las funciones relevantes de Libprelude es la de garantizar la transmisión de los datos entre los componentes en forma segura mediante el uso de SSL.[Yas09]

Libprelude también facilita la tarea para que software de terceros (third-party software) puedan comunicarse con los componentes de Prelude-IDS. Esta biblioteca provee características comunes utilizadas por todos los sensores. [PreCo]

Podemos afirmar por tanto que el package Libprelude es un requisito para cualquier dispositivo del sistema que actúe como sensor o como manager.

Como mencionáramos, una de las fortalezas del proyecto Prelude-IDS ha sido la incorporación de otras herramientas OSS para que sean soportadas por Libprelude. De esta forma se permite que los equipos de desarrollo puedan trabajar en forma

independiente pero actuar como sensores en la solución. Esto simplifica además el despliegue de diversos sensores, como es el caso del Snort IDS.

V.6.4.3 – LibpreludeDB

LibpreludeDB es una biblioteca que proporciona una capa de abstracción del tipo y el formato de la base de datos utilizada para almacenar las alertas IDMEF. Permite así que los desarrolladores utilicen la base de datos IDMEF de forma fácil y eficiente sin tener que preocuparse acerca de SQL e independientemente del tipo y formato de los datos originales. [PreCo]

Los hosts que deseen reportar a través de la interfaz Web Prewikka deberán tener instalado necesariamente LibpreludeDB. [Yas09]

Si bien no se trata de un componente según nuestro SIEM genérico, la LibpreludeDB es el complemento requerido por el Prelude-Manager para la gestión del almacenamiento de logs.

V.6.4.4 – Prelude-LML

Prelude-LML (Prelude Log Monitoring Lackey), es un analizador de logs que permite a Prelude-IDS colectar y analizar información proveniente de todo tipo de aplicaciones y sistemas, mediante sus respectivos logs. Por tanto, podemos aprovechar su motor de análisis en cualquier sistema de generación de registros como ser:

- Sistemas Unix/Linux
- Switches y routers
- Firewalls
- Impresoras
- Otros sistemas que generen logs en formato Syslog como ser Windows NT/2K/XP, cargados con Ntsyslog o similar.

Considerando las funciones de colección de eventos y el análisis (“parsing”) de los mismos, es correcto asociar al Prelude-LML tanto con el colector de logs, como con la función de normalización de los mismos, para el SIEM genérico.

De esta forma mediante Prelude-LML y la utilización de Syslog en los dispositivos a sensar, es posible monitorear toda una red.

Para hacerse de la información de los dispositivos Prelude-LML tiene dos modos de operación:

- Escaneando los archivos de log en los dispositivos donde se esté ejecutando Syslog Server o cualquier otro sistema de log.
- Recibiendo mensajes UDP de Syslog desde los dispositivos a sensar.

Como vimos, la función principal de Prelude-LML es la de analizar registros, y para esto cuenta con un sistema de plugins el cual lleva a cabo las tareas de seguimiento y análisis a fin de descubrir anomalías en la seguridad o actividades maliciosas. En caso de encontrar un patrón conocido transformará los logs involucrados en una alerta con formato IDMEF.

Como mencionáramos, Prelude-IDS es capaz de monitorear casi cualquier tipo de log, ya sea del tipo Syslog, archivos planos genéricos, logs de sistema, y otros. En función de este aspecto, al utilizar múltiples archivos de log y con formatos diferentes, es posible configurar al Prelude-LML para que conozca cómo manejar cada formato de log. El formato de éstos logs así como qué tipo de archivos debe manejar se especifica en una sección de la configuración del propio componente (Prelude-LML). En dicha sección se describe, por ejemplo, el cómo manejar el encabezado del log original y mapearlo a uno de tipo IDMEF; así como el adjuntar el nombre del host y el proceso involucrado a la información de alerta de tipo IDMEF.

En caso de contar con dispositivos que no tengan compatibilidad nativa con Prelude-IDS, es posible optar que éstos reporten a un servidor Syslog, el cual podrá a su vez ser monitoreado por Prelude-LML. [Yas09]

V.6.4.5 – Prelude Correlator

Prelude Correlator es aquel componente de la arquitectura Prelude-IDS que permite correlacionar los eventos de uno o múltiples Prelude-Manager. Para esto es posible cargarle reglas de correlación personalizadas utilizando el lenguaje de programación LUA. Si el flujo de eventos coincide con una de las reglas de correlación, se generará una alerta. [Yas09]

El nombre es bastante explícito con su función, lo cual facilita su asociación con el motor de reglas / motor de correlación de nuestro SIEM genérico. Tal vez sea este el componente más crítico de la solución y, a la vez, el menos documentado.

A lo largo del ciclo de vida del proyecto Prelude-IDS, se han utilizado diversos lenguajes de programación para este componente, siendo originalmente Python, algunos acercamientos a SEC (Simple Event Correlator), para finalmente llegar a LUA en la actualidad. La argumentación para la elección de LUA es la flexibilidad requerida, que no era alcanzada con los lenguajes elegidos originalmente.

La documentación referida a este módulo es un punto pendiente y necesario para extender las funcionalidades actuales. Tómese en cuenta que en el caso de los productos comerciales, la implementación del motor de correlación es uno de los secretos mejor guardados.

V.6.4.6 – Prewikka

La interfaz Prewikka es una interfaz Web GUI (Graphical User Interface). Esta componente cumple la función de monitoreo asociada a nuestro SIEM genérico.

Existen dos versiones de esta interfaz, la versión libre y la versión Pro (comercial). La versión Pro tiene como funcionalidades extra:

- Un sistema de ticket integrado
- Gestión de sensores remotos
- Gráficas estadísticas totalmente personalizables
- Vistas virtuales de alertas
- Exportación de alertas a archivos PDF
- Autenticación segura desde un servidor LDAP

Esta interfaz está organizada en pestañas y menúes, donde se permite, en función del nivel de permisos del usuario, realizar diversas tareas. Dentro de la

ventana principal existe una pestaña dedicada a los eventos. Allí se muestran todos los eventos reportados en la última hora, día o lapso que el usuario defina para su despliegue.

Otra pestaña llamada Configuración, está dedicada a la creación de filtros predefinidos para la presentación de los eventos.

Como muestra de la interfaz, se observan las Figura 38, Figura 39, Figura 40 y Figura 41.

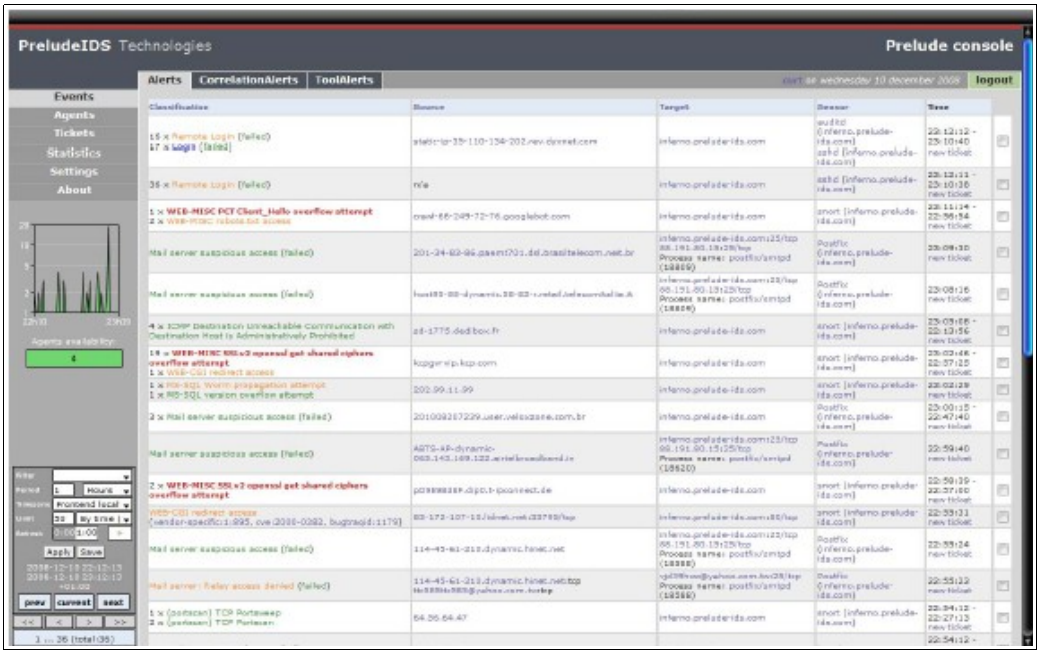


Figura 38 - Interfaz Prewikka de Prelude-IDS – Despliegue de eventos [Yas09]

Para obtener más información acerca de un evento en particular se cuenta con la sección ”Event summary”. Se desplegarán allí todos los eventos que coincidan con la descripción del evento en particular, su origen, su destino y el sensor involucrado en el reporte. En caso de que existan varios eventos similares, el sistema los desplegará en forma consolidada como un único evento.

Figura 39 - Interfaz Prewikka de Prelude-IDS – Clasificación de eventos [Yas09]

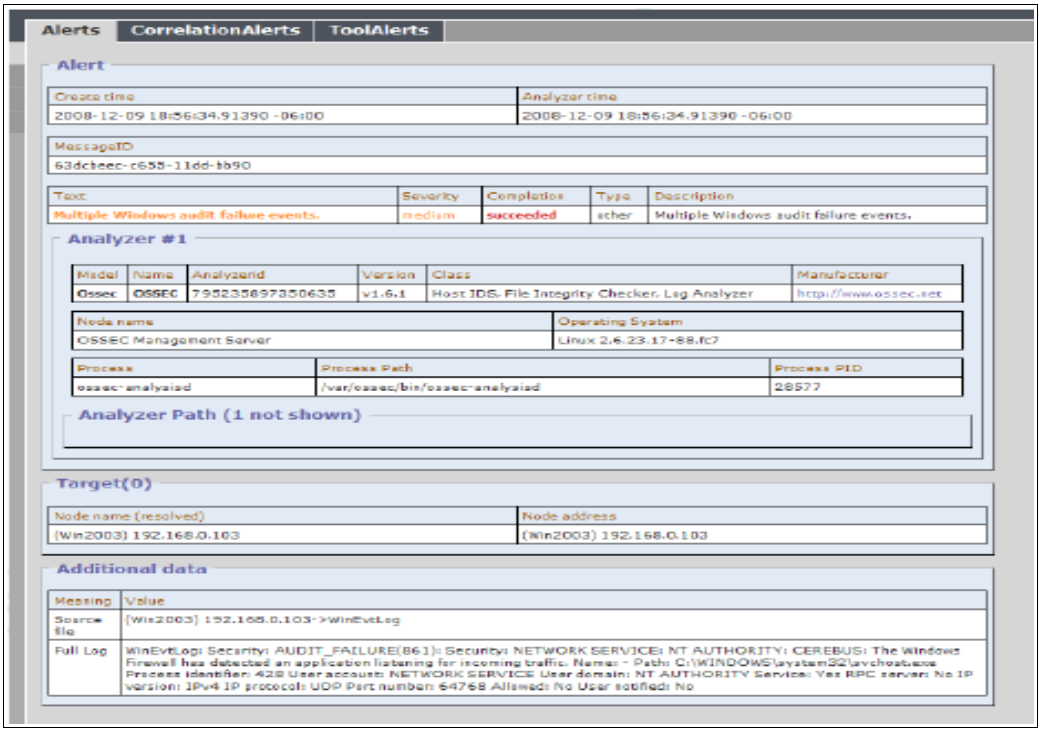


Figura 40 - Interfaz Prewikka de Prelude-IDS – Despliegue detallado [Yas09]

Prewikka también cuenta con una tabla donde figuran los agentes que le reportan. En ella se informa sobre los agentes y sensores registrados en el Prelude-Manager, sus características y estado actual, la lista de alertas que cada sensor ha reportado, fechas de últimos reportes, etc.

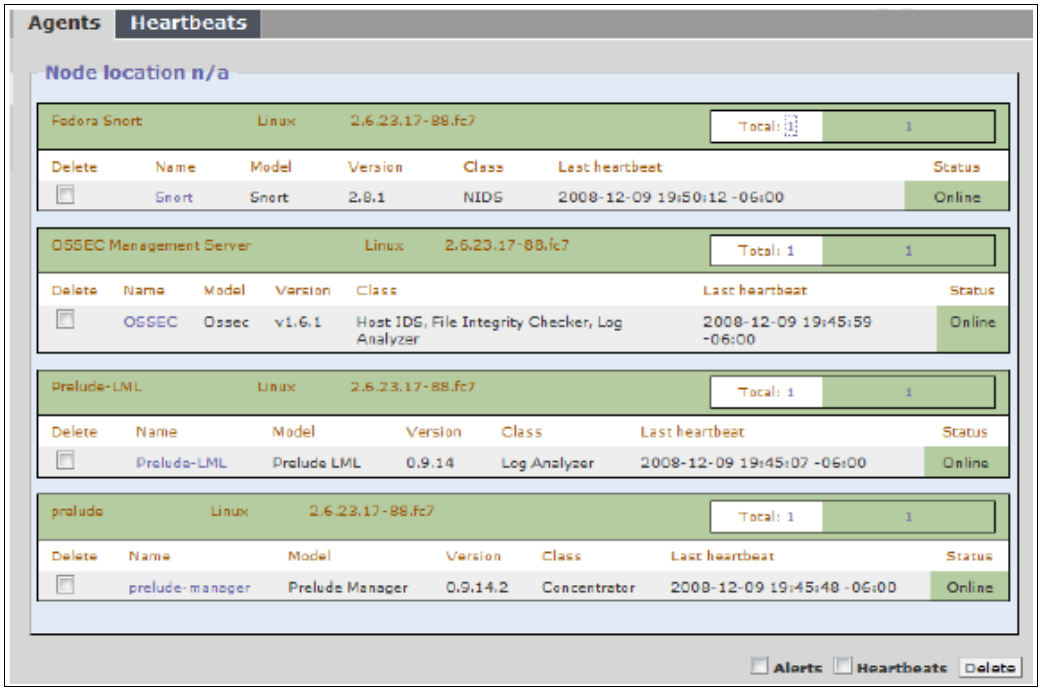


Figura 41 - Interfaz Prewikka de Prelude-IDS – Agentes y sensores [Yas09]

Por otra parte la versión Prewikka Pro tiene como funcionalidad extra la configuración de los agentes involucrados, como se muestra en la Figura 42. En ésta, es posible definir: el intervalo de reporte de estado (heartbeat-interval), dirección del servidor, nombre del analizador, nombre del nodo, localización, categoría y dirección, entre otros parámetros.

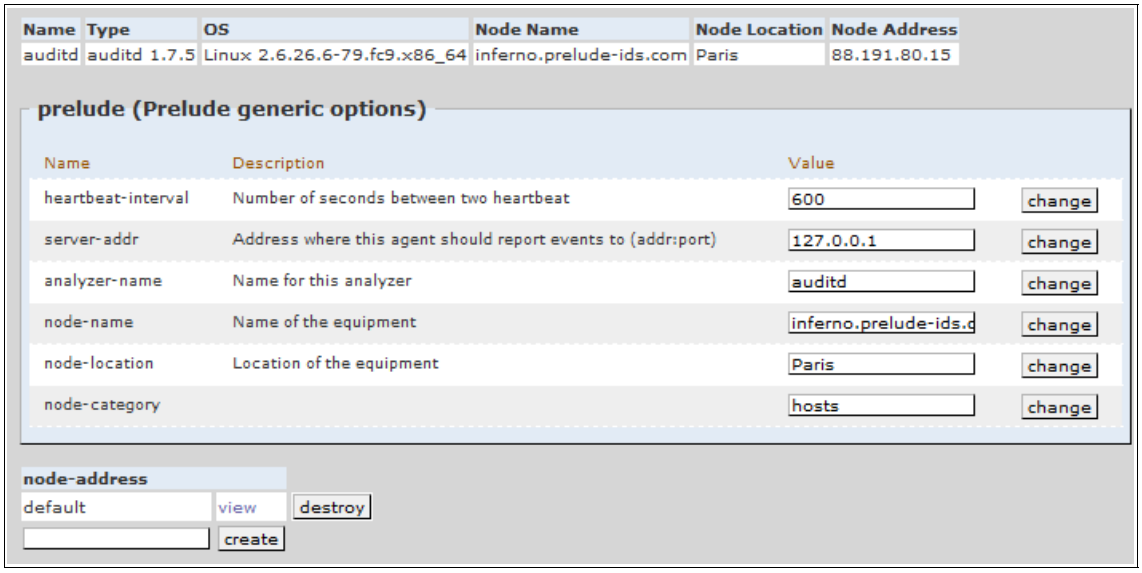


Figura 42 - Interfaz Prewikka de Prelude-IDS – Configuración de sensores [Yas09]

V.6.4.7 – PFLogger

PFLogger es el componente utilizado para recolectar los logs desde sistemas de filtros basados en Packet Filter[18] (PF) de BSD. Una vez instalado y configurado, escuchará desde aquellos sistemas BSD que estén ejecutando PF a fin de realizar el análisis de los logs respectivos, redireccionándolos y enviando alertas al Prelude-Manager.[19] No se trata de un módulo no previsto en nuestro SIEM genérico, es simplemente un caso particular de un colector de eventos.

Como vemos, Prelude PFlogger es un sensor especializado y exclusivo, donde su razón de ser es debida a la gran cantidad de soluciones de filtrado de paquetes que se valen de las bondades de Packet Filter. Tal como presentábamos en el párrafo anterior, Prelude PFlogger simplifica este aspecto de recolección de logs al estar "escuchando" en la interfaz virtual *pflog0,* donde PF redirecciona sus logs, y los envía como alertas al Prelude-Manager. Por tanto, Prelude PFLogger no requiere de pflogd para operar, pudiendo estar ambos ejecutándose en el mismo host.[20]

V.6.4.8 – Prelude Import

Prelude Import es una extensión comercial disponible para la arquitectura Prelude-IDS. El propósito de esta herramienta es brindar capacidades de importación de datos desde las aplicaciones que reporten eventos en un formato específico. También puede ser utilizado para emitir una alerta a partir de un shell script.

Como componente de nuestra solución genérica, podemos considerar que Prelude Import está al mismo nivel del módulo de normalización de eventos, sólo que en este caso no alimenta a nuestro motor de reglas / motor de correlación con eventos originados por dispositivos fuente, sino que permite tener información adicional para el análisis de los eventos recibidos.

De esta forma son soportados tres tipos diferentes de formatos de alerta:

- IDMEF XML: Importa eventos en formato IDMEF-XML y lo convierte en el formato nativo a Prelude-IDMEF
- Nessus XML: Importa el reporte XML de Nessus. Se utiliza para generar eventos en base a los reportes de vulnerabilidades generados por el scanner de Nessus.
- Objetos IDMEF: Un formato IDMEF específico de Prelude-IDS, altamente manejable para representación textual. [PreIm]

V.7 – Elección de la herramienta

Una vez estudiadas las cuatro herramientas SIEM seleccionadas, consideramos que la elección por parte del Grupo Gartner de HP ArcSight SIEM, tiene razones de peso. Es claramente ésta la herramienta que mayor versatilidad presenta, siendo posible su implantación en entornos diversos. La amplia variedad de sensores soportados, la posibilidad de modularización completa de la solución a fin de ajustarse

[18] PF (Packet Filter) es el sistema de BSD para filtrado de paquetes y de tráfico en general. Además permite, entre otras funcionalidades, el hacer NAT (Network Address Translation)

[19] Referencia de PFLogger: http://www.openbsd.org/4.3_packages/powerpc.html

[20] Referencia del último párrafo: http://freshmeat.net/projects/prelude-pflogger/

a cada caso particular, su madurez, escalabilidad, así como la posibilidad de expansión de un despliegue distribuido, la hacen un excelente ecosistema de productos a elegir.

Ahora bien, dado que al presente se desconoce sobre los planes del cliente así como de la existencia de presupuesto comprometido para la adquisición de una solución de estas características en el MDN; que la posibilidad de instalar una versión "demo" queda condicionada a la capacidad de EPS necesarios, los cuales no son conocidos por el cliente como especificación de requerimiento; y, adicionalmente, que la documentación pública de HP ArcSight no es del detalle esperable a la hora de realizar una implantación de este tipo, es que optamos por no implementar el piloto con este SIEM.

Como contrapartida, vemos que la documentación en aquellos proyectos OSS es más amplia, profunda y variada. Tal es el caso de Prelude-IDS, donde la misma es notablemente más detallada, con varios y surtidos ejemplos originados por el soporte oficial del proyecto así como el aporte de la comunidad OSS.

Sin embargo, dado que los sensores soportados por HP ArcSight son suficientemente amplios, no se avisora impedimento técnico alguno para una migración futura a esta herramienta, en caso de que el cliente lo considere. En el caso de evaluar la migración, se deberá considerar además, otros aspectos tales como el económico y de licenciamiento, el expertice del personal afectado a la tarea, las dimensiones del despliegue considerado.

Por las consideraciones antes vistas, se elige a Prelude-IDS como SIEM. A continuación se detallan las razones de esta elección.

V.7.1 – SIEM Seleccionado

Ya anticipamos en los párrafos anteriores, nuestra elección de Prelude-IDS para el presente proyecto. Pasaremos ahora, a explicar las razones de dicha elección.

Los puntos que se consideraron fueron:

- **Razones económicas**. Al tratarse de una solución OSS, no se posee costo de adquisición. El único CapEx involucrado, es el referente a la adquisición del hardware para los diferentes componentes, así como las horas hombre para su implantación.
- **Documentación**. El proyecto posee una excelente Wiki donde se explica las funcionalidades de cada componente. También incluye información de como integrarlo a terceras partes, fundamentalmente los sensores existentes. Como también indicáramos, existe también una amplia utilización del proyecto durante varios años, lo que genera aporte comunitario a la misma de relevancia, contándose con diversas fuentes de información para su implementación, ajustes, configuraciones, entre otras.
- **Madurez del proyecto**. Si nos remitimos a los orígenes del proyecto, el mismo posee más de diez años. Adicionalmente, es uno de los proyectos OSS en materia de seguridad más difundidos y aceptados, y que cuenta con una activa comunidad de usuarios.
- **Comunicación de los componentes**. El soporte de comunicación de Prelude-IDS, es una biblioteca que permite la utilización de SSL para el cifrado en el intercambio de información entre los componentes con el

SIEM. Esta biblioteca ha sido integrada por los proyectos de desarrollo de sensores más populares y diversos, como en el caso de Snort que incorporó la misma como uno de sus componentes.

- **Integración con los componentes presentes en la red del MDN**. Considerando los diversos activos/componentes de la red del MDN, la implantación de Prelude-IDS no presenta desafíos insalvables. Los activos informáticos presentan alguna facilidad o es posible incorporarla, lo que permite su integración con Prelude-IDS.
- **Capacidad de migración**. Ya indicamos que si bien la elección es Prelude-IDS, consideramos que HP ArcSight es la mejor opción en caso de no contarse con las restricciones para el presente proyecto. Dado que la forma de implantación modular que soporta Prelude-IDS es compatible con el despliegue de HP ArcSight, entonces entendemos que es posible migrar una red de sensores basada en un SIEM a otro. En el caso de que el MDN considere dicha posibilidad, la elección de Prelude-IDS no es una barrera. [SNEH11]

VI. Propuesta para el MDN

VI.1 – Introducción

Cumpliendo con el objetivo central del presente proyecto se plantea el diseño de una red distribuida de sensores de seguridad informática a ser propuesta para la red informática de la DGSE del MDN.

Como aspectos generales del diseño se tendrá en consideración los siguientes:

- requerimientos en Seguridad Informática acordados con el MDN;
- infraestructura informática relevada en la DGSE;
- servicios que corren sobre dicha infraestructura;
- el potencial y la capacidad que ofrece la mencionada red;
- otros aspectos a mejorar detectados a partir de los relevamientos y la prueba de concepto a realizar;
- mejoras y sugerencias que el Equipo de Proyecto entiende aconsejable implementar en dicho entorno.

Para esto se propone la incorporación de componentes nuevos a instalar y la utilización de otros componentes, los cuales ya dispone el cliente en su red y que, ajustes y configuraciones mediante, podrían cumplir un rol o bien como sensores en sí mismos o como actores adicionales que permitirán la operación de la solución propuesta.

En esta sección se presentarán propuestas en seguridad sugeridas en base a una red general típica para una empresa de porte similar a la de la organización bajo estudio. En virtud del Acuerdo de Confidencialidad suscripto con el MDN, la misma no representa la red estudiada, pero permite realizar la presentación de los componentes en forma general.

VI.2 – Descripción de la red involucrada

La Red de la Organización ficticia bajo estudio esta distribuida en una topología estrella. Cuenta con switches de acceso que agregan el tráfico e interconectan a otras áreas mediante enlaces de fibra óptica de 1GbE. Por su parte, el core está compuesto de un firewall que interconecta a nivel de capa 3 los distintos segmentos así como el enlace WAN. El esquema se muestra en la Figura 43.

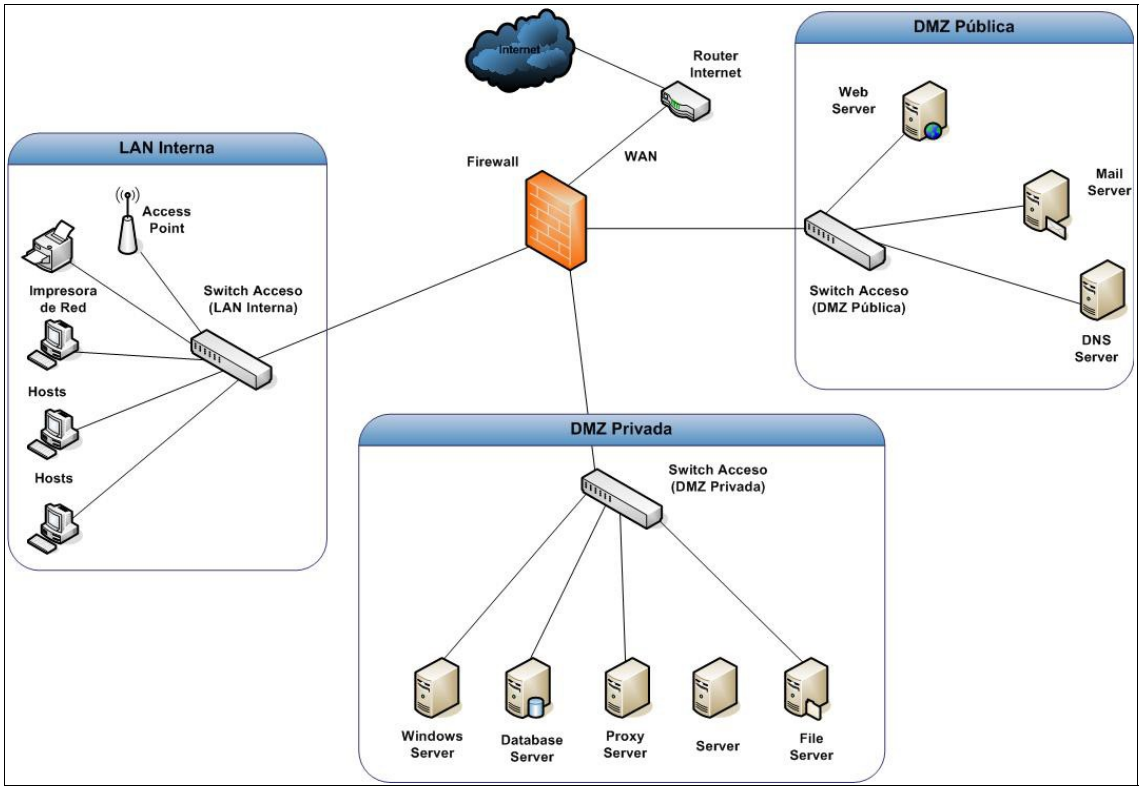


Figura 43 - Red genérica original.

Todos los servidores de uso interno (DMZ Privada) están ubicados en un mismo lugar físico, conectados al core de red y comparten un mismo dominio de broadcast. Únicamente ofrecen servicios al segmento de red interno (LAN Interna) de la Organización. Todos estos servidores presentan únicamente aquellos puertos necesarios para cumplir con el servicio.

El segmento identificado como DMZ Pública es una red separada. Se ofrecen servicios plenamente identificados hacia Internet y algunos tanto hacia Internet como hacia la LAN Interna.

Por otra parte, el router que gestiona el enlace externo de la Organización, Internet, no es administrado por personal de la misma.

VI.3 – Requerimientos en Seguridad Informática

A continuación se enumeran los activos de red que consideraremos críticos. Luego de esto se describirán los requerimientos en seguridad considerados para la presente red. Nuevamente, nuestra consideración se basa en un caso general, no necesariamente exhaustivo. En base a esta definición se propondrán las medidas en seguridad a adoptar para cada caso.

VI.3.1 – Activos Críticos

Identificamos como los *activos críticos* los siguientes:

- Servidores cuyos servicios estén expuestos a Internet:
  - servidores Web;
  - servidor de Correo;
  - servidores de DNS.
- Servidores de desarrollo de aplicaciones.
- Servidores de bases de datos.
- Servidor de DNS interno.
- Servidores de archivos que contengan información sensible para la Organización.
- Otros servidores internos que resulten esenciales para la Organización.
- Estaciones de trabajo que manejen información sensible.
- Firewalls.
- Switches de Core.
- Switches secundarios en general.

VI.3.2 – Activos no críticos

Se consideran como activos *no críticos*[21] dentro de la Red de la Organización a:

- todas aquellas estaciones de trabajo que no se explicitaron en la sección anterior;
- aquellos servidores que corren otras aplicaciones y servicios que no se mencionaron en la sección anterior;
- las impresoras de red[22];
- otros elementos considerados "menores" de red.

VI.3.3 – Requerimientos en Seguridad considerados para la red genérica

A continuación se enumeran los requerimientos generales para el diseño de la Red de Sensores de Seguridad Informática (SensorNET). Los mismos no contemplan los requerimientos particulares de organización alguna, sino los que entendemos son generales a la red presentada. En la realidad, estos requerimientos deberían surgir de los requerimientos propios del cliente, el conocimiento del negocio, el relevamiento de la infraestructura existente y de las entrevistas con el personal técnico de la organización en cuestión.

- Requerimiento #1: Visualización del tráfico procesado por el firewall.
- Requerimiento #2: Trazabilidad del tráfico procesado por el firewall.
- Requerimiento #3: Visualización del tráfico procesado por determinados servidores de la DMZ Privada.
- Requerimiento #4: Analizar el comportamiento de los servidores de la DMZ Pública.

[21] Si bien estos activos no contarán con sensores dedicados, otros sensores que se propondrán en el modelo aportarán información adicional sobre el estado de "salud" de los mismos.

[22] Considerando en este punto a aquellas impresoras en las que no se imprimirá información sensible para la Organización.

- Requerimiento #5: Analizar el comportamiento de los servidores de la DMZ Privada.
- Requerimiento #6: Capturar potencial malware existente o futuro en la LAN Interna.
- Requerimiento #7: Disponer de una consola centralizada de seguridad informática para la visualización de eventos de seguridad relevantes.

VI.4 – Diseño de la Red de Sensores de Seguridad Informática

VI.4.1 – Esquema de la Red Propuesta: SensorNET

A partir del relevamiento de la infraestructura de Red del cliente y de sus requerimientos, para este documento hipotéticos, es que se presenta en la Figura 44 un esquema de la infraestructura existente afectada al despliegue de la red de sensores a implementar, el nuevo equipamiento, así como nuevos sensores y configuraciones propuestas para las distintas áreas. El lector reconocerá los segmentos de red y los elementos ya presentados de la red original, más un nuevo segmento de red destinado exclusivamente a SensorNET, y en el que se instalarán los equipos centrales de la nueva red de sensores.

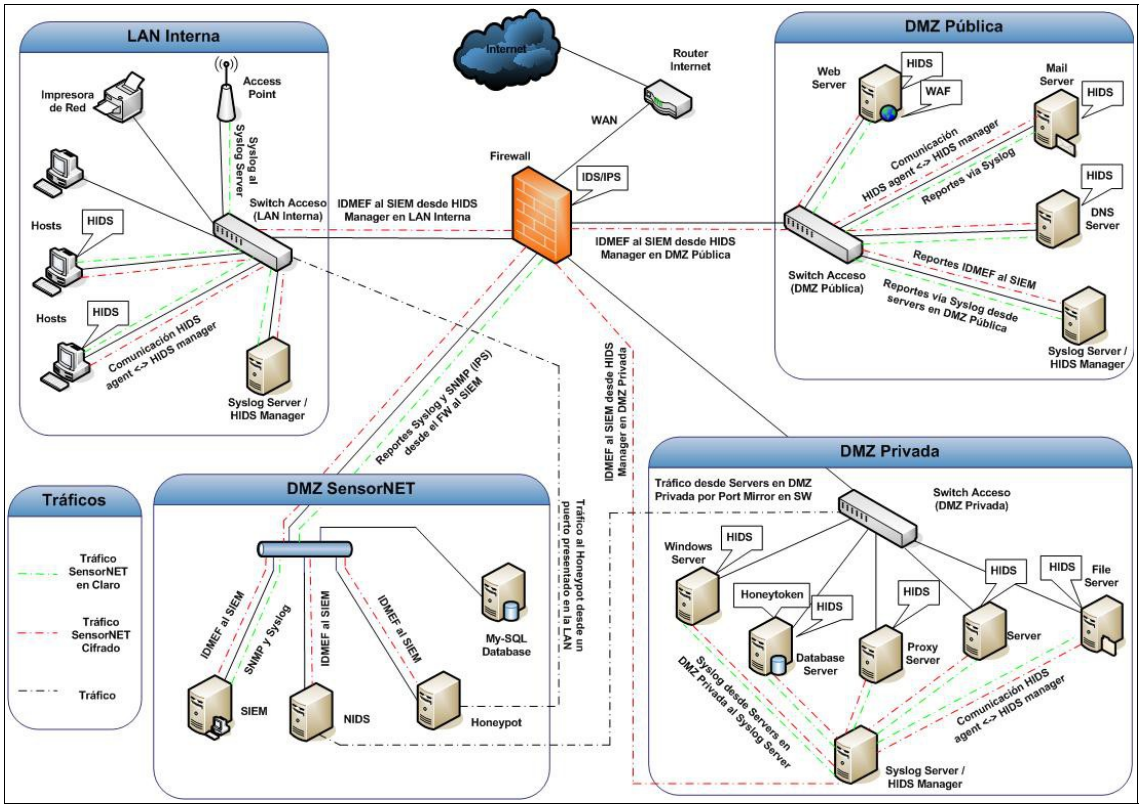


Figura 44 - Diagrama lógico de la red distribuida de sensores propuesta

En la Figura 44, los trazos punteados representan aquellos flujos de datos pertenecientes a los reportes de seguridad de los distintos sensores, identificándolo

según su categoría. Esta es una vista macro de la solución completa, que pretende generar en el lector el impacto de cómo una amplia variedad y considerable porción de la infraestructura de la Organización pasa a cumplir el rol de sensor.

A lo largo de las siguientes secciones se hará el desarrollo y la justificación del por qué de los elementos seleccionados existentes en la Red de la Organización para la incorporación a ésta red de sensores así como de aquellos otros elementos necesarios y sus respectivas funciones.

VI.4.2 – Elementos involucrados en el cumplimiento de los requerimientos

Tomando en cuenta el equipamiento con que ya cuenta la organización, se plantea en esta sección cómo dar cumplimiento a los requerimientos originales del cliente. A continuación se describirá cuáles de dichos activos participarán en el cumplimiento de lo solicitado (vale decir en la nueva red de sensores) así como las modificaciones necesarias en su configuración.

Este equipo de trabajo ha procurado una afectación mínima de modo de no perjudicar la operativa y que se pueda lograr una mejora en la visibilidad del estado de la seguridad para los equipos afectados y su entorno.

- Firewall: Para cumplir los requerimientos #1 y#2 es necesario activar las funcionalidades existentes en este equipamiento para el reporte de eventos mediante Syslog o SNMP. Como además sería deseable contar con una inteligencia que procese estos reportes, el habilitar estas capacidades y que dichos reportes sean enviados directamente a un SIEM es lo que se propone.

- Switches de Core y Acceso: Mediante activaciones y configuraciones de la funcionalidad de Port-Mirroring en aquellos puertos que tengan conectados los activos que la Organización desee monitorear, es posible presentar el tráfico a los NIDS a instalar, cumpliéndose así con el requerimiento #3. Configurando en forma análoga los switches de acceso de la DMZ y de core es posible extender este requerimiento a otros segmentos de red. Como es de esperar, este requerimiento exigirá notablemente las prestaciones de los switches mencionados; por lo tanto, se deberá tener especial consideración en los anchos de banda utilizados por los puertos y en la capacidad de procesamiento (paquetes por segundo) disponible en dichos elementos.

VI.4.3 – Elementos extras necesarios para el cumplimiento de los requerimientos

En función de los requerimientos planteados, es necesario la incorporación de otros componentes a fin de lograr el cumplimiento de los requerimientos presentados.

Considerando los mismos, se hace necesario desplegar:

- NIDS: Ya sea actuando en un servidor físico específico o virtualizado, alineado al cumplimiento del requerimiento #3, tal como fuera indicado previamente.

- HIDS: Mediante la instalación de sensores de tipo Host IDS, tanto en los servidores de la DMZ Pública como en los de la DMZ Privada, es posible cumplir con los requerimientos #4 y #5.

- Honeypot: Se propone el uso de un honeypot de baja interacción, ya sea ejecutándose en un servidor físico específico o virtualizado, a fin de cumplir el requerimiento #6.

- SIEM: A fin de cumplir con el requerimiento #7, se recomienda fuertemente la instalación de un nuevo servidor.

Es claro la necesidad de incorporación de nuevos servidores a la red de la Organización. Algunos de los mismos se ubicaran en la nueva DMZ SensorNET (SIEM, NIDS, Honeypot), mientras que otros se encontraran más cercanos a sus fuentes.

Indistintamente de su ubicación, se propone utilizar o bien servidores físicos independientes o un nuevo servidor con capacidades suficientes como para implementar una solución virtualizada total y que contenga al SIEM, algunos NIDS, al Honeypot y que además proporcionará el espacio para el almacenamiento de todos los eventos.

Sobre los requerimientos para un solución virtualizada total estimamos un servidor con procesador multinúcleo, memoria RAM de unos 8 a 12 GB, almacenamiento superior a los 500 GB de capacidad y algún tipo de RAID. La mayoría de éstos recursos se emplearán en el procesamiento que requiera el motor de la base de datos, las capacidades de correlación, el almacenamiento y reporte de eventos que el SIEM utilizará.

VI.4.4 – Sistemas de Comunicaciones y de Reportes necesarios en el cumplimiento de los requerimientos

Cumpliendo las consideraciones mencionadas en las dos secciones anteriores se estaría cubriendo los requisitos originalmente planteados. Es necesario definir ahora los protocolos de comunicación y la mensajería utilizada por los diferentes componentes.

Los mismos serán:

- Syslog: Será utilizado en todos aquellos switches, servidores y access-point que soporten este formato, reportando sus eventos relevantes directamente al SIEM.

- SNMP: Otra alternativa para estos dispositivos es el uso de traps SNMP. En cualquiera de las dos alternativas, deberá activarse y configurarse debidamente esta capacidad.[23]

- IDMEF: Este formato de mensajería se empleará en aquellos sensores y servidores que, debidamente configurados, cuenten con un agente propio y compatible que reporte directamente al SIEM.
El hecho de contar con el SIEM a instalar definido [SNEH11], permite adelantar que en cada servidor que así lo permita se instalará el módulo pertinente, logrando el uso de la referida mensajería para la comunicación directa con el SIEM. Para los servidores en que esto no sea posible, se podrá tanto desarrollar plugins como usar otras herramientas.

VI.4.5 – Otros aspectos en seguridad que conforman la solución completa

En esta sección se abordarán aquellos aspectos adicionales que mejoran la solución ya presentada.

Vale decir que los sensores y reportes aquí tratados vienen a complementar los ya descritos hasta el momento, completando así lo que entendemos como una "propuesta completa" para la red estudiada.

Un aspecto no menor[24] es el asegurar la confidencialidad y la integridad en el intercambio de información entre todos los sensores que participen de la red.

La propuesta que este Equipo hace es el uso de aquellas herramientas que, en concordancia con los sistemas operativos existentes, permitan la utilización de algún tipo de mecanismo de cifrado o de comunicaciones seguras. Dado que los mensajes Syslog viajan en texto claro[25] se recomienda el uso de herramientas, tales como Syslog-ng u otras, que soporten SSL de manera de contemplar el aspecto de la confidencialidad. Asimismo, permitirán establecer comunicaciones seguras sobre TCP y garantizando la entrega de los mensajes.

Ante este escenario, de las investigaciones en diversas tecnologías que el Equipo ha realizado y procurando además el cumplir con los Requerimientos #4 y #5 del cliente, se propone el despliegue de una solución particular de HIDS. Esta solución es OSSEC[26]. Cuenta con agentes para Linux y Microsoft Windows que se comunican con sus respectivos managers utilizando mecanismos de compresión de datos y criptográficos[27], y es compatible en forma nativa con el SIEM a implementar. La misma cumpliría con la canalización de reportes de eventos en forma segura, el análisis y. monitoreo del comportamiento de los servidores, e incluso liberar anchos de banda de los enlaces.

[23] La Organización debería optar por un mecanismo de reporte (Syslog o SNMP, o ambos) a conveniencia, y en función de sus objetivos.

[24] El tema seguridad en los protocolos de intercambio de información aquí tratados fue abordado en el documento [SNEA11].

[25] Se resalta que la amplia mayoría de los elementos activos de la actualidad sólo transmiten eventos por Syslog en texto claro.

[26] Sitio de OSSEC: http://www.ossec.net/

[27] OSSEC emplea el algoritmo de cifrado de clave simétrica Blowfish en estas comunicaciones.

Este modelo pretende incorpora un servidor dedicado por cada segmento de red, el cual tendrá como misión:

- ser el gestor de todos los eventos reportados por cada HIDS corriendo en modo agente en aquellos equipos designados dentro de su segmento;
- almacenar eventos y logs;
- procesar los eventos y logs;
- reportar al SIEM en formato IDMEF[28];
- minimizar el uso del ancho de banda al tener un único elemento por segmento reportando directamente al SIEM.

Se propone además instalar OSSEC en todas aquellas estaciones de trabajo consideradas críticas, y especialmente en aquellos dispositivos portables (notebooks).

Los otros aspectos que el Equipo sugiere son:

- activar en todos los elementos activos de la Red la capacidad de reportes mediante Syslog o traps SNMP de sus principales eventos. Muchos de los existentes en el mercado son compatibles con el SIEM seleccionado;

- instalación de web application firewall (WAF) en aquellos servidores web definidos. Si bien sería una medida muy positiva para contrarrestar determinados tipos de ataque, demanda un estudio detenido de los sitios y las aplicaciones web que éstos servidores hostean, cantidad de accesos así como consideraciones frente a los recursos computacionales que los mismos deberán disponer. Se puede apuntar aquí a que los WAF se ejecuten o bien en el propio servidor web o en otro dedicado a tal fin, lográndose un esquema de proxy reverso donde todo el tráfico web es procesado por un único servidor. En el caso particular que el cliente utilice servidores web Apache (el más utilizado globalmente), como en caso de montar un proxy reverso, recomendamos ModSecurity como WAF.

- implementar honeytokens en aquellas bases de datos seleccionadas. Si bien esto no protege de por sí la información almacenada, oficiará de sensor ante copias o accesos indebidos a señuelos que los administradores definan. Para alcanzar esto se deberá, además de definir, generar y almacenar dichos honeytokens, entrenar a los IDS para la detección;

- implementar honeypots más sofisticados (alta interacción o honeynet). Si la organización tiene intención de capturar malware, se podría mantener un esquema, donde sean presentados únicamente los servicios señuelos a determinados enlaces o conjunto de equipos. Entendemos que esta es una tarea que demanda labor y un entorno controlado. Por ello, y continuando en la línea de señuelos, para la propuesta completa se sugiere el uso de honeypots más avanzados. Nuevamente se puntualiza que esta tarea demanda estudio y trabajo, máxime si se pretende que el reporte sea directamente al SIEM. En este sentido entendemos como una alternativa interesante al popular Nepenthes (actualmente discontinuado) su actual sucesor Dionaea.

[28] Los reportes IDMEF al SIEM a desplegar serán sobre SSL, cumpliendo así con uno de los pilares de la Seguridad Informática, la confidencialidad.

VI.5 – Escalabilidad

Uno de los requerimientos esenciales para un “Enterprise grade” en este tipo de soluciones es la capacidad de escalabilidad. Al momento de considerar la misma se debe tener en cuenta las dos dimensiones posibles de la misma: “Scale Out” y “Scale In”.

En el primer caso (scale out), se refiere a un despliegue distribuido. Ya fue discutido en [SNEH11], que tanto la solución sugerida como la que se implementará en la prueba de concepto, soporta varios esquemas de distribución. El más básico, consiste en la posibilidad de utilizar sensores en diferentes puntos y lograr su operación coordinada desde la consola de gestión. A esto también se le agrega la capacidad de conformar una arquitectura jerárquica; es decir, permitir que existan SIEM departamentales, geográficos o en función de una segmentación a elección, los cuales concentrarán la información de su ámbito de administración, luego de lo cual redirigirán la misma a un SIEM central, que poseerá una visión global de toda la Organización.

En el segundo caso (scale in), se aprecia claramente al momento de analizar el elemento central. Primero, es notorio que el soporte de base de datos, tanto en el volumen de almacenamiento como en la capacidad de I/O, es básico para la adecuada operación. También, si consideramos las capacidades de correlación de la solución, esto conlleva al requerimiento de capacidades computacionales para dicho fin.

Para los SIEMs propuestos, en el caso de Prelude-IDS, la solución se basa en componentes de software por lo que su capacidad queda condicionada a la infraestructura de cómputo utilizada. Es posible comenzar con cierta capacidad y luego migrar a una mayor cambiando la infraestructura donde se ejecutan los componentes centrales. En el caso del HP ArcSight, la posibilidad de poseer la solución basada en un appliance o en software, permite también un crecimiento. Sólo hay que considerar que en el caso del appliance es necesario realizar un número mayor de pasos, especialmente si en el crecimiento se desea disgregar alguna de las funciones que originalmente estaban soportadas por el applicance a migrar.

VI.6 – Compatibilidades

La herramienta a incorporarse como SIEM deberá poseer una amplia gama de compatibilidades con diversos sensores. Para la herramienta propuesta y elegida para la prueba de concepto (Prelude-IDS), se conoce que posee soporte nativo y a muchas otras aplicaciones por medio del registro de eventos (logs), así como un lenguaje que permite extenderlo a otras [SNEH11]. En el caso de HP ArcSight la situación es muy similar. La única diferencia entre los SIEMs indicados, es que en el caso de Prelude-IDS se posee adicionalmente una API que permite que un sensor dialogue directamente en la mensajería de los componentes (IDMEF), lo cual no es posible en el caso de de HP ArcSight dado que la misma es propietaria.

Al momento de definir compatibilidades, es necesario visualizar cuáles son los soportes nativos de la solución elegida. Este Equipo relevó que, tanto Prelude-IDS como HP ArcSight, poseen un amplio número de plugins para el procesamiento de eventos desde diversos dispositivos fuente. Esto no es menor, máxime la heterogeneidad de dispositivos, servicios y aplicaciones que puede ser de interés

incorporar como sensor. Un ejemplo es la capacidad nativa de procesar la mensajería del módulo del servidor web Apache ModSecurity, tal como fuera una recomendación propuesta presentada para la incorporación de WAF.

Otro punto a destacar, es la existencia de un entorno para desarrollar nuevos plugins. La razón de ello, es la imposibilidad de tener plugins para todos los potenciales sensores que existen en el mundo. La misma incluye la necesidad de poder incorporar como sensores aquellas herramientas que fueron desarrolladas por la propia organización. Tanto para el caso de Prelude-IDS como el de HP ArcSight, estas facilidades de programación se tienen incluidas en los componentes que recepcionan los mensajes de las diversas fuentes.

VI.7 – Futuros trabajos

Varias de las recomendaciones para la solución completa requerirán investigación sobre qué es más conveniente incorporar y su desarrollo e implementación en sí mismos, más que la propia incorporación a la red de sensores desplegada. Tal es el caso de la instalación de WAF o honeypots más sofisticados, incluso honeytokens con el posterior entrenamiento de los sistemas IDS (NIDS y HIDS).

Sin embargo este Equipo entiende que en función de la criticidad y de los activos a proteger (tanto servicios como equipos) puede bien valer el esfuerzo. Lógicamente esto dependerá también del nivel de madurez de la Organización y recursos en general para perseguir estos objetivos.

Por otra parte, la integración a la red de sensores de otras soluciones de gestión que la Organización disponga o piense desplegar, tales como sistemas Network Management Systems (NMS) o afines para la gestión de su infraestructura es una tarea que demandará analizar e investigar qué informaciones intercambiar entre estos sistemas, alternativas y compatibilidades. Un aspecto a destacar es que, de por sí, un NMS estará recibiendo y procesando volúmenes importantes de información, por lo que las características de los enlaces y del equipamiento gestor será un punto relevante en su dimensionamiento.

Otra línea de investigación es la incorporación de otros dispositivos o activos como sensores. En caso de existir aplicaciones desarrolladas “in house” por el cliente, las cuales sea de interés monitorear por parte de la Organización, es posible incorporar sus registros de auditoria (logs) al análisis del SIEM.

Si bien existe un motor de correlación en Prelude-IDS, es posible extender sus capacidades a partir de programación de nuevas extensiones y mejorar la función de correlación. [SNEH11]

Finalmente, otra línea de trabajo más que interesante y que puede ser de beneficio para la Organización que apunte a asegurar otras redes, es replicar este esquema de red de sensores.

De esta manera se dispondría de una capacidad de visión global al centralizar determinados eventos críticos a la seguridad. Esto se lograría con un modelo

distribuido de SIEMs, con SIEMs por cada una de las dependencias externas y comunicándose (enlaces VPN quizás) al SIEM central, permitiendo cumplir con lo planteado. Nuevamente aspectos como capacidades de procesamiento y enlaces se tornan relevantes frente a toda consideración.

VI.8 – Conclusiones sobre la red de sensores propuesta

A lo largo de esta sección fueron planteados diversos análisis y propuestas apuntando a alcanzar los requerimientos que originalmente se plantearon para el cliente. Estos análisis presentados, en conjunto con el conocimiento adquirido por este Equipo de Proyecto a partir de los relevamientos efectuados, así como las entrevistas mantenidas, devienen en una propuesta completa para la implementación de una red de sensores distribuida de Seguridad Informática a incorporarse a la Red de la Organización bajo estudio.

Este Equipo entiende que el despliegue de la misma dependerá de varios factores: objetivos de la Organización, capacidades técnicas, recursos de hardware y software, entre otros. Es de destacar que la concreción, independientemente de si se apunta a una solución de tipo OSS o de código propietario, dependerá del trabajo en conjunto de distintos actores: administradores de servidores, administradores de bases de datos, administradores de la red y personal de seguridad informática. Este trabajo conjunto es necesario puesto que en cada uno de los activos que se incorporen, tanto en los existentes como en los requeridos, será necesario el instalar y configurar piezas de software, activar funcionalidades, realizar configuraciones generales y documentaciones.

Por otra parte, el despliegue de una solución como ésta, y una vez puesta en marcha, exigirá la atención permanente y respuesta de los recursos humanos afectados a la tarea de la seguridad informática ante ciertas alarmas. Si bien esto reorienta la función, el fin último acaba siendo el desarrollo de una capacidad proactiva, aspecto que deberá saber gestionar el personal para poder anticiparse con éxito a posibles incidentes.

A su vez se logra otra importante capacidad: la trazabilidad de un incidente de seguridad informática. Al contar con los registros y eventos almacenados, frente a un incidente, el equipo investigador podrá trazar todos aquellos eventos relacionados con dicho incidente y tomar medidas a futuro una vez mitigado el mismo. Se tiene entonces una mejora continua en la tarea.

VII. Piloto

VII.1 – Introducción

A fin de establecer una prueba de concepto de la red de sensores de seguridad informática para la red de la DGSE del MDN, se determinó la instalación de un piloto correspondiendo a una versión reducida del planteamiento presentado en el documento Red Distribuida de Sensores de Seguridad Informática para el Ministerio de Defensa Nacional.

El piloto se implementó mediante la instalación de los siguientes componentes:

- Un SIEM con los componentes Prelude-IDS:
  - Prelude-Manager.
  - Prelude-LML.
  - Prelude-Correlator.
  - Prewikka.
- Un sistema de detección de intrusos de red (NIDS), utilizando el producto Snort IDS, con los componentes necesarios a fin de integrarse a la arquitectura Prelude.
- Un honeypot corriendo Nepenthes con los componentes necesarios a fin de integrarse a la arquitectura Prelude-IDS.

Adicionalmente, se requieren otros componentes tales como una base de datos MySQL[29], un servidor Web Apache, etc. [SNRP11]

VII.2 – Consideraciones preliminares

En el marco del alcance considerado para el proyecto, se efectuó el despliegue de la red piloto a fin de realizar una prueba de concepto de la solución planteada.

Además de los componentes instalados para el mismo, se utilizaron componentes ya presentes en la red del MDN, los cuales operaron ya sea como sensores en sí mismos, o como actores adicionales que permitieron la operación del piloto.

La red alcanzada por el piloto pertenece a la DGSE del MDN. Por motivos de confidencialidad de la red original presentaremos en la sección siguiente otra red genérica sobre la cual desplegaremos una propuesta similar a la originalmente realizada. [SNRP11]

[29] La instalación de los diversos componentes de Prelude-IDS exige algunos prerrequisitos para su correcta operación. Dos de estos componentes son un servidor de base de datos y un servidor web. Se opta entonces por MySQL y Apache, respectivamente. Estos son necesarios para el soporte de la arquitectura y de la consola Web de operación (Prewikka).

VII.3 – Red piloto

La red genérica sobre la cual plantearemos el despliegue de la solución piloto se presenta en la Figura 45:

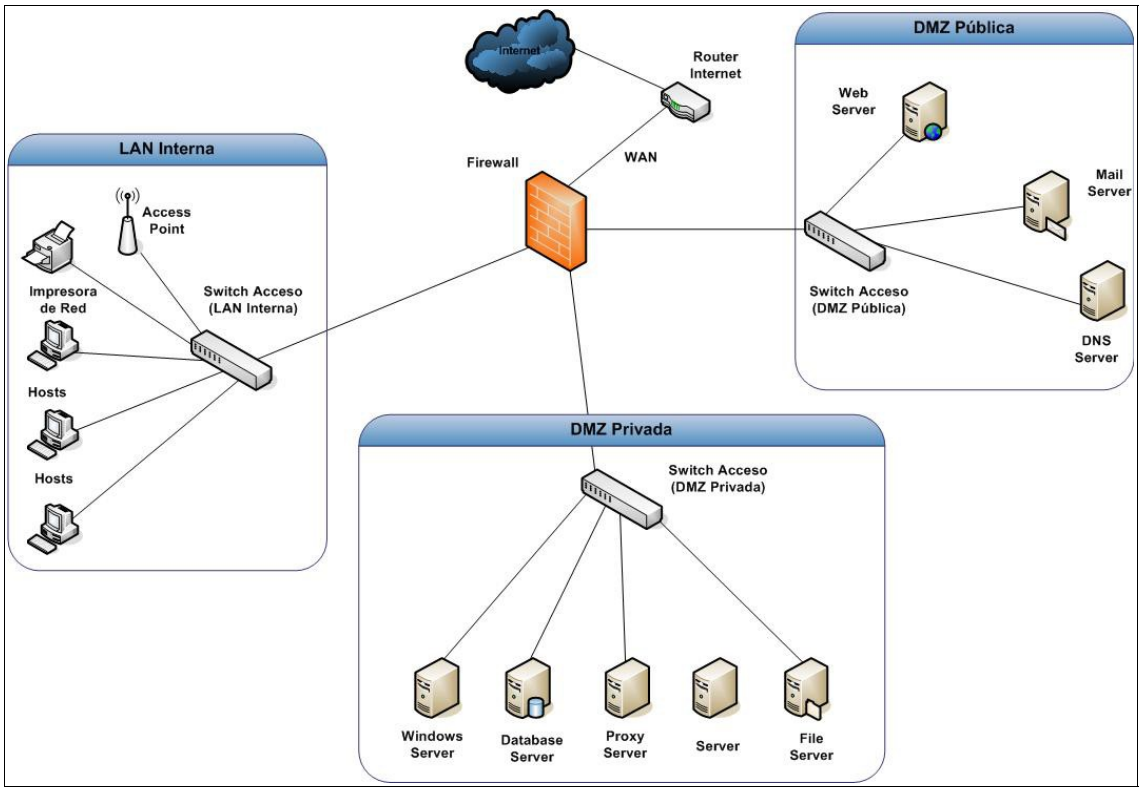


Figura 45 - Red genérica original.

Sobre dicha red se implantaron los componentes mencionados al comienzo de la presente sección así como otras configuraciones y sistemas de reporte adicionales.

En particular se utilizaron como sensores, además de los ya citados, algunos servidores de la DMZ Privada, DMZ Pública, y el firewall. Se consideraron sus respectivos logs, siendo estos fuentes de información y por ende actuando como sensores a los efectos de la red piloto.

Para el despliegue en la red del NIDS instalado (Snort IDS), fue necesaria la configuración de Port Mirror en un switch ya existente. Si bien por definición el propio switch es un elemento activo, a los fines del piloto su única función fue la de redireccionar una copia de los frames seleccionados para luego ser procesados por el NIDS.

Junto al NIDS se instala además en la nueva DMZ SensorNET un servidor al que denominamos SIEM, el cual poseerá los componentes centrales de la solución Prelude-IDS. También se decide la utilización de este mismo como servidor de base de datos para el almacenamiento de la información. En una solución de mayor porte, es recomendable que este último opere separado de los otros componentes de la solución utilizada.

Cabe destacar que si bien es posible el desarrollo de la solución Prelude-IDS en una modalidad “All-in-one”, la propuesta del piloto incluye el modelar una arquitectura distribuida. Si bien se optó por incluir dentro del SIEM al motor del servidor de base de datos, que ya contenía los componentes de Prelude-LML, Prelude-Correlator y Prewikka, esto no es necesariamente la única posibilidad. Los demás componentes de la solución, el Snort IDS y Nepenthes Honeypot, se instalaron en dos servidores independientes.

Si bien existen varios proyectos de Honeypot, se opta por la utilización de Nepenthes, dada su integración nativa al Prelude-IDS.

De esta forma llegamos al despliegue que se muestra en la Figura 46.

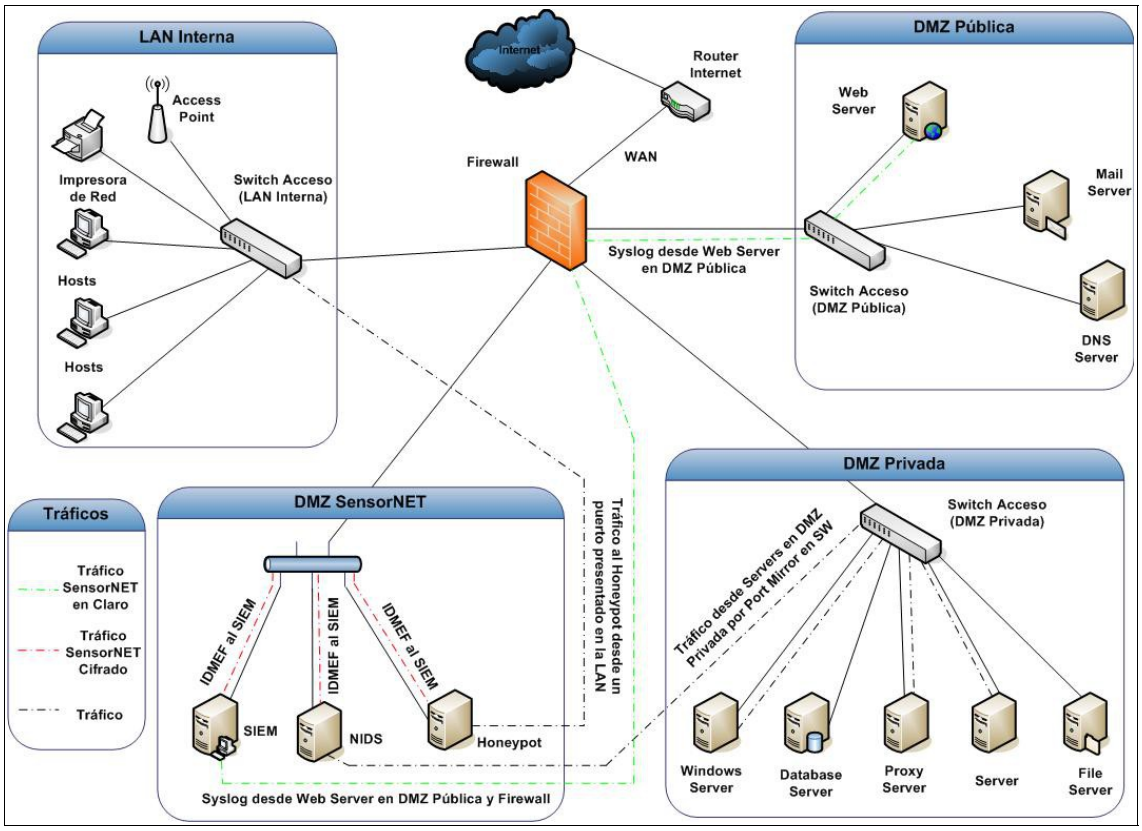


Figura 46 - Red genérica original más piloto.

VII.4 – Resultados obtenidos

El piloto se encontró operativo con todos sus componentes, desde el 29 de diciembre de 2011, hasta el 30 de enero de 2012.

Durante el mes transcurrido, se procedió a realizar visitas periódicas a fin de verificar la correcta operación del mismo.

Desde antes de la finalización de la instalación de todos los componentes, fue posible observar desde la Prewikka eventos desde los sensores ya disponibles. A fin de verificar la correcta operación de los sensores, se efectuaron algunas pruebas que

permitieron no sólo verificar la generación y recepción de los mensajes IDMEF, sino que también se verificó la capacidad de correlación del Prelude-Correlator.

Desde este punto de vista, el resultado fue más que satisfactorio, por lo que se verificó que la solución es totalmente válida y es posible el despliegue de una red en producción basada en la herramienta elegida. [SNRP11]

A efectos de clarificar como trabaja Prelude-IDS y las vistas que ofrece su interfaz web es que se muestran a continuación alguna de ellas.

Las primeras corresponden a las desplegadas en la pantalla inicial. Allí Prewikka desplegará el panel de “Events” con aquellos eventos recibidos en la última hora, bajo la lengüeta de “Alerts”. El panel tiene una opción de “Refresh” cada un minuto por defecto.

Además de la lengüeta de “Alerts”, el panel posee la de “CorrelationAlerts”, la cual despliega aquellas alertas específicamente generadas por el Prelude-Correlator. Como viéramos en la discusión de dicho componente realizada en el estudio comparativo de los diferentes SIEM, el mismo genera alertas a partir del análisis de los eventos recibidos desde los diversos sensores presentes en la red y registrados en el sistema.

Finalmente, la lengüeta “ToolAlerts” se refiere a la detección de herramientas de ataque o similares encontradas por algunos de los sensores (referidos como “analyzer” en la documentación). La idea es agrupar las alertas vistas en la lengüeta “Alerts” bajo la “tool” que las generó.

Estas tres lenguetas se muestran en la Figura 47, Figura 48 y Figura 49:

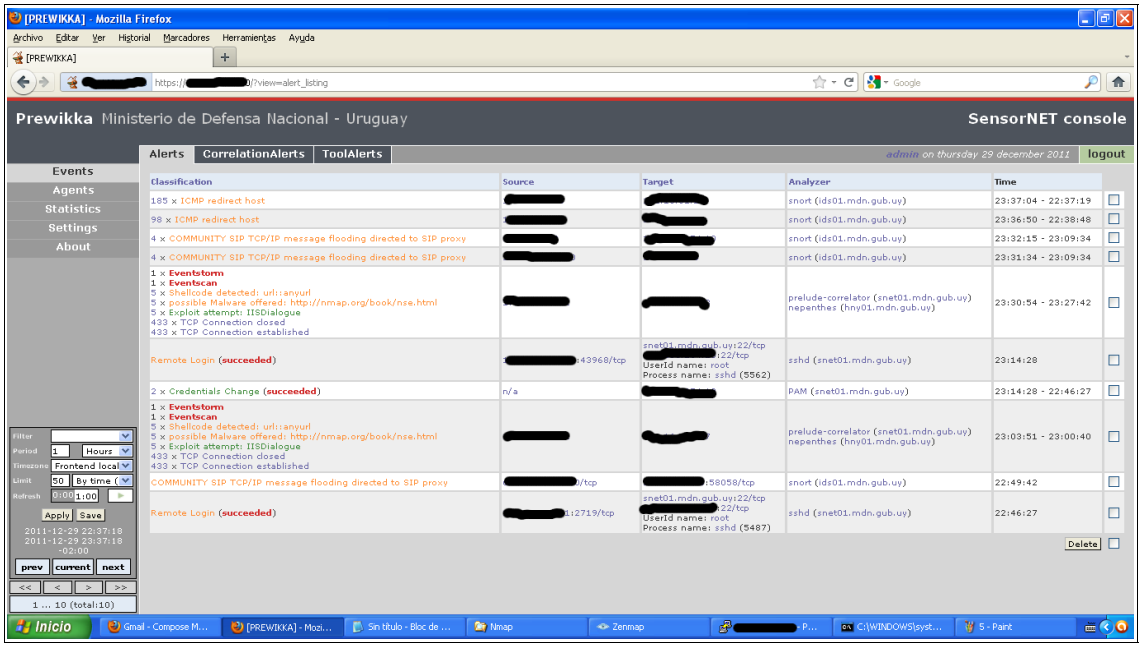


Figura 47 - Panel principal con sus pestañas de alertas.

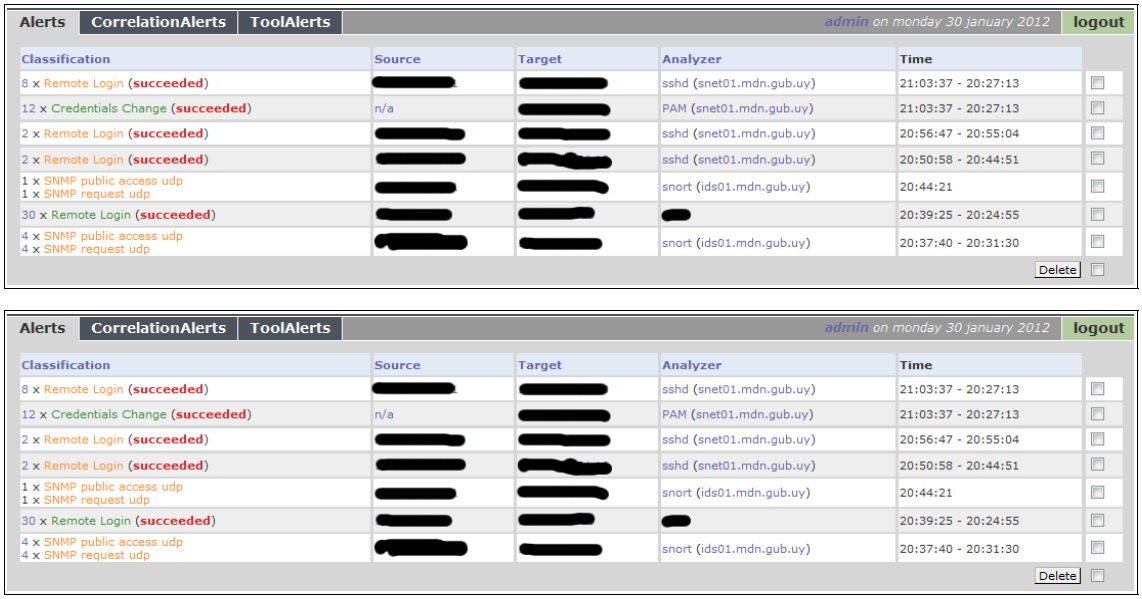


Figura 48 - Algunos de los eventos mostrados.

Figura 49 - Vista de eventos de correlación.

Continuado la presentación de la Prewikka, pasaremos al panel “Agents”. En el mismo se despliegan todos los sensores directamente conectados al Prelude-Manager.

La agrupación se realiza por host y, dentro del mismo, por el tipo de sensor o “analyzer” desplegado. Nótese que aquí no encontraremos, por ejemplo, el sensor correspondiente al Syslog Server de un servidor Linux, sino al Prelude-LML que recibe y procesa los mensajes Syslog de dicho servidor.

La Figura 50 muestra los diferentes “Agents” desplegados para el piloto.

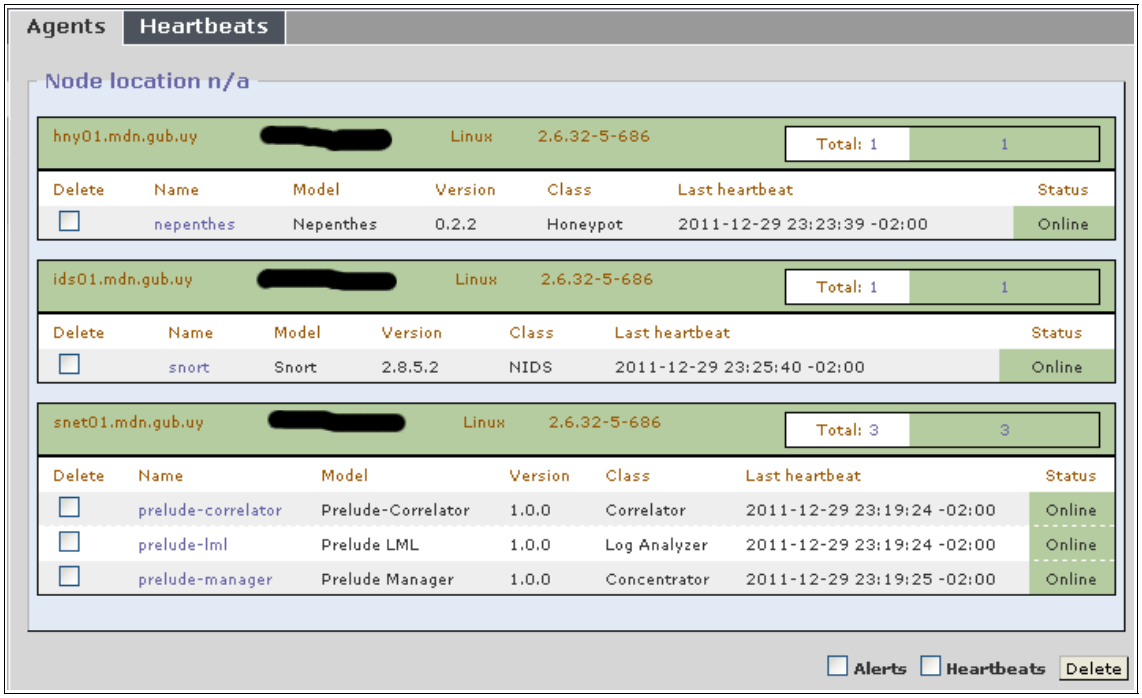


Figura 50 - Detalle del despliegue de los agentes y su tipo.

El siguiente panel corresponde a “Statistics”. Mediante el mismo es posible visualizar en forma gráfica las diferentes alertas procesadas por el piloto. Dicha visualización se realiza según el tipo de “analyzer”, dirección IP de origen o destino del evento detectado, nivel de criticidad del evento o tiempo histórico.

Esta visualización es realizada para los eventos de la última hora, aunque es posible realizar consultas más amplias mediante las opciones disponibles.

Algunos ejemplos de los reportes se muestran a continuación.

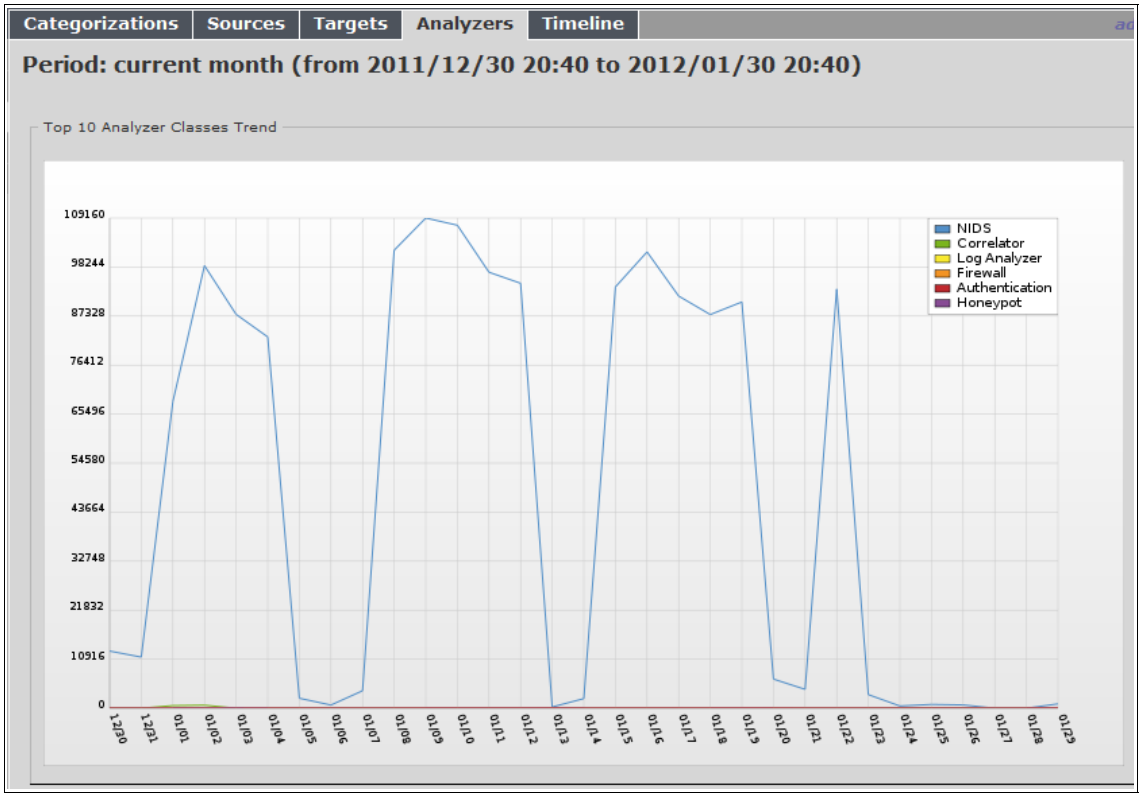


Figura 51 - Reporte estadístico según tipo de sensor

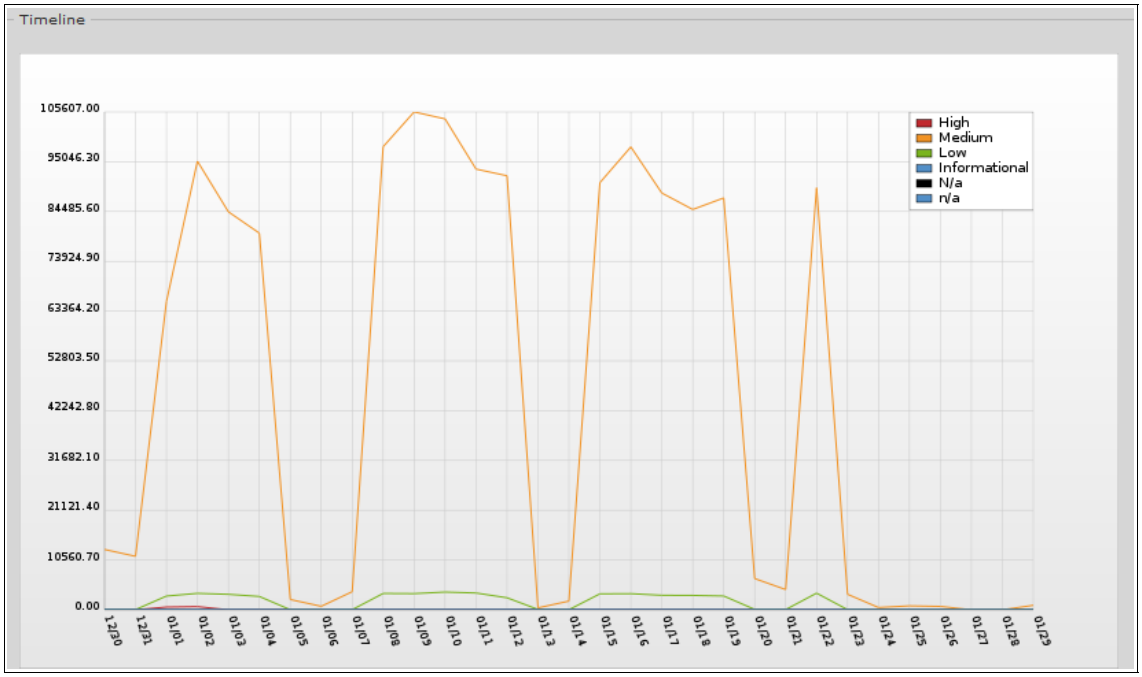


Figura 52 - Reporte mensual según nivel de criticidad de los eventos

VII.5 – Conclusiones de las pruebas Piloto

La prueba de concepto realizada mediante el piloto fue altamente satisfactoria. No sólo fue posible el despliegue de los nuevos sensores basados en Prelude-IDS, sino que también fue posible utilizar las capacidades de los activos presentes en la red del MDN.

Si bien el número de sensores no fue el que correspondería a una red de producción, considerando como tal la presentada en la Figura 44, fue posible generar una arquitectura distribuida y escalable, siendo sólido y consistente el comportamiento de los componentes de la solución elegida.

Fue posible colectar más de 11.000 eventos en un sólo día, sin que esto implicara degradación en la performance de la solución. Tampoco se tuvieron indicadores adversos en el caso del switch utilizado para el Port Mirror, por lo que también se validó la operación dentro de las especificaciones de servidores analizados.

El único “debe” en este sentido, fue el Honeypot. Como se indica en [SNRP11], se requeriría más investigación a futuro a fin de lograr un componente estable no sólo en su comportamiento, sino también en la integración con la solución SIEM.

Si bien la implementación de una red de sensores de seguridad informática es una tarea que requiere ciertos “skills”, se necesita que el proceso sea fácilmente comprendido, ejecutado y documentado.

Por otra parte, un tema que no debe dejarse pasar por alto, es el adecuado “tunning” de MySQL. En nuestro despliegue mínimo, fue claro que dejar la instalación por defecto, afecta adversamente la respuesta en el caso de realizar consultas intensivas sobre los eventos pasados. La lección aprendida es que deberá considerarse en forma especial, todo lo relativo al soporte de la base de datos, tanto en un su configuración de software, como respecto a la configuración de hardware donde esta sea ejecutada. [SNRP11]

Respecto de los eventos analizados, si bien es necesario refrendar que los mismos no corresponden a incidentes, tendemos a suponer que los mismos pueden considerarse como falsos positivos, siendo necesario el correcto ajuste a las “reglas del negocio” por parte de la solución. En este sentido, es clara la prevalencia en los eventos del Snort IDS, por lo cual es esperable la ocurrencia de falsos positivos en este tipo de soluciones.

VIII. Conclusiones Finales

A la hora de redactar las conclusiones finales del proyecto, el Equipo se plantea dos vertientes:

1. Las relacionadas con el trabajo de investigación, análisis y concreción de una prueba de concepto de la temática elegida.
2. Las relacionados con el desarrollo de un proyecto en forma general.

En el primer caso, el trabajo logró desarrollar, mejorar, y uniformizar entre todos los integrantes del Equipo, los conocimientos relacionados a Seguridad Informática, focalizando en el desarrollo de la comprensión de las trazas digitales que dejan un Sistema Informático.

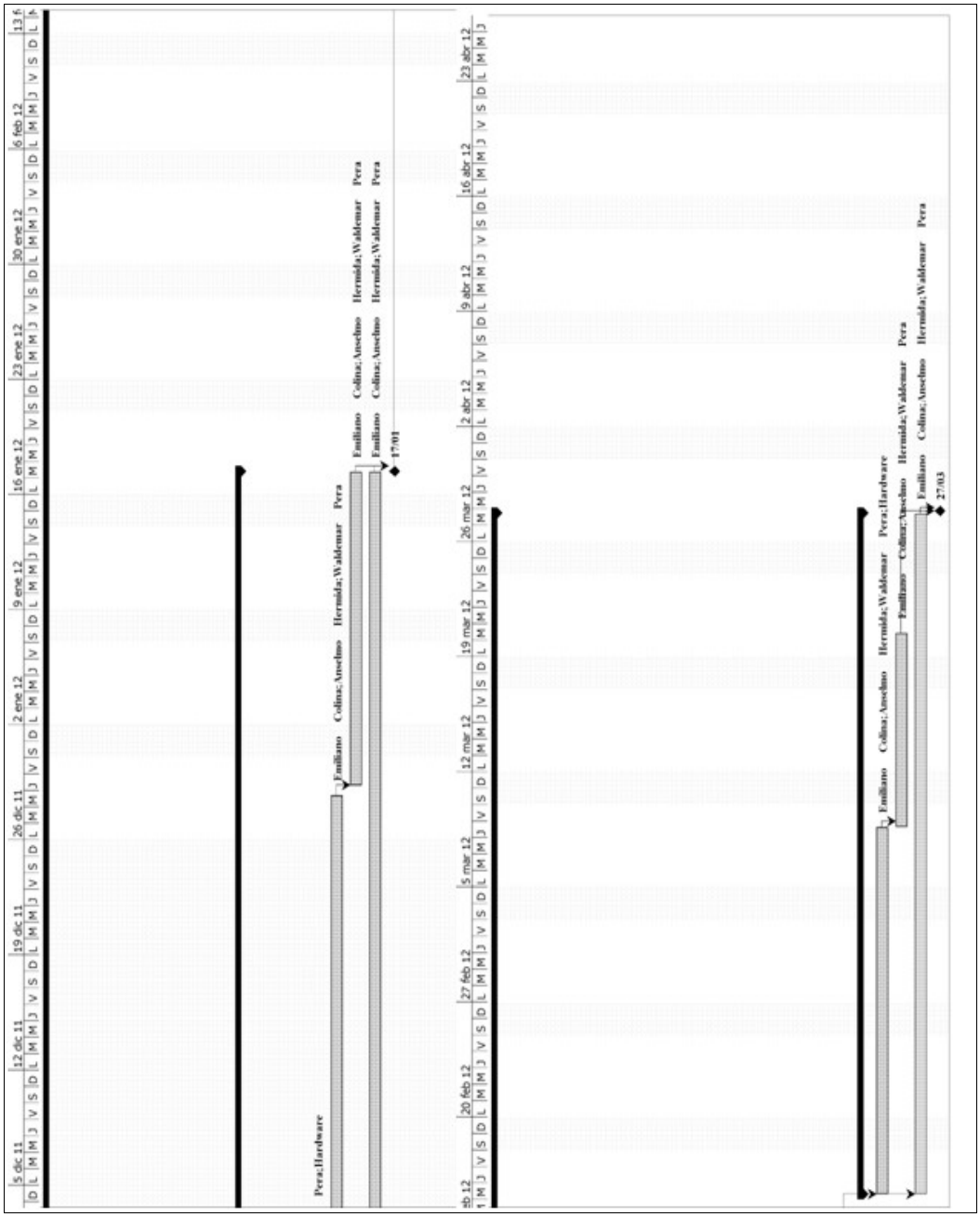
En el segundo punto, si bien los integrantes del Equipo tiene ya varios años de ejercicio profesional, nos enfrentó a situaciones que si bien fueron previstas en el análisis de riesgo, impactaron significativamente en los tiempos previstos. La interacción con el cliente y sus tiempos fueron tales que consumieron la gran parte de los “buffers” previstos. El adicional en nuestro caso fue el marco de Acuerdo de Confidencialidad, el cual llevó a que muchos de los tiempos fueran mayores dado la necesidad de un número mayor de revisores y de un mayor celo de los mismos.

Si bien el proyecto logró la mayoría de sus cometidos dentro de los plazos establecidos, exigió un período de prórroga para lograr su completitud en la fase final.

IX. Anexo: Planificación del Proyecto

Se presenta a continuación el listado de tareas y Diagrama de Gantt relativos a la planificación original realizada por el Equipo para el desarrollo del Proyecto SensorNET.

	Nombre	Duración	Inicio	Terminado
1	**Proyecto "SensorNET**	**552 days**	**2/05/11 8:00**	**27/03/12 17:00**
2	**Fase 1 - Estudio de "estado del arte"**	**190 days**	**2/05/11 8:00**	**30/08/11 10:00**
3	i) Investigación en tecnologías de sensores	60 days	2/05/11 8:00	1/06/11 13:00
4	ii) Listar y catalogar los tipos de herramientas posibles a implementar	10 days	1/06/11 13:00	7/06/11 10:00
5	iii) Estudio individual de herramientas destacadas	60 days	7/06/11 10:00	7/07/11 15:00
6	iv) Relevamiento de la red del MDN alcanzable por este proyecto	30 days	2/05/11 8:00	17/05/11 10:00
7	v) Comparación de las herramientas antes estudiadas y su viabilidad a la realidad de la red d...	30 days	7/07/11 15:00	12/08/11 17:00
8	vi) Elección de herramientas y su justificación	30 days	15/08/11 8:00	30/08/11 10:00
9	vii) Generación de documento entregable con el estudio del estado del arte de las alternativa...	190 days	2/05/11 8:00	30/08/11 10:00
10	Hito de cierre	0 days	30/08/11 10:00	30/08/11 10:00
11	**Fase 2 - Diseño Conceptual**	**195 days**	**15/09/11 8:00**	**17/01/12 9:00**
12	i) Especificación de alcance del piloto: equipamiento involucrado y duración del mismo.	30 days	15/09/11 8:00	30/09/11 10:00
13	ii) Coordinación con actores involucrados según especificación anterior.	15 days	30/09/11 10:00	7/10/11 16:00
14	iii) Autorización para instalación de equipamiento core del piloto.	15 days	7/10/11 16:00	17/10/11 13:00
15	iv) Instalación, configuración y activación del equipamiento core del piloto.	60 days	17/10/11 13:00	16/11/11 17:00
16	v) Presentación de un esquema de la red sobre la que se basará la solución a implementar.	40 days	17/11/11 8:00	28/12/11 17:00
17	vi) Diagrama modular de la solución.	35 days	29/12/11 8:00	17/01/12 9:00
18	vii) Generación de documento entregable con el modelo de implementación de una red distrib...	195 days	15/09/11 8:00	17/01/12 9:00
19	Hito de cierre	0 days	17/01/12 9:00	17/01/12 9:00
20	**Fase 3 - Piloto y Defensa**	**80 days**	**15/02/12 8:00**	**27/03/12 17:00**
21	i) Liberación formal del piloto.	45 days	15/02/12 8:00	8/03/12 16:00
22	ii) Generación de documento entregable con el informe de evaluación del piloto y las respecti...	20 days	8/03/12 16:00	20/03/12 11:00
23	iii) Generación de los entregables académicos: documentación del proyecto, paper y poster.	80 days	15/02/12 8:00	27/03/12 17:00
24	Hito de cierre	0 days	27/03/12 21:00	27/03/12 17:00


Pera;Hardware
Emiliano
Colina;Anselmo
Hermida;Waldemar
Pera
Emiliano
Colina;Anselmo
Hermida;Waldemar
Pera
17/01
Emiliano
Colina;Anselmo
Hermida;Waldemar
Pera;Hardware
Emiliano
Colina;Anselmo
Hermida;Waldemar
Pera
Emiliano
Colina;Anselmo
Hermida;Waldemar
Pera
27/03

X. Glosario

Actualización de archivos de firmas: El fabricante de la solución investiga y crea firmas para nuevas amenazas y comportamiento malicioso conforme se descubre, y publica nuevas firmas regularmente. En general el fabricante notifica cuando está disponible una nueva actualización para ser descargada. [Cis05]

Adware: Un programa de la clase adware es cualquier programa que automáticamente se ejecuta, muestra o baja publicidad web al computador después de instalar el programa o mientras se está utilizando la aplicación. 'Ad' en la palabra 'adware' se refiere a 'advertisement' (anuncios) en inglés.

Algunos programas adware son también shareware, y en estos los usuarios tiene la opción de pagar por una versión registrada o con licencia, que normalmente elimina los anuncios.

Archivo de firmas: Un archivo de firma comprende un paquete cuya intención es servir como una actualización a las bases de datos de firmas que reside en un IPS o IDS. Esta base de datos es utilizada por el IPS o IDS para comparar el tráfico de red contra los patrones de datos contenidos en la librería de archivos de firma. El IPS/IDS utiliza esta comparación para detectar una presunta conducta de tráfico malicioso. [Cis05]

CapEx: CAPital EXpenditures (CAPEX o capex o gastos de capital) son erogaciones o inversiones de capital que crean beneficios. Una Capex se realiza cuando un negocio invierte tanto en la compra de un activo fijo como para añadir valor a un activo existente con una vida útil que se extiende más allá del año imponible. Los CAPEX son utilizados por una compañía para adquirir o mejorar los activos fijos tales como equipamientos, propiedades o edificios industriales. En contabilidad, un CAPEX es añadido a una cuenta de activos (capitalización) y por ende incrementando el valor base del activo (el coste o valor de un activo ajustado por motivos impositivos)

En el contexto utilizado en el presente documento, es el costo de adquisición de activos.

Cracker: El término cracker (del inglés crack, romper) se utiliza para referirse a las personas que rompen o quiebran algún sistema de seguridad.

Crimeware: Tipo de programa de computadora diseñado específicamente para cometer crímenes del tipo financiero o similar, intentando pasar desapercibido por la víctima. Por extensión, también hace referencia a aplicaciones web con iguales objetivos.

DoS/DDoS: En seguridad informática, un ataque de denegación de servicio, también llamado ataque DoS (de las siglas en inglés “Denial of Service”), es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la

conectividad de la red por el consumo del ancho de banda de la red o sobrecarga de los recursos computacionales del sistema de la víctima.

Se genera mediante la saturación de los puertos con flujo de información, haciendo que la infraestructura se sobrecargue y no pueda seguir prestando servicios, por eso se le denomina "denegación", pues hace que el servidor no dé abasto a la cantidad de solicitudes. Esta técnica es usada por los llamados Crackers para dejar fuera de servicio a servidores objetivo.

Una ampliación del ataque DoS es el llamado ataque distribuido de denegación de servicio, también llamado ataque DDoS (de las siglas en inglés “Distributed Denial of Service”) el cual se lleva a cabo generando un gran flujo de información desde varios puntos de conexión.

La forma más común de realizar un DDoS es a través de una botnet, siendo esta técnica el ciberataque más usual y eficaz por su sencillez tecnológica.

Firma: Tal como una huella digital, la firma identifica a un worm (gusano), virus, anomalía de protocolo o tráfico malicioso específico. [Cis05]

Gusano (Worm): Un gusano es un malware que tiene la propiedad de replicarse a sí mismo. Los gusanos utilizan las partes automáticas de un sistema operativo que generalmente son invisibles al usuario.

A diferencia de un virus, un gusano no precisa alterar los archivos de programas, sino que reside en la memoria y se duplica a sí mismo. Los gusanos casi siempre causan problemas en la red (aunque sea simplemente consumiendo ancho de banda), mientras que los virus siempre infectan o corrompen los archivos de la computadora que atacan.

Hardening: En computación, “hardening” es el proceso de asegurar un sistema reduciendo la superficie de vulnerabilidad. Se dice que la superficie de vulnerabilidad es mayor cuanto mayor sea el número de funciones que debe cumplir el sistema. Desde esta perspectiva, un sistema de función dedicada es más seguro que uno múlti función.

La reducción de la superficie de vulnerabilidad, implica la reducción de los vectores de ataque disponibles. Típicamente, este incluye la remoción de software innecesario para la función asignada al sistema, los “usernames” o “logins” innecesarios, o la remoción de servicios no utilizados.

Hijacking: Hijacking significa "secuestro" en inglés y en el ámbito informático hace referencia a toda técnica ilegal que lleve consigo el adueñarse o robar algo (generalmente información) por parte de un atacante. Es por tanto un concepto muy abierto y que puede aplicarse a varios ámbitos, de esta manera podemos encontramos con el secuestro de sesiones en un portal web, conexiones de red, sesiones de terminal, servicios y un largo etcétera en cuanto a servicios informáticos se refiere.

Log (Registro): La palabra “log” es un término anglosajón, equivalente a la palabra bitácora en lengua castellana (es habitual encontrar “bitácora” en publicaciones sobre informática de origen español). Sin embargo, se utiliza en los países de habla hispana como un anglicismo derivado de las traducciones del inglés en la jerga informática.

Un log es un registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática, es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué (who, what, when, where y why) un evento ocurre para un dispositivo en particular o aplicación.

Ocultación de huellas: u ocultación de rastros es la actividad ejecutada por un atacante, una vez producida la intrusión, para pasar desapercibido en el sistema.

OpEx: OPerational EXpenditures (OPEX o opex o gastos de operación) es una herramienta para el cálculo de gastos operativos.
En un estado de resultados, "los gastos de operación" es la suma de los gastos de funcionamiento de una empresa por un período de tiempo, como un mes o un año.

P2P: Una red Peer-to-Peer o red de pares o red entre iguales o red entre pares o red punto a punto (P2P, por sus siglas en inglés) es una red de computadoras en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre estos nodos.

Ransomware: Ransomware es un malware generalmente distribuido mediante SPAM y que mediante distintas técnicas imposibilita al dueño de un documento acceder al mismo. El modo más comúnmente utilizado es cifrar con clave dicho documento y dejar instrucciones al usuario para obtenerla, posterior al pago de "rescate".

Rogue software: El Rogue software (en español, software bandido o también falso antivirus) es un tipo de programa informático malintencionado cuya principal finalidad es hacer creer que una computadora está infectada por algún tipo de virus, induciendo a pagar una determinada suma de dinero para eliminarlo.

Rootkit: Un rootkit es una herramienta o un grupo de ellas, que tiene como finalidad esconderse a sí misma y esconder otros programas, procesos, archivos, directorios, claves de registro, y puertos que permiten al intruso mantener el acceso a un sistema para remotamente comandar acciones o extraer información sensible. Existen rootkits para una amplia variedad de sistemas operativos, como GNU/Linux, Solaris o Microsoft Windows.

Sandbox: Sandbox, palabra que del inglés significa caja de arena («Sand»+«box»), es un sistema informático de aislamiento de procesos, mediante el cual, se pueden ejecutar distintos programas con seguridad y de manera separada. A menudo se utiliza para ejecutar código nuevo, o software de dudosa confiabilidad, con objeto de evitar la corrupción de datos del sistema en donde estos se ejecutan.

SPAM: Se llama SPAM, correo basura o mensaje basura a los mensajes no solicitados, no deseados o de remitente no conocido, habitualmente de tipo publicitario, generalmente enviados en grandes cantidades (incluso masivas) que perjudican de alguna o varias maneras al receptor. La acción de enviar dichos

mensajes se denomina spamming. La palabra SPAM proviene de la segunda guerra mundial, cuando los familiares de los soldados en guerra les enviaban comida enlatada. Entre estas comidas enlatadas estaba “Spam” una carne enlatada, que en los Estados Unidos era y es muy común.

Spyware: Un spyware es un programa que funciona dentro de la categoría malware. Se instala furtivamente en un computador para recopilar información sobre las actividades realizadas en éste.

Snifer: Software destinado para detectar y capturar tramas de tráfico que viaja por la red. Los sniffers o husmeadores tienen diversos usos como monitorear redes para detectar y analizar fallos o ingeniería inversa de protocolos de red. También es habitual su uso para fines maliciosos, como robar contraseñas, interceptar mensajes de correo electrónico, espiar conversaciones de chat, etc.

Troyano: En informática, se denomina troyano o caballo de Troya (traducción literal del inglés “Trojan horse”) a un software malicioso que se presenta al usuario como un programa aparentemente legítimo e inofensivo pero que al ejecutarlo ocasiona daños.

Los troyanos pueden realizar diferentes tareas, pero, en la mayoría de los casos, crean una puerta trasera (en inglés “backdoor”) que permite la administración remota, del sistema afectado, a un usuario no autorizado.

Un troyano no es estrictamente un virus informático, y la principal diferencia es que los troyanos no propagan la infección a otros sistemas por sí mismos.

Tunelización: Es la utilización de ciertos protocolos de red que encapsulan a otro u otros protocolos. La técnica de tunelizar se suele utilizar para trasportar un protocolo determinado a través de una red que, en condiciones normales, no lo aceptaría. Otro uso de la tunelización de protocolos es la creación de diversos tipos de redes privadas virtuales.

Virus: Un virus es un malware que tiene por objeto alterar el normal funcionamiento de la computadora, sin el permiso o el conocimiento del usuario. Los virus, habitualmente, reemplazan archivos ejecutables por otros infectados con el código de este. Los virus pueden destruir, de manera intencionada, los datos almacenados en un ordenador, aunque también existen otros más inofensivos, que solo se caracterizan por ser molestos.

XI. Referencias

[SNEA11] SensorNET – Documento “Estado del Arte en redes de Sensores de Seguridad Informática” – 2011.

[SNEH11] SensorNET – Documento “Estudio de herramientas para redes de sensores de Seguridad Informática”. - 2011.

[SNRD11] SensorNET – Documento “Red Distribuida de Sensores de Seguridad Informática para el Ministerio de Defensa Nacional” - 2011.

[SNRP11] SensorNET – Documento “Reporte del Piloto Prelude-IDS”, SIEM” - 2011.

[GMQ11] http://www.arcsight.com/collateral/whitepapers/Gartner_Magic_Quadrant_2011.pdf

[ZPSGT04] http://www.pcworld.com/article/116841/zombie_pcs_silent_growing_threat.html, Jul 9, 2004.

[Ard08] Prof. Lic. Julio Ardita, “Módulo de Seguridad Informática”. Mar. 2008. Disponible en: http://www.cybsec.com/upload/ESPE_IDS_vs_IPS.pdf

[Baum02] R. Baumann, C. Plattern, “White Paper: Honeypots”. Feb. 2002. Disponible en: http://security.rbaumann.net/download/whitepaper.pdf

[BauPla02] R. Baumann, C. Plattern, “Honeypots”. Tesis de grado. Feb. 2002. Disponible en: security.rbaumann.net/download/diplomathesis.pdf

[Bor01] Lic. Cristian Borghello, “Seguridad Informática, Sus Implicancias e Implementación”. Tesis de grado. Set 2001. Disponible en: http://www.segu-info.com.ar/tesis/

[Bot05] Nicolás Botero Arana, “Modelo de gestión de Seguridad con soporte a SNMP”. Jun. 2005. Disponible en:
http://www.javeriana.edu.co/biblos/tesis/ingenieria/Tesis190.pdf

[Cis05] Cisco Systems, “Servicios Cisco para Sistemas de Prevención de Intrusiones”. 2005. Disponible en:
http://www.cisco.com/web/LA/productos/servicios/docs/IPS_external_qa_clients_Spanish.pdf

[Coh87] Fred Cohen, "Computer Viruses Theory and Experiments". Computers and Security, vol. 6, pp. 22-35.1987

[CoLaOrPu05] Córdoba Jonathan, Laverde Ricardo, Ortiz Diego, Puentes Diana "Los IDS y los IPS. Una comparación práctica". 2005. Disponible en: http://www.criptored.upm.es/guiateoria/gt_m142w.htm

[CoLPBST01] Fred Cohen, Dave Lambert, Charles Preston, Nina Berry, Corbin Stewart, and Eric Thomas, "A Framework for Deception", IFIP-TC11, 'Computers and Security'. 2001. Disponible en:
http://all.net/journal/deception/Framework/Framework.html

[Danf06] R. Danford: "2nd Generation Honeyclients", SANS Internet Storm Center. 2006. Disponible en:
handlers.dshield.org/rdanford/pub/Honeyclients_Danford_SANSfire06.pdf

[DeCu07] H. Debar, D. Curry, "The Intrusion Detection Message Exchange Format (IDMEF)", rfc4765. Mar. 2007.

[FSI11] Grupo de Seguridad Informática, INCO – FING - UDELAR, "Fundamentos de Seguridad Informática – Seguridad en Redes (parte 2)", 2011. Disponible en:
http://www.fing.edu.uy/inco/cursos/fsi/teorico/2011/FSI-2011-Aplicaciones-II.pdf

[GaLiWa08] Moses Garuba, Chunmei Liu, and Nicki Washington, "A Comparative Analysis of Anti-Malware Software, Patch Management, and Host-Based Firewalls in Preventing Malware Infections on Client Computers", Fifth International Conference on Information Technology: New Generations (itng 2008). 2008. IEEE Computer Society

[GarPer04] Joaquín García Alfaro, Xavier Perramón Tornil, "Aspectos avanzados de seguridad en redes", Jul. 2004. ISBN: 84-9788-212-1

[Ger08] R. Gerhards "The Syslog Protocol, draft-ietf-syslog-protocol-23", Mar. 2008. Disponible en: http://tools.ietf.org/html/draft-ietf-syslog-protocol-23

[HaDeWe10] M. Han, X. Dewu, C. Wei, "Improvement and implementation of IDMEF Data Model", Second International Conference on Computer Modeling and Simulation, 2010.

[HoeSteMon03] C. Hoepers, K. Steding-Jessen, A. Montes, "Honeynets Applied to the CSIRT Scenario", 2003.
Disponible en:http://www.honeynet.org.br/papers/hnbr-first2003.pdf

[Ken03] Kenneth E. Nawyn, "A Security Analysis of System Event Logging with Syslog", May. 2003. Disponible en:
http://www.sans.org/reading_room/whitepapers/logging/security-analysis-system-event-logging-syslog_1101

[KerBro05] John E. Kerivan and Kenneth Brothers, "Self-Defending Security Software", IEEE Military Communications Conference, 2005. IEEE Computer Society

[Kye03] Honeynet Project, “Know Your Enemy: Defining Virtual Honeynets - Different types of Virtual Honeynets.” Ene. 2003. Disponible en:
http://old.honeynet.org/papers/virtual/

[Kye05] Honeynet Project, “Know Your Enemy: GenII Honeynets - Easier to deploy, harder to detect, safer to maintain”. May. 2005. Disponible en:
http://old.honeynet.org/papers/gen2/index.html

[Kye06] Honeynet Project, “Know Your Enemy: Honeynets - What a honeynet is, its value, overview of how it works, and risk/issues involved”. May. 2006.
Disponible en: http://old.honeynet.org/papers/honeynet/index.html

[KumCon10] Brijesh Kumar and Constantine Katsinis, “A Network Based Approach to Malware Detection in Large IT Infrastructures”, Ninth IEEE International Symposium on Network Computing and Applications, 2010. IEEE Computer Society

[Mac] Miguel Machado, “Syslog Protocolo y Servicios” Disponible en: http://www.fing.edu.uy/~mmachado/docs/Syslog-MiguelMachado.pdf

[Neu49] John Von Neuman, “Theory of Self-Reproducing Automata”, Part 1: Transcripts of lectures given at the University of Illinois, Dec. 1949, Editor: A. W. Burks, University of Illinois. 1966

[Per02] Julio Perez, “Syslog y Archivos de registro de eventos”. 2002. Disponible en: http://iie.fing.edu.uy/ense/asign/admunix/logs.htm

[PoDD03] F. Pouget, M. Dacier, H. Debar. White Paper: “Honeypot, Honeynet, Honeytoken: Terminological issues”. Set. 2003.
Disponible en: http://www.eurecom.fr/util/publidownload.fr.htm?id=1275

[Ron03] Ronald Menardo, “Securing Network Management Information”. Set. 2003. Disponible en: http://www.giac.org/paper/gsec/3542/securing-network-management-information/105764

[Sar08] Juan Pablo Sarubbi, “Seguridad Informática. Técnicas de defensa comunes bajo variantes del sistema operativo Unix”. 2008. Disponible en: http://books.google.com.uy/

[Spit02] L. Spitzner, “Honeypots: Tracking Hackers”. Addislon-Wesley, ISBN from-321-10895-7. 2002.

[Spit03] L. Spitzner, “Honeytokens: The Other Honeypot”. 2003. Disponible en: http://www.securityfocus.com/infocus/1713

[Spit03b] L. Spitzner, “Honeypots: Definitions and Value of Honeypots”. May 2003. Disponible en: http://www.tracking-hackers.com/papers/honeypots.html

[Stol89] C. Stoll, “The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage”. Doubleday, ISBN 0-385-24946-2. 1989.

[TorGar05] Juan Carlos Torres, Richard Garcia Rondón “Control, Administración e integridad de Logs”. Ago. 2005. Disponible en:
http://www.wikilearning.com/monografia/control_administracion_e_integridad_de_logs-que_es_un_log/3485-2

[WuShi06] Yanjun Wu and Wenchang Shi, “Portal Monitoring Based Anti-Malware Framework:Design and Implementation”, IEEE International Performance Computing and Communications Conference, 2006. IEEE Computer Society

[ZwiCoop01] Elizabeth D. Zwicky, Simon Cooper & D. Brent Chapman “Building Internet Firewalls. 2nd ed.”. O’Reilly Press. 2001

[MiHaVa11] David Miller, Shon Harris, Allen Harper, and Stephen VanDyke, “Security Information and Event Management (SIEM) Implementation.” McGraw-Hill – Network Pro Library 2011

[ASP11] ArcSight, Supported Products Oct, 2011, disponible en:
http://www.arcsight.com/collateral/ArcSight_Supported_Products.pdf

[ASL10] Presentación de ArcSight en el Technology Day Genève, 17 Mars 2010

[QRPSD] http://q1labs.com/products/supported-devices.aspx

[QRPGS] http://q1labs.com/products/qradar-siem.aspx

[QRSPL] http://q1labs.com/products/qradar-siem/product-literature.aspx

[KaMu03] Dominique Karg, Jesus Muñoz, “Descripción general del sistema OSSIM”, oct 2003. Disponible en: http://sourceforge.net/projects/os-sim/ , archivo OSSIM-desc-es.pdf

[AVPSvOS10] AlienVault Professional SIEM and AlienVault Open Source SIEM (OSSIM), de AlienVault-Professional-SIEM-vs-OSSIM.pdf disponible en

[BCPAG10] Building Collector Plugins – Admin Guide 2010, AlienVault Building Collector Plugins.pdf, disponible en

[ASSD10] AlienVault SIEM Sistem Description v1.0 2010-2011, archivo AlienVault_Unified_System_Description_1.0 disponible en http://www.alienvault.com

[ASCO] http://alienvault.com/community

[ASRD] http://alienvault.com/resources/documentation

[ASPU] http://alienvault.com/products/unified-siem

[Yas09] Curt Yasm, “Prelude as a Hybrid IDS Framework” – SANS Institute 2009, disponible en:

http://www.sans.org/reading_room/whitepapers/awareness/prelude-hybrid-ids-framework_33048

[PreMU] Manual User, disponible en:
https://dev.prelude-technologies.com/wiki/prelude/ManualUser

[PreLC] Prelude Logs Compatibility, disponible en: http://www.prelude-technologies.com/en/development/documentation/compatibility/index.html

[PreCo] Prelude components, disponible en:
http://www.prelude-technologies.com/en/development/documentation/prelude-components/index.html

[PreLml] Prelude LML, disponible en:
https://dev.prelude-technologies.com/wiki/prelude/PreludeLml

[PreIm] Prelude Import, disponible en:
https://dev.prelude-technologies.com/wiki/prelude/PreludeImport

[PrePP] https://trac.prelude-ids.org/projects/prelude/wiki

[Pre MP] https://trac.prelude-ids.org/wiki/prelude/ManualPrewikka

[PreMU] https://trac.prelude-ids.org/wiki/prelude/ManualUser

[PreDO] https://trac.prelude-ids.org/wiki/prelude/DatabaseOptimisation

[PrePC] https://trac.prelude-ids.org/wiki/prelude/PreludeCompatibility

[PrePCo] https://trac.prelude-ids.org/wiki/prelude/PreludeComponents

SensorNET, una visión unificada

Emiliano Colina, Anselmo Hermida, Waldemar Pera

***Resumen*—El presente artículo tiene como objetivo la presentación de los principios básicos de funcionamiento y aplicación de una red de sensores de seguridad informática. En el mismo se realiza, brevemente, una presentación de diferentes SIEM (Security Information and Event Management). Finalmente, se describirá una aplicación práctica de una red de sensores.**

***Palabras claves*—SIEM, IDS, Prelude-IDS, HP ArcSight, MDN.**

I. Introducción

Es habitual escuchar en las noticias sobre diferentes situaciones que ponen en duda las capacidades de respuesta de las organizaciones ante brechas en su seguridad informática. No hace mucho tiempo atrás, era habitual que una de las funciones a la cual los operadores del NOC/SOC (Network Operations Center/Security Operations Center) dedicaran gran parte de su tiempo, fuera la revisión de los logs generados por los sistemas bajo su administración. El crecimiento del número de activos, la heterogeniedad de los mismos, y la prevalencia cada vez mayor de los APT (Advanced Persistent Threat), hacen impresindible un cambio de metodología.

La existencia de diversas fuentes de información que permiten la trazabilidad de la información a través de los activos y la evolución de tecnologías de sensores de seguridad informática, orientan el desarrollo de un nuevo paradigma para enfrentar, en forma sistémica, la actual realidad.

El Ministerio de Defensa Nacional (MDN) se encuentra trabajando en el desarrollo de diferentes proyectos en materia de seguridad informática. SensorNET se enmarca dentro de los mismos.

II. Teoría

A. Visibilidad de activos computacionales.

El concepto de auditoría se encuentra ampliamente desarrollado en ámbitos financieros y contables. En el caso de los sistemas de información, es cada vez más frecuente encontrar características similares para auditoría en éstos.

Sin embargo, la existencia de la trazabilidad de la información no garantiza que la función de auditoria sea realizada. Generalmente los responsables de los NOC/SOC deben dedicar los escasos recursos humanos en temas más urgentes en pos de mantener la operativa en funcionamiento.

En organizaciones donde el nivel de madurez respecto a la seguridad informática lleva a la creación de un grupo especializado en materia de vigilancia de la misma, igualmente se encuentran con escollos importantes; como ser el volumen de información a analizar y la disparidad de formatos con que la misma es obtenida.

Queda claro que no basta con la existencia de la información en sí, sino que es imprescindible el desarrollo de sistemas auxiliares, especialmente creados para este fin, que se incorporen en la cadena de valor transformando la información en conocimiento.

Adicionalmente, la existencia cada vez mayor de reglamentaciones específicas en materia de responsabilidad en el manejo de la información, especialmente de información de terceros bajo custodia y/o operación, obligan a las organizaciones a disponer de elementos probatorios en el cumplimiento del celo de la misma, lo cual generalmente lleva la necesidad de realizar reportes ante la autoridad reguladora.

B. Sensores de Seguridad Informática.

En adición a las capacidades de auditoría ya indicadas para los sistemas de información, en el correr de los años se han desarrollado diferentes tecnologías especialmente orientadas a la seguridad informática. Es el caso de los NIDS/NIPS (Network Intrusion Detection System/Network Intrusion Prevention System) o sus equivalentes a nivel de “hosts” HIDS/HIPS.

Además, la mayoría de los elementos activos asociados a la seguridad perimetral como ser firewalls, generan trazas que reportan el pasaje de la información a través de ellos.

En el mismo sentido que los IDS/IPS, otras tecnologías como el caso de Honeypot, crean facilidades para conocer al “enemigo” que, mayoritariamente por motivaciones económicas o políticas, pretende hacerse de información que pueda efectar la posición o imagen de la organizacón afectada.

El modelo donde se soporta la gran mayoría de estos sensores es dispar, siendo muy baja la capacidad de consolidar la información generada por cada uno de ellos, elevando aún más la entropía de información para el analista de seguridad.

Esto lleva a la necesidad de una nueva herramienta, diferente a los sensores en si mismos, cuya función sea la de consolidar y uniformizar la información disponible, conduciendo al concepto de consola única de seguridad informática.

Si esta nueva herramienta sólo se orientará a la función de consolidación y uniformización de las fuentes de información, si bien permitiría estar un escalón por encima de la situación anterior, igualmente presenta el problema de dejar para los operadores humanos el análisis de toda esta

Artículo presentado como parte del del Proyecto de Grado – SensorNET.
Tutores del proyecto: Ing. Eduardo Cota e Ing. Alejandro Blanco.
Instituto de Ingeniería Eléctrica – Facultad de Ingeniería – Universidad de la República.
Año 2012.

información colectada. En este nivel, la herramienta sería un "Log Management".

El salto cualitativo ocurre cuando la herramienta, a partir de la información consolidada, es capaz de aplicar "inteligencia" a fin de procesar el gran volumen de información disponible, generando una visión global e independiente del sensor que la genera. Esta consola única para la seguridad informática se denomina SIEM.

C. SEM. SIM y SIEM.

Los acrónimos SEM (Security Event Management), SIM (Security Information Management), y SIEM (Security Information and Event Management) han sido utilizados en forma dispar por varios productores de herramientas orientadas a la gestión avanzada de la información recolectada por sensores de seguridad informática.

El término SIEM, fue acuñando por Mark Nicollet y Amrit Williams de Gartner en 2005[1], describiéndolo como aquel con capacidades de obtener, analizar y presentar información desde dispositivos de red y seguridad, aplicaciones de gestión de identidad y acceso, gestor de vulnerabilidades y herramientas para el cumplimiento de políticas, sistemas operativos, base de datos y logs de aplicaciones, e información de amenzas externas.

El foco clave es el monitoreo y la asistencia en la gestión de los privilegios de usuarios y servicios, cambios de configuración de sistemas, así como también proveer revisión de los logs de auditoría y respuesta ante incidentes.

Desde esta óptica, las capacidades que debe poseer un SIEM son:

- Agregación de datos.
- Correlación.
- Alerta.
- Visualización (Dashboards).
- Cumplimiento.
- Retención.

La alineación completa o parcial de los productos existentes en el mercado, tanto comerciales como Open Source Software (OSS), variará en cada uno y puede generar la inclusión o no de un producto específico como SIEM, en función de que se considere indispensable alguna de las capacidades no disponibles, según el autor del análisis.

En nuestro caso, modelaremos un SIEM genérico a fin de detallar las funciones de cada uno de sus bloques. Posteriormente, efectuaremos una breve presentación de alguna de las herramientas disponibles hoy en el mercado. El análisis se basa en el presentado en [2]. Un diagrama del modelo planteado se muestra en la Fig. 1.

III. Módelo genérico de un SIEM

A. Dispositivo fuente.

Denominamos dispositivos fuente a todo aquel dispositivo que pueda suministrar información de auditoría de seguridad informática, sea el suministro o no de dicha información su principal función.

En esta categoría encontramos los sistemas operativos, appliances especializados (ej. firewalls, IDS/IPS, etc.), software antimalware o honey* (honeypot y/o honeynet).

Como ya indicáramos, la problematica asociada a los dispositivos fuente es la diversidad de su tipo y los protocolos de comunicación soportados. Los SIEMs tienden a resolver esta situación desarrollando conectores o plugins especializados para cada uno de los dispositivos de interés.

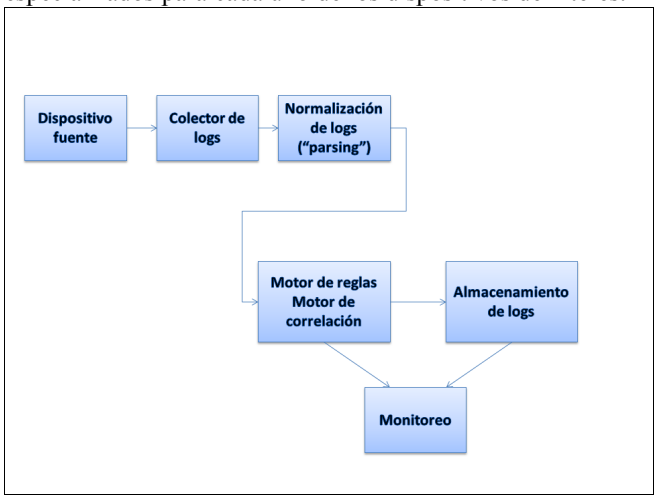


Fig. 1 – Modelo de SIEM genérico. [2]

Un punto no menor respecto a los protocolos de comunicación, es la posibilidad o no de brindar algún soporte de confidencialidad. Si consideramos que la gran mayoría de los dispositivos fuente utilizan protocolos como Syslog o SNMP, es muy habitual que los mismos no brinden soporte a este requerimiento. Inclusive, aunque se trate de dispositivos orientados a la seguridad perimétrica o como sensores de seguridad informática, no hay garantía de que los protocolos de reporte utilicen algún mecanismo de cifrado.

Si bien los dispositivos fuente no son parte en sí mismos de un SIEM, son una pieza de vital importancia en todo el proceso. Son ellos los encargados de alimentar de información al SIEM por lo que en el caso de no contar con ellos, el SIEM carecería de utilidad.

Algunas veces, los dispositivos fuente pueden poseer información que no está disponible para ser enviada hacia el exterior. En estos casos, son de utilidad aquellas piezas de software agrupadas genéricamente bajo el nombre de agentes, los cuales tienen como función recolectar la información contenida dentro del dispositivo y, previo procesamiento local o no, enviarla al SIEM.

B. Colector de logs.

Este bloque en nuestro SIEM genérico, es el responsable de recibir todos los logs generados por los dispositivos fuente, cada uno en su formato nativo. El mecanismo por el cual el SIEM obtiene dichos logs, varía en función del SIEM que se esté utilizando, pero en su forma más básica, puede ser diferenciado en dos métodos fundamentales: el dispositivo fuente envía sus logs al SIEM ("push"), o el SIEM recupera los logs del dispositivo ("pull"). Cada unos de estos métodos tienen sus pros y contras, dependiendo del entorno, pero son efectivos en el hecho de que el SIEM obtenga la información de los dispositivos fuente.

C. Normalización de logs.

En este punto, los logs de los diferentes dispositivos fuente se encuentran aún en su formato nativo por lo que sólo hemos desarrollado, hasta el momento, un repositorio

único para ellos (Log Management). A fin de que los mismos puedan ser utilizados por el SIEM es necesario procesar los mismos y convertirlos a un formato único. El acto de modificar los distintos formatos a uno sólo se denomina normalización. Para cada tipo de SIEM este formato único seguramente diferirá, dado que se espera que la consistencia del mismo sea sólo necesaria entre productos del mismo fabricante o proyecto de desarrollo. Este es otro punto a considerar en el caso de desear la interacción entre el SIEM de la organización y otros interesados, por ejemplo un CERT (Computer Emergency Response Team) nacional. Aquí también deberían considerarse temas de confidencialidad o procesos de "sanitización" de la información compartida con terceros.

En general, el formato de los logs dependerá del equipo que los genera. A fin de lograr que sean convertidos en el formato único soportado por el SIEM, es necesario poseer un conocimiento detallado de los mismos de manera de lograr su conversión. En general, los SIEM proporcionan varios "conversores" para los dispositivos fuente más habituales, aunque es necesario recordar que existen situaciones en las cuales el propio desarrollador o administrador del SIEM deberá programar esta "conversión". Nuevamente, tanto el número de "conversores" disponibles, como la capacidad de programar nuevos, es un diferenciador entre distintos SIEM.

La normalización de los eventos no sólo hace más sencilla la lectura de los logs, también permiten generar reglas, como las de correlación, en una forma estándar.

D. Motor de reglas/Motor de correlación.

El motor de reglas se expande por encima de la normalización de eventos de los diversos dispositivos fuente, de manera que puedan dispararse alertas al SIEM debido a condiciones específicas dentro de los logs. El método de escribir reglas en el SIEM usualmente comienza en forma moderadamente simple, pero puede tornarse extremadamente complejo. Típicamente, son reglas en forma de lógica booleana, donde se examinan ciertos campos de información en búsqueda de patrones predefinidos. El hecho de que previamente a esta búsqueda se realice la normalización de los diferentes logs, permite que dicha búsqueda sea uniforme y fácilmente trasladable a todos los eventos que son procesados por el SIEM.

Ahora bien, este motor de correlación no es simplemente un conjunto de reglas booleanas. Si fuera tan simple como eso, el desarrollo de dichos motores de correlación, su flexibilidad, efectividad y eficiencia, no serían propiedades que diferencian unos de otros, sin importar si los mismos se basan en soluciones comerciales u OSS. A la hora de realizar la acción de correlación, además de las expresiones booleanas, es necesario darle contextualización a las mismas. En este sentido, lo que se busca es que se logre una comprensión más acabada de cómo se originan los eventos, cuál dispositivo, sistema operativo o aplicación los origina, la ubicación dentro de la infraestructura informática, y la existencia o no de otros dispositivos fuente que pudieran estar relacionados.

La idea detrás de un motor de correlación es permitir que la inteligencia humana sea llamada a trabajar sólo después de que un proceso consolide, de manera inteligente, la mayor cantidad posible de los eventos recibidos; el especialista sólo deberá actuar cuando la discrecionalidad de la inteligencia del algoritmo no sea suficiente para poder determinar si se está o no frente a un incidente.

E. Almacenamiento de logs.

Para poder trabajar con grandes volúmenes de información, los cuales convencionalmente son la realidad involucrada con un SIEM, es necesario desarrollar una estrategia de retención ya sea por propósitos históricos, o por mandato de cumplimiento reglamentario. La estrategia de retención se puede basar en tres paradigmas: una base de datos, archivos de texto plano o archivos binarios.

La utilización de una base de datos, es el método más habitual para el almacenamiento de logs en las soluciones SIEM. Los motores de base de datos habitualmente utilizados son soluciones de escala "enterprise". La primera ventaja en la utilización de una base de datos es que existe un método para, en forma estructurada, almacenar la información, en nuestro caso, los logs o eventos recibidos y procesados por el SIEM. Como segunda ventaja, se observa el contar con un entorno para la manipulación de dicha información a través de un lenguaje como el SQL. Normalmente el acceso a la información en este formato es bueno, aunque a veces se requiera de la optimización del mismo a fin de ser utilizada en forma más adecuada por el SIEM.

Respecto a los archivos en texto plano, casi en su totalidad los diferentes Syslogs Servers los utilizan para la retención de sus eventos. La primera ventaja en este esquema es que no se requiere una aplicación adicional para la gestión de los archivos. En el caso de base de datos se requiere una aplicación especializada, que es el motor de la base de datos. Otra ventaja es el hecho de que los archivos son legibles por un ser humano, por lo cual también pueden ser inspeccionados manualmente o con herramientas diferentes al propio SIEM. Como contrapartida, la primera desventaja es el escalamiento. A menos que se tomen los recaudos correspondientes, se tiende a significativo aumento del requerimiento del espacio de almacenamiento con respecto a una solución que utilice una base de datos. También en este aspecto, la cantidad de archivos puede tornarse muy grande, o el tamaño del archivo puede agregar dificultad en su manejo. En este caso nos encontraremos con las limitantes propias del sistema de archivos que use el sistema operativo que contiene los registros, o que la información no se encuentre con alguna estructura jerárquica, lo cual conspira también contra la velocidad de procesamiento en el caso de búsqueda de alguna información particular.

Finalmente, los archivos en formato binario son utilizados en algunas ocasiones como un formato propietario, siendo sólo accesibles por el SIEM para el cual han sido diseñados. En general, el SIEM es la única aplicación que tiene la capacidad de "comprender" este formato propietario.

F. Monitoreo.

El bloque final de nuestro SIEM es el que interactúa con los logs almacenados por el SIEM. Una vez que todos los logs en el SIEM y los eventos han sido procesados, es

necesario el uso de la información útil, de lo contrario los logs únicamente existirían en el SIEM con el propósito de almacenamiento. El SIEM tendrá alguna consola a fin de proporcionar una forma de interacción con el mismo, la cual generalmente es alguna aplicación Web, aunque también podría ser algún tipo de cliente particular que pueda ejecutarse en una estación de trabajo, para alguna modalidad cliente/servidor. Sea cual fuere la interfaz utilizada, la misma permitirá la interacción con la información almacenada por el SIEM y con los resultados del motor de correlación. Dependiendo de la implementación en particular, la misma consola puede también permitir la administración del propio SIEM.

Otra de la funciones de la consola del SIEM es el brindar un entorno que permita la generación de nuevas configuraciones para el motor de reglas o el motor de correlación. A su vez se permitirá ajustar los métodos por los cuales se extrae información desde los dispositivos fuente, principalmente en el caso de que dicha información se obtenga mediante algún proceso de "pooling" programado. También ofrecerá trabajar con la información almacenada por el SIEM, permitiendo la generación de reportes o la exportación de la misma con fines forenses o para ser compartida con otro centro de seguridad.

IV. Implementaciones

A. Selección de productos a estudiar.

Una vez presentadas las principales características de un SIEM genérico, pasaremos a comentar el estudio de alguna de sus implementaciones. Si bien la taxonomía previamente presentada permite comprender y establecer una base de comparación entre diferentes productos, claramente no todos se ajustarán en forma estricta a la misma.

En esta sección nos dedicaremos a una breve presentación de cuatro SIEM seleccionados, mediante los cuales es posible el despliegue de una red distribuida de sensores.

Para la elección de los mismos, se buscó que dos fueran del tipo OSS y dos del tipo "Comercial". Estas categorías no son excluyentes, pues como es habitual en varios proyectos OSS, los SIEM OSS seleccionados también poseen una versión comercial. En el caso que aplique, se presentará el producto OSS y se describirá la diferencia con su equivalente "Comercial".

Dado que el número de SIEM en el mercado es bastante amplio, se tomó como criterio de elección estudios realizados por Gartner conocidos como "Magic Quadrant", en este caso para SIEM[3]. Este estudio es factor especialmente utilizado para seleccionar las soluciones "Comercial".

En el caso de las soluciones OSS, se estudió el tiempo de trabajo de la misma, su nivel de desarrollo, su integración con otras soluciones, y si, en el presente, el proyecto continúa en actividad.

En base a los criterios expresados, las herramientas que elegimos para estudiar son:

- HP ArcSight ESM (Comercial)
- Q1 Labs QRadar (Comercial)
- AlienVault OSSIM (OSS y Comercial)
- Prelude-IDS (OSS y Comercial)

B. HP ArcSight ESM

ArcSight cuenta con varios productos de seguridad adaptables a una amplia gama de empresas de diferente tamaño. Según la información suministrada por su sitio Web, hoy en día su producto SIEM es utilizado tanto en organizaciones corporativas como en ambientes educativos y gubernamentales; ya sea para detectar amenazas a la seguridad de sus redes como para cumplir con normativas de seguridad.

En las Fig. 2 y Fig. 3 se presentan los diferentes productos ArcSight, indicando en cada uno la capa de la arquitectura ArcSight en que trabajan y como interactúan entre si.

Fig. 2 – Productos modulares de HP ArcSight.[4]

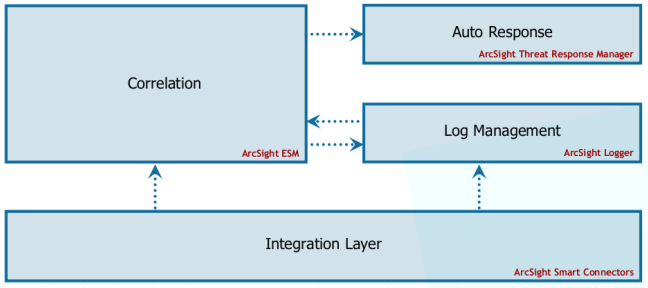


Fig. 3 – Comunicación e interacción entre componentes de la arquitectura ArcSight.[4]

En general los productos ArcSight cuentan con paquetes opcionales que habilitan a sus equipos para cumplir con reglamentaciones para el manejo de información como ser: SOX (utilizado en entornos bancarios y financieros), PCI DSS (utilizado por agentes que participan en la cadena de manipulación de información asociada a tarjetas de crédito), HIPAA (utilizado para la gestión de registros médicos) entre otros.

Estos dispositivos están disponibles para ser implantados en dos opciones; como hardware independiente o como software para ser instalado en un equipo ya existente.[2]

Dado que ArcSight utiliza terminología específica para sus productos, y a fin de conocerlos, haremos una breve descripción de los mismos:

- **ArcSight ESM Manager:** Actúa como el procesador central del ArcSight ESM. Se encarga de descifrar y descomprimir los eventos que llegan desde el Smart Connector, procesa los logs según las reglas predefinidas, los almacena en la base de datos y realiza las consultas históricas sobre la misma.
- **ArcSight ESM Database:** Se trata de la base de datos Oracle que es utilizada para el

almacenamiento de los eventos recibidos desde otros dispositivos, la configuración del sistema y el conjunto de reglas.

- **ArcSight Partition Archiver:** Es el encargado de gestionar y actualizar la base de datos.
- **ArcSight ESM Console:** Se trata de una consola Java que es la interfaz del ArcSight ESM. Puede ser utilizada por el personal de seguridad dedicado a la gestión de los incidentes.
- **ArcSight SmartConnector:** Es una pieza de software (o hardware) encargada de recolectar en sus formatos nativos los logs provenientes de los dispositivos fuente. Sus funciones incluyen la normalización de los logs a un formato común (CEF, Common Event Format) en la cual el ArcSight ESM trabajará, y el envío de los mismos hasta el ArcSight Manager de una forma segura.
- **ArcSight ESM Web:** Se trata de una interfaz web que actúa como una consola más "liviana", permitiendo un acceso rápido a la información específica de incidentes de seguridad.
- **Common Event Format (CEF):** Es el formato en el que ArcSight ESM analizará y almacenará los registros generados por los dispositivos fuente.
- **Evento Base:** Es un evento generado por un dispositivo fuente, por ejemplo un switch o un router.
- **Evento correlacionado:** Es aquel generado a partir de otros eventos y del cumplimiento de una o más reglas, o del cruce de algún umbral/norma preestablecida. Los eventos correlacionados son generados a partir del motor de correlación. Un ejemplo de esto es una sucesión de eventos ocurridos en un orden lógico específico.
- **Evento de correlación:** Es el producto de las condiciones que se reúnen dentro de un regla. Dicho evento es enviado nuevamente hacia el motor de correlación para ser comparado con otros almacenados, a fin de proveer un análisis automático adicional. [2]

C. Q1 Labs QRadar

Al igual que ArcSight los productos de Q1 Labs están disponibles tanto en hardware como en software. Su producto principal en seguridad se denomina QRadar SIEM. Este sistema incluye todo lo necesario para la recolección de información y la gestión de eventos de seguridad, incluyendo además el almacenamiento de logs, el monitoreo del tráfico, la generación de reportes

Comencemos por el appliance QRadar SIEM. Este sistema será quien reciba y almacene los reportes desde los dispositivos sensados de la red o de otros sistemas de seguridad, ya sea mediante eventos de Syslog o de Windows Event Log. Así mismo QRadar SIEM Appliance puede recibir flujos de datos desde otras aplicaciones como ser NetFlow de Cisco, J-flow de Juniper, Nessus o QualysGuard.

Al observar la descripción precedente, queda claro que este appliance constituye una solución de tipo "All-in-one", ya que puede recibir los eventos de los dispositivos fuente directamente, siendo todo el posterior procesamiento realizado en forma interna. Desde este punto de vista, este appliance abarca todos los componentes de nuestra solución genérica.

Además, este sistema cuenta con puertos de monitoreo para la captura del del tráfico sensado ("sniffing"), completando así un análisis del tráfico hasta la capa aplicación.

Una vez recibidos los datos desde los diferentes sensores, procederá a normalizarlos, clasificarlos, aplicarles las reglas de filtrado y matching correspondientes, almacenarlos y finalmente analizarlos y correlacionarlos.

En la Fig. 4 se muestra un esquema de cómo el sistema SIEM QRadar procesa los diferentes tipos de eventos así como aquellos módulos involucrados en dicho proceso.

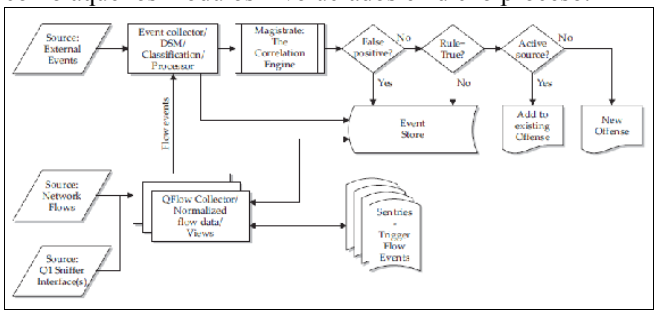


Fig. 4 – Arquitectura básica de QRadar.[2]

El proceso se inicia luego de que un evento o flujo ha sido recibido por la unidad QRadar, la cual filtrará este evento o flujo utilizando objetos conocidos como BB: Building Blocks. Los datos convertidos, denominados Views, serán pasados a los Sentries o Centinelas. Estos últimos generarán eventos internos que serán enviados al motor de procesamiento de reglas el cual se denomina Magistrate.

En esta descripción, el Event Collector y Qflow Collector, constituyen el Colector y Normalización de logs de nuestro modelo genérico, mientras que el Magistrate lo es con respecto al Motor de reglas/Motor de correlación.

Lo interesante de esta arquitectura es la existencia de un módulo especializado en la recepción de flujos, sea en la forma de NetFlow, sFlow u otros, o de capturas en modo "sniffer". Esto constituye una forma de incorporación de la detección en base a anomalías de tráfico, aceptando que su operación no es tan parametrizable como en el caso del procesamiento de eventos.

En este punto, el Magistrate utiliza un sistema de ponderación en base a varios atributos configurables para determinar si la situación es lo suficientemente hostil como para implicar una ataque. Un ataque disparará un aviso al profesional de seguridad para que analice la situación a fin de mitigar los daños y solucionar la amenaza. Estas alertas se pueden configurar para ser enviadas por correo electrónico, implicando claramente la intervención del especialista a fin de procesar el aviso generado.

Una vez remediada la amenaza es posible utilizar QRadar para elaborar informes detallados sobre las acciones específicas que provocaron las circunstancias hostiles y las medidas adoptadas para asegurar la situación.

En particular el modelo 2100 de QRadar es el indicado para organizaciones de pequeño y mediano porte, por

tratarse de una solución SIEM "All-in-one". Sin embargo este sistema puede ser ampliado con el módulo colector QFlow de la serie 1100, el cual proporciona la capacidad de monitorear flujos en segmentos distribuidos. Un ejemplo de esta solución se muestra esquemáticamente en la Fig. 5.

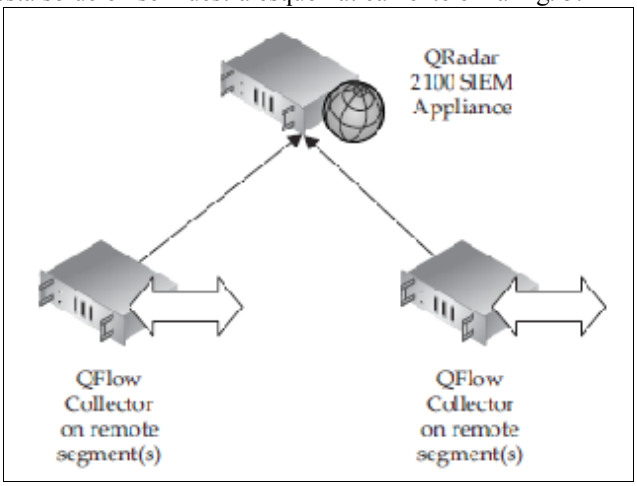


Fig. 5 – Solución Qradar 2100 SIEM con módulos Qflow Collector.[2]

Para organizaciones de mediano a gran porte el equipo recomendado es el QRadar 3100 Enterprise SIEM. Éste equipo puede ser expandido adicionándole módulos QFlow Collector de la serie 1100, módulos dedicados para el procesamiento de eventos como ser el Event Processor de la serie 1600, o módulos Flow Processor de la serie 1700. Un ejemplo de esta solución se muestra en la Fig. 6.

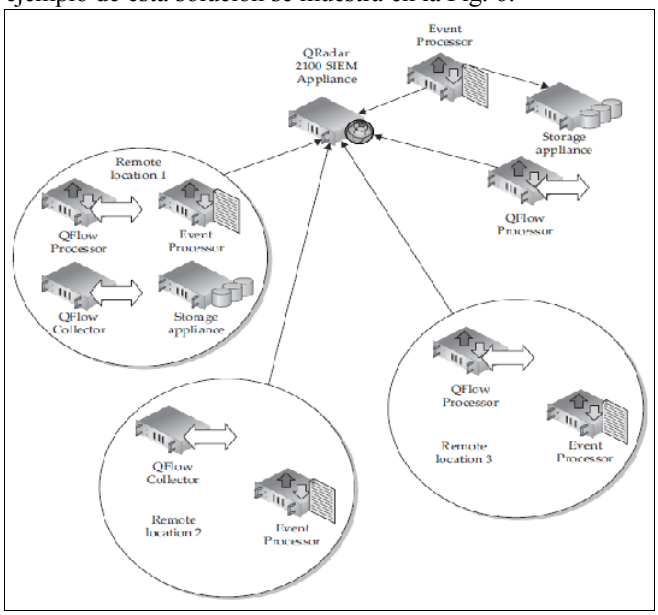


Fig. 6 – Solución Qradar 3100 SIEM con módulos de expansión.[2]

En esta descripción se puede observar como los componentes comienzan a operar en forma distribuida. Tal vez la diferencia con respecto a otras soluciones es que en el caso de QRadar, el formato appliance es el elegido para la distribución de la solución no considerándose la opción de componente de software a instalar. Si bien esto asegura un óptimo acoplamiento de los componentes, limita la libertad de elección y tiene la tendencia a que se pueda construir formas extremadamente propietarias de operación, limitando así flexibilidad de agregar componentes de terceros. Solo quedaría la integración a través de los Event Processor.

Los sistemas suplementarios de procesamiento mencionados, se implementan para aumentar la capacidad del sistema, lo que permite un aumento en el número de eventos. Éstos pueden ser implantados en cualquier punto de la red, logrando además reducir el tráfico hacia el gestor central debido a la normalización de los eventos y flujos recibidos. Cada dispositivo de las serie 1600 o 1700, ofrece 2 TB de almacenamiento adicional, e incluso admiten ampliar dicha capacidad mediante dispositivos externos. Otra opción es el utilizar los sistemas de almacenamiento NAS (Network Attached Storage) existentes en la organización (iSCSI por ejemplo).

Dado que los sistemas QRadar son compatibles con los anteriores, esto puede ser una manera útil de aumentar la capacidad de almacenamiento de este SIEM.[2]

D. AlienVault OSSIM

OSSIM es una herramienta OSS disponible para ser descargada, instalada, modificada y operada. Como es de esperar, al existir una versión comercial, la versión gratuita tiene limitantes de performance, soporte y capacidad de almacenamiento. De hecho, debido a éste último aspecto, la propia AlienVault no considera a OSSIM como un SIEM completo. Sin embargo la versión OSS brinda un buen acercamiento al producto para aquellos interesados en instalar un SIEM en su organización.

El concepto base detrás de OSSIM es "no re-inventar la rueda". Para esto OSSIM utiliza varias herramientas probadas y ampliamente difundidas de código abierto, de forma de aprovechar en cada una sus aspectos más destacados. A este conjunto AlienVault le suma por su parte un motor de correlación, elaboración de informes y herramientas para la gestión centralizada.

Podemos destacar en general la flexibilidad de configuración brindada a los usuarios para adaptar esta herramienta a sus necesidades específicas.[2]

	Open Source	Professional SIEM
Support	Community	7x24
Quality Assurance	Community	Professional Q&A
Security	Not audited	Audited
Performance	Moderate	30 x Open Source, Assured
SIEM Intelligence	Logical Correlation Simple Taxonomy	Cross Correlation Rich Taxonomy
Logger	N/A	Unlimited Forensic Storage
Reports	< 25 + Jasper	> 200 + Web Wizard
Scalability/HA	N/A	HA, Distributed ,Multitenant, Unlimited
Compliance	High Level Reports	High and Low Taxonomy-based
Updates	None	Daily rules and reports
User Management	Individual, simple controls	Templates and Granular Controls

Fig. 7 – Comparativa de versión OSS y Comercial de AlienVault.

Si bien la versión libre puede ser utilizada durante el tiempo que se desee permitiendo en todo momento migrar a la versión profesional, existen algunas diferencias de base entre los dos productos. La Fig. 7 muestra un resumen de dichas diferencias.

Básicamente las dos versiones proporcionan las funciones generales como ser: detectar amenazas dentro de la red, recoger y almacenar eventos y otras informaciones (ej inventario) de la red, correlacionar la información antes mencionada, generar informes y administrar la infraestructura de seguridad de la red.

Entre las diferencias mencionadas en la Fig. 7 destacamos:

- **Seguridad**: Mientras que la versión Professional permite establecer comunicaciones cifradas entre sus componentes distribuidos así como con la mayoría de los dispositivos fuente; OSSIM no es compatible con tales comunicaciones.
- **Performance**: Cada componente de la versión Professional ofrece treinta veces mas rendimiento que la versión libre. En particular, cada sensor (colector) de la primera permite gestionar más de 5000 eventos por segundo; mientras que la performance de OSSIM no esta asegurada.
- **Correlación Cruzada**: AlienVault Professional puede llevar a cabo tanto una correlación lógica como una correlación cruzada, integrando información de inventario con la de IDS, aumentando así las capacidades de detección de ataques. OSSIM no ofrece la capacidad de correlación cruzada.
- **Logger**: Disponible sólo para la versión comercial, esta compuesto por una familia de dispositivos dedicados a proporcionar un transporte seguro y autenticado así como capacidad de almacenamiento para la información y los eventos. De esta forma los eventos pueden cifrarse, asegurando su integridad de extremo a extremo, y manteniendo su valor forense y legal.
- **Entornos distribuidos**: AlienVault Professional permite desarrollar una solución totalmente distribuida, permitiendo inclusive el desarrollo de soluciones de seguridad para un MISP (Management ISP). OSSIM no tiene dicho soporte, permitiendo sólo algunos elementos y con escasa documentación.

E. Prelude-IDS

Al igual que otras herramientas OSS, Prelude-IDS cuenta con la contribución de múltiples profesionales en seguridad alrededor del mundo. Sumando lo anterior a sus capacidades para colectar, normalizar, clasificar y correlacionar eventos de seguridad, independientemente de la marca y licencias de los dispositivos, convierte a Prelude-IDS en un sistema SIEM universal. Esta capacidad se basa en parte en algunas definiciones de arquitectura. Es claro que uno de los puntos claves en cualquier arquitectura de intercambio de información, es la elección/definición de la mensajería utilizada.

En el caso de las herramientas comerciales, el intercambio en forma "abierta" se da en la primera etapa, en lo que se refiere a la recepción de mensajes desde los dispositivos fuentes. Sin embargo, los intercambios subsecuentes, los cuales se realizan entre los componentes de la propia solución, se efectúan mediante protocolos propietarios para cada una de las mismas.

Esto no debe confundirse con la disposición de elementos "custom" que permiten desarrollar interpretadores para mensajería no soportada en forma nativa. Lo que aquí se indica es que no existe una API que permita realizar la comunicación directamente con los componentes de la solución.

En el caso de Prelude-IDS, se siguió un camino diferente. Si bien existe la posibilidad de desarrollar interpretadores para el procesamiento de fuentes no soportada en forma nativa, el carácter "abierto" al tratarse de un proyecto OSS, también impactó en la opción de la mensajería utilizada.

Se hizo la opción por un formato de mensajería abierta como es el caso del IDMEF[5]. Este formato de mensajería es el resultado del trabajo desarrollado por el grupo de IDWG (Intrusion Detection Working Group) de la IETF (Internet Engineering Task Force). Esto no sólo implica que la mensajería tiene un formato no propietario, sino que se permite su integración en forma más profunda. En particular, el proyecto Prelude-IDS desarrolló una biblioteca, libprelude, la cual constituye una API que permite la comunicación directa a los componentes de la arquitectura.

La biblioteca libprelude, no sólo resuelve la mensajería a utilizar, sino que realiza una implementación de transporte sobre SSL que garantiza la confidencialidad de la misma y los controles de acceso adecuados para los componentes. En este sentido, Prelude-IDS entrega una opción adicional que no se encuentra en las otras soluciones estudiadas.

El impacto de esta elección no es sólo positivo desde el punto de vista conceptual, sino también desde un punto de vista práctico. El proyecto, en sus más de diez años de vida, ha logrado la incorporación de su biblioteca en proyectos OSS como Snort-IDS, Nepenthes, etc.

Otro aspecto que también tiene referencia en la elección de una mensajería abierta, es la posibilidad de lograr la integración en forma estándar de diferentes soluciones. En particular, la adopción por parte del eCSIRT (European CSIRT Network – http://www.ecsirt.net), de la mensajería IDMEF, brinda un apoyo político a la utilización de esta mensajería.

F. Selección ideal.

Una vez estudiadas las cuatro herramientas SIEM seleccionadas (estudio que escapa al presente artículo), consideramos que la elección por parte de Gartner de HP ArcSight SIEM, tiene razones de peso. Es claramente ésta la herramienta que mayor versatilidad presenta, siendo posible su implantación en entornos diversos. La amplia variedad de sensores soportados, la posibilidad de modularización completa de la solución a fin de ajustarse a cada caso particular, su madurez, escalabilidad, así como la posibilidad de expansión "scale out", la hacen un excelente ecosistema de productos a elegir.

Ahora bien, dado que al presente se desconoce sobre los planes del Ministerio de Defensa Nacional (MDN), cliente motivador del proyecto; de la existencia de presupuesto comprometido para la adquisición de una solución de estas características; que la posibilidad de instalar una versión "demo" queda condicionada a la capacidad de EPS necesarios, los cuales no son conocidos por el cliente como especificación de requerimiento; y, adicionalmente, que la documentación pública de HP ArcSight no es del detalle esperable a la hora de realizar una implantación de este tipo, es que optamos por no implementar el piloto con este SIEM.

Como contrapartida, vemos que la documentación en aquellos proyectos OSS es más amplia, profunda y variada. Tal es el caso de Prelude-IDS, la cual es notablemente más

detallada, con varios y surtidos ejemplos originados por el soporte oficial del proyecto así como el aporte de la comunidad OSS.

Sin embargo, dado que los sensores soportados por HP ArcSight son suficientemente amplios, no hay impedimento alguno para una migración futura a esta herramienta, en caso de que el cliente lo considere. Esta línea de pensamiento, entre otras, es la que nos conduce a la elección de Prelude-IDS, como indicaremos más adelante al momento de la definición de la solución teórica y la implantación del piloto.

G. Selección para prueba de concepto.

Ya anticipamos en los párrafos anteriores, nuestra elección de Prelude-IDS para el proyecto motivador del presente artículo. Pasaremos ahora, a explicar las razones de dicha elección.

Los puntos que se consideraron fueron:

- **Razones económicas**. Al tratarse de una solución OSS, no se posee costo de adquisición. El único CapEx involucrado, es el referente a la adquisición del hardware para los diferentes componentes, así como las horas hombre para su implantación.
- **Documentación**. El proyecto posee una excelente Wiki donde se explican las funcionalidades de cada componente. También incluye información de como integrarlo a terceras partes, fundamentalmente los sensores existentes. Como también indicáramos, existe una amplia utilización del proyecto durante varios años, lo que genera aporte comunitario a la misma de relevancia, contándose con diversas fuentes de información para su implementación, ajustes y configuraciones, entre otras.
- **Madurez del proyecto**. Si nos remitimos a los orígenes del proyecto, el mismo posee más de diez años. Adicionalmente, es uno de los proyectos OSS en materia de seguridad más difundidos y aceptados, y que cuenta con una activa comunidad de usuarios.
- **Comunicación de los componentes**. El soporte de comunicación de Prelude-IDS, es una biblioteca que permite la utilización de SSL para el cifrado en el intercambio de información entre los componentes con el SIEM. Esta biblioteca ha sido integrada por los proyectos de desarrollo de sensores más populares y diversos, como en el caso de Snort que incorporó la misma como uno de sus componentes.
- **Integración con los componentes presentes en la red del MDN**. Considerando los diversos activos/componentes de la red del MDN, la implantación de Prelude-IDS no presenta desafíos insalvables. Según la información recabada durante el desarrollo del proyecto, los activos informáticos presentes a la fecha poseen recursos para la trazabilidad de eventos, los cuales son soportados por Prelude-IDS y permiten su integración a la solución.
- **Capacidad de migración**. Ya indicamos que si bien la elección para el proyecto es Prelude-IDS, consideramos que HP ArcSight es la mejor opción. Dado que la forma de implantación modular que soporta Prelude-IDS es similar con el despliegue de HP ArcSight, entonces entendemos que es posible migrar una red de sensores basada en un SIEM a otro. En el caso de que el MDN considere dicha posibilidad, la elección de Prelude-IDS no es una barrera para la misma.

V. Aplicación

A. Prueba de concepto, MDN.

A fin de establecer una prueba de concepto de la red de sensores de seguridad informática para la red del MDN, se determinó la instalación de un piloto correspondiendo a una versión reducida de una propuesta más general.

El piloto se implementó mediante la instalación de los siguientes componentes:

- Un SIEM con los componentes Prelude-IDS:
  - Prelude-Manager.
  - Prelude-LML.
  - Prelude-Correlator.
  - Prewikka.
- Un sistema de detección de intrusos de red (NIDS), utilizando el producto Snort IDS, con los componentes necesarios a fin de integrarse a la arquitectura Prelude-IDS.
- Un Honeypot corriendo Nepenthes con los componentes necesarios a fin de integrarse a la arquitectura Prelude-IDS.

Adicionalmente, los servidores poseen componentes como una base de datos MySQL, un servidor Web Apache, etc.

Dado el Acuerdo de Confidencialidad bajo el cual se desarrolló el piloto, sólo se presentará un esquema genérico para una red y el piloto desplegado. El planteamiento se resume en la Fig. 8 .

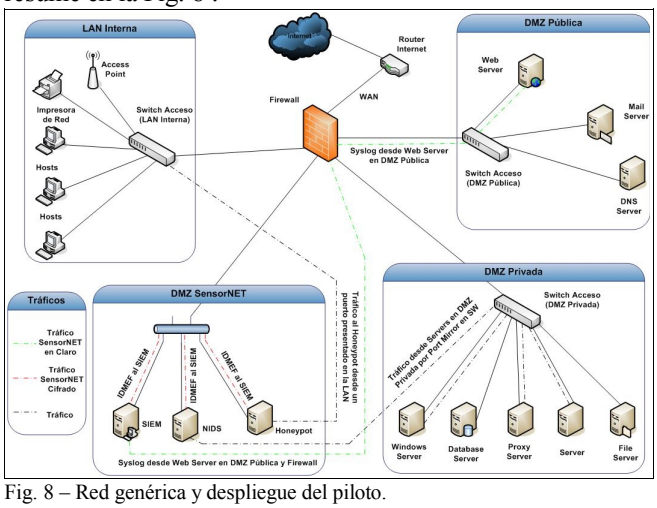


Fig. 8 – Red genérica y despliegue del piloto.

El piloto se encontró operativo con todos sus componentes, desde el 29 de diciembre de 2011, hasta el 30 de enero de 2012.

Durante el mes transcurrido, se procedió a realizar visitas periódicas a fin de verificar la correcta operación del mismo.

Desde antes de la finalización de la instalación de todos los componentes, fue posible observar desde la consola,

eventos desde los sensores ya disponibles. A fin de verificar la correcta operación de los sensores, se efectuaron algunas pruebas que permitieron no sólo verificar la generación y recepción de los mensajes IDMEF, sino que también se verificó la operación de correlación a partir del Prelude-Correlator.

Desde este punto de vista, el resultado fue más que satisfactorio, verificando que la solución es totalmente válida y es posible el despliegue de una red en producción basada en la herramienta elegida.

VI. Conclusión

Hemos visto diferentes opciones de SIEM y realizado un estudio de las mismas para la aplicación en la red objetivo del proyecto motivador.

Dicho estudio nos ha llevado a elegir Prelude-IDS como el SIEM sobre el cual desarrollamos tanto nuestra solución teórica como el piloto.

Sin embargo, indicamos que HP ArcSight es un excelente producto que debería ser considerado en el caso de disponer capacidad presupuestaria para su adquisición.

Referencias

[1] *The Future of SIEM - The market will begin to diverges* http://techbuddha.wordpress.com/2007/01/01/the-future-of-siem-%E2%80%93-the-market-will-begin-to-diverge/

[2] David Miller, Shon Harris, Allen Harper, y Stephen VanDyke, *Security Information and Information Management (SIEM) Implementation*, McGraw-Hill – Network Pro Library 2011

[3] Gartner, *Magic Qudrant for SIEM*, May 12, 2011 http://www.arcsight.com/collateral/whitepapers/Gartner_Magic_Quadrant_2011.pdf

[4] Presentación de ArcSight en el Technology Day Genève, 17 Mars 2010, http://www.eb-qual.ch/cms/media/5-ArcSight_Labbe.pdf

[5] *Intrusion Detection Message Exchange Format (IDMEF),* IETF RFC 4765, March 2007, http://www.ietf.org/rfc/rfc4765.txt

Emiliano Colina Nació en Montevideo, Uruguay, el 23 de octubre de 1977. Actualmente cursa décimo semestre de Ingeniería Eléctrica opción Telecomunicaciones, Facultad de Ingeniería de la Universidad de la República.

Anselmo Hermida Nació en Montevideo, Uruguay, el 21 de mayo de 1976. Actualmente cursa décimo semestre de Ingeniería Eléctrica opción Telecomunicaciones, Facultad de Ingeniería de la Universidad de la República.

Posee el título de Ingeniero Tecnológico en Electrónica expedido por la Universidad del Trabajo del Uruguay (U.T.U.) desde el año 1999.

Waldemar Pera Nació en Montevideo, Uruguay, el 14 de noviembre de 1973. Actualmente cursa décimo semestre de Ingeniería Eléctrica opción Telecomunicaciones, Facultad de Ingeniería de la Universidad de la República.

Posee el título de Ingeniero Tecnológico en Electrónica expedido por la Universidad del Trabajo del Uruguay (U.T.U.) desde el año 1995.