

OSSIM

Open Source Security Information Management

Descripción General del Sistema

Equipo

El equipo desarrollador del proyecto en la actualidad es:

Dominique Karg, <dk@ossim.net>

Responsable Técnico

Jesús D. Muñoz, <jesusd@ossim.net>

Responsable de Documentación

David Gil, <dgil@ossim.net>

Desarrollo

Santiago González, <sgonzalez@ossim.net>

Integración de Software

Julio Casal, <jcasal@ossim.net>

Coordinador

Visión General del Proyecto

OSSIM quiere suplir un hueco en las necesidades que un grupo profesionales del mundo de la seguridad día a día nos encontramos.

Nos sorprende que con el fuerte desarrollo tecnológico producido en los últimos años que nos ha provisto de herramientas con capacidades como las de los IDS, sea tan complejo desde el punto de vista de seguridad de obtener una visión de una red con un grado de abstracción que permita una revisión práctica y asumible.

Nuestra intención inicial en el desarrollo de este proyecto es mejorar esta situación a través de una función que podríamos resumir con el nombre de:

CORRELACIÓN

O la posibilidad de obtener una visibilidad de todos los eventos de los sistemas en un punto y con un mismo formato, y a través de esta situación privilegiada relacionar y procesar la información permitiendo aumentar la capacidad de detección, priorizar los eventos según el contexto en que se producen, y monitorizar el estado de seguridad de nuestra red.

La idea de correlación está también implícita en la visión de nuestro proyecto en el sentido de agregación e integración de productos: queremos incluir un número de magníficos productos desarrollados en estos años en un Framework general, que permitirá nuevas posibilidades al interrelacionar todas sus funcionalidades.

En el camino nos hemos encontrado con nuevas necesidades que nos han permitido aumentar la precisión de nuestro sistema, desarrollando la capacidad que ya forma parte del núcleo de OSSIM:

VALORACIÓN DE RIESGOS

Como forma de decidir en cada caso la necesidad de ejecutar una acción a través de la valoración de la amenaza que representa un evento frente a un activo, teniendo en cuenta la fiabilidad y probabilidad de ocurrencia de este evento.

Desde este momento nuestro sistema se vuelve más complejo pues ha de ser capaz de implementar una *Política de Seguridad*, el *Inventario de la Red*, nos ofrecerá un *Monitor de Riesgos* en tiempo Real, todo ello configurado y gestionado desde un *Framework*... No debemos en cualquier caso dejar que esta complejidad nos aparte de nuestro objetivo que es la integración de productos.

El resultado es por lo tanto realmente ambicioso y hereda todas las funcionalidades y el gran esfuerzo de desarrollo de una comunidad de expertos siendo nuestro papel el de meros integradores y organizadores.

Este proyecto quiere ser así mismo una muestra de la capacidad del mundo de código abierto de crecer en sí mismo y aportar soluciones punteras en sectores concretos como el de la seguridad de redes, donde las soluciones del mundo libre aportan otro importante valor: la auditabilidad o capacidad de auditoría de los sistemas que instalamos en nuestra red.

Control Documental

| | |
|------------------------------|------------------------------------|
| Título: | OSSIM: Descripción General Sistema |
| Código de Referencia: | OSSIM0.17 |
| Versión: | 0.17 |
| Fecha Edición: | 21 de octubre de 2003 |
| Autor: | Julio Casal |

Control de Versiones

| VERSIÓN | PARTES QUE CAMBIAN | DESCRIPCIÓN DEL CAMBIO | FECHA DE CAMBIO |
|----------------|---------------------------|-------------------------------|------------------------|
| 0.15 | Inicia | - | 06/11/2002 |
| 0.16 | Correcciones | | 17/06/2003 |
| 0.17 | Corr. Generales | | 21/10/2003 |

INDICE DE CONTENIDOS

| | |
|---|-----------|
| Equipo..... | 2 |
| Visión General del Proyecto..... | 3 |
| 1. Introducción | 7 |
| 1.1. <i>Introducción.....</i> | 7 |
| 1.2. <i>¿Que es OSSIM?.....</i> | 8 |
| 1.2. <i>Infraestructura Open-Source de Monitorización de Seguridad.....</i> | 9 |
| 2. El Proceso de Detección..... | 10 |
| 3. Funcionalidad..... | 13 |
| 3.1 <i>Detectores de Patrones.....</i> | 13 |
| 3.2 <i>Detectores de Anomalías.....</i> | 14 |
| 3.3 <i>Centralización / Normalización.....</i> | 15 |
| 3.4 <i>Priorización.....</i> | 16 |
| 3.5 <i>Valoración del Riesgo.....</i> | 17 |
| 3.6 <i>Correlación.....</i> | 18 |
| Entrada y Salida..... | 18 |
| 3.6.1 <i>Modelo de Correlación.....</i> | 18 |
| 3.6.2 <i>Métodos de Correlación.....</i> | 19 |
| 3.6.2.1 <i>Método 1: Correlación mediante Algoritmos Heurísticos.....</i> | 20 |
| 3.6.2.2 <i>Método 2: Correlación mediante Secuencias de Eventos.....</i> | 22 |
| 3.6.3 <i>Niveles de Correlación.....</i> | 23 |
| 3.6.3 <i>Niveles de Correlación.....</i> | 23 |
| 3.7 <i>Monitores.....</i> | 26 |
| 3.8. <i>Consola Forense.....</i> | 28 |
| 3.9 <i>Cuadro de Mandos.....</i> | 29 |
| 4. Arquitectura..... | 30 |
| 4.1. <i>Arquitectura General.....</i> | 30 |
| 4.2 <i>Flujo de los datos.....</i> | 31 |
| 5. Proyecto y Contactos | 34 |

1. Introducción

1.1. Introducción

El desarrollo de este trabajo quiere dar respuesta a esta imagen que durante estos últimos años se ha repetido tantas veces:

“En el éxito de una intrusión, una vez franqueadas las defensas perimetrales se sucede el compromiso de decenas de máquinas. Existe un flujo de conexiones permanentes y duraderas durante varias horas, conexiones anómalas que dibujan un camino completamente contrario a lo que debería ser aceptable; procedentes desde el exterior crean un puente de entrada a la red interna donde una tras otra van comprometiendo más máquinas trazando un camino cada vez más anómalo, y peligroso..

Los usuarios y administradores de la organización víctima, que en ese momento se encuentran trabajando en las máquinas nunca notan nada extraño en el momento, y rara vez localizan el ataque a posteriori.”

La analogía con el mundo real, aunque algo exagerada y cómica sería la de un ladrón que entra dando una patada a la puerta de cualquier oficina, se pasea por los despachos y pasillos donde la gente trabaja pues es medio día, visita los archivos, fotocopia la documentación que le interesa, encuentra una caja fuerte y sin ningún tipo de complejos se pone a darle martillazos. Mientras los empleados siguen ausentes concentrados en su trabajo...

Algo falla en la detección de ataques de las redes corporativas, aparentemente tenemos la tecnología apropiada, a través sistemas de detección de intrusos somos capaces de detectar los eventos más concretos, sin embargo no somos capaces de revisar todas las alertas que estos nos envían debido a dos razones:

- **la cantidad**
- **la poca fiabilidad**

En otras palabras, obtenemos demasiadas alertas y estas no son fiables, obtenemos demasiados **falsos positivos**.

Obtenemos así mismo información muy detallada, pero atómica, parcial y sin capacidad de abstracción, no somos capaces de detectar ataques definidos por comportamientos más complejos, nuestro segundo problema son los **falsos negativos**.

1.2. ¿Que es OSSIM?

OSSIM es una distribución de productos open source integrados para construir una infraestructura de monitorización de seguridad.

Su objetivo es ofrecer un marco para centralizar, organizar y mejorar las capacidades de detección y visibilidad en la monitorización de eventos de seguridad de la organización.

Nuestro sistema constará de las siguientes **Herramientas de Monitorización**:

- a. *Cuadro de Mandos para visibilidad a alto nivel*
- b. *Monitores de Riesgo y Comportamiento para la monitorización a nivel medio*
- c. *Consola Forense y Monitores de Red para el bajo nivel*

Estas herramientas se alimentarán de las **nuevas capacidades** desarrolladas en el “postproceso” de los SIM y cuyo objeto es aumentar la fiabilidad y sensibilidad de la detección:

- a. *Correlación*
- b. *Priorización*
- c. *Valoración de Riesgos*

El postproceso a su vez es alimentado por los preprocesadores, estos son un número de detectores y monitores ya conocidos por la mayoría de administradores que integraremos en nuestra distribución:

- d. *IDS (detectores de patrones)*
- e. *Detectores de anomalías*
- f. *Firewalls*
- g. *Monitores varios*

Por último deberemos tener una herramienta de administración que configure y organice los diferentes módulos tanto externos como propios que integrará OSSIM, esta herramienta será el **Framework** y mediante ella podremos definir la Topología, inventariar activos, definir una Política de seguridad, definir las reglas de Correlación y enlazar las diferentes herramientas integradas.

1.2. Infraestructura Open-Source de Monitorización de Seguridad

Solución vs Producto

OSSIM no quiere ser un producto, sino una solución, un sistema personalizado para las necesidades de cada organización formado por la conexión e integración de varios módulos especialistas.

En nuestra solución tan importante como el código son los conceptos o definiciones de:

- a. *La Arquitectura*
- b. Los Modelos y Algoritmos de *Correlación*
- c. La definición del *Entorno y el Framework*
- d. La definición del Modelo de *Gestión de la Seguridad Perimetral*
- e. El Mapa y los Procedimientos de Auditoría de *la Capacidad de Detección*

Nuestro interés en este proyecto *gnu* es tanto ofrecer para su mejora el código como generar la discusión y el conocimiento de estos modelos y algoritmos.

Arquitectura Abierta

OSSIM es una arquitectura de monitorización abierta pues integra diversos productos del mundo libre, intentando seguir siempre los estándares y las tendencias del mundo open source (los cuales creemos que en soluciones de monitorización serán los estándares en todos los entornos).

Solución Integral

Es una solución integral pues es capaz de ofrecer las herramientas y funcionalidad para monitorización de todos los niveles desde el más bajo (firmas detalladas de un IDS, dirigido al técnico de seguridad), hasta el más alto (El Cuadro de Mandos dirigido a la Dirección Estratégica), pasando por: Consolas Forenses, niveles de Correlación, Inventariado de Activos y Amenazas, y Monitores de Riesgos.

Software de Código Libre

OSSIM se propone como un proyecto de integración, nuestra intención no es desarrollar nuevas capacidades sino aprovecharnos de la riqueza de "joyas" del software libre, programas desarrollados por la inspiración de los mejores especialistas del mundo (como pueden ser snort, rrd, nmap, nessus, o ntop...) integrándolas en una arquitectura abierta que heredará todo su valor y capacidades. Nuestro desarrollo será el encargado de integrar e interrelacionar la información de estos productos.

Estas herramientas de código libre son, por la naturaleza de este, probadas y mejoradas por decenas o centenas de miles de instalaciones en el mundo convirtiéndose en elementos robustos y altamente probados y por tanto *fiables*.

Por el hecho de ser código abierto son así mismo *confiables* y exentas de cualquier duda de posibles puertas traseras al ser *auditables* por cualquiera que lo desee.

2. El Proceso de Detección

Si tuviéramos que resumir en una frase *de que trata* o *qué se busca* en nuestro proyecto esta sería la siguiente: **“Aumentar la Capacidad de Detección”**

Introduciremos en este apartado los conceptos relacionados con la detección de redes que se desarrollarán a lo largo del documento.

Detectores

Definiremos un detector como cualquier programa capaz de procesar información en tiempo real, información normalmente a bajo nivel como tráfico o eventos de sistema y lanzar alertas ante la localización de situaciones previamente definidas.

La definición de estas situaciones se puede hacer de dos formas:

1. A través de patrones, o reglas definidas por el usuario
2. A través de grados de anomalía

La Capacidad de Detección

La capacidad de detección ha aumentado enormemente en los últimos años, pensemos por ejemplo en su máximo exponente los IDS, capaces de detectar patrones al nivel más bajo de detalle.

Para discutir sobre la capacidad de un detector la definiremos mediante 2 variables:

- *Sensibilidad* o la capacidad de análisis, en profundidad y complejidad, que posee nuestro detector a la hora de localizar un posible ataque.
- *Fiabilidad*, que como su nombre indica es el grado de certeza que nos ofrece nuestro detector ante el aviso de un posible evento.

La Incapacidad de Detección

Veremos a lo largo de este documento que pese a al desarrollo “en la profundidad de detección” de estos sistemas, nos encontramos muy lejos de que su capacidad sea aceptable.

De la incapacidad de los detectores de afrontar estas dos propiedades nos encontramos con los dos principales problemas de la actualidad:

- *Falsos Positivos*. La falta de fiabilidad en nuestros detectores es el causante del mayor problema actual, es decir alertas que realmente no corresponden con ataques reales.
- *Falsos Negativos*. La incapacidad de detección implicaría que un ataque es pasado por alto.

Podemos resumir los anteriores puntos en la siguiente tabla:

I. Capacidad de los Detectores

| | Propiedad | Efecto ante su ausencia |
|--------------|---|-------------------------|
| Fiabilidad | El grado de certeza que nos ofrece nuestro detector ante el aviso de un posible evento. | Falsos Positivos |
| Sensibilidad | La capacidad de análisis, en profundidad y complejidad, que posee nuestro detector a la hora de localizar un posible ataque | Falsos Negativos |

El “Proceso de Detección”

Llamaremos al “Proceso de Detección” al proceso global desarrollado por el SIM, incluyendo tanto los diferentes detectores y monitores de la organización como los realizados por el sistema para procesar esta información.

El Proceso de Detección implica normalmente tres fases bien distinguidas:

- **Preproceso:** La detección en si misma, la generación de alertas por los detectores y la consolidación previa al envío de información.
- **Colección:** El envío y recepción de toda la información de estos detectores en un punto central.
- **Postproceso:** El tratamiento que realizaremos una vez tenemos toda la información centralizada.

Postproceso

El preproceso y la colección son capacidades ya clásicas y no aportan nada nuevo, pero en el postproceso, una vez tenemos toda la información en un mismo punto, podemos implementar mecanismos para poder mejorar la sensibilidad y fiabilidad de la detección. Aumentaremos la complejidad del tratamiento incluyendo métodos que se encargarán de descartar falsos positivos o al contrario priorizar o descubrir patrones más complejos que nuestros detectores han pasado por alto.

En OSSIM desarrollaremos 3 métodos en el postproceso:

- 1. Priorización:** Donde priorizaremos las alertas recibidas mediante un proceso de contextualización desarrollado a través de la definición de una Política Topológica de Seguridad y el Inventariado de nuestros sistemas.
- 2. Valoración de Riesgo:** Cada evento será valorado respecto del Riesgo que implica, es decir, de una forma proporcional entre el activo al que aplica, la amenaza que supone y la probabilidad del evento.
- 3. Correlación:** Donde analizaremos un conjunto de eventos para obtener una información de mayor valor.

El siguiente cuadro muestra como afectan estos procesos a las anteriores propiedades:

II. Postproceso

| | Proceso | Efecto |
|----------------------|--|--|
| Priorización | Valoración de la amenaza mediante contextualización de un evento | Aumenta la Fiabilidad |
| Valoración de Riesgo | Valoración del Riesgo respecto del valor de activos | Aumenta la Fiabilidad |
| Correlación | Relación de varios eventos para obtener una información de mayor valor | Aumenta la Fiabilidad, Sensibilidad, y Abstracción |

Nuestro sistema se alimentará por lo tanto de “alertas” ofrecidas por los detectores y producirá tras el tratamiento de las mismas lo que en este documento llamaremos “alarmas”.

Una alarma será normalmente el resultado del proceso de varias alertas, tendrá normalmente un mayor grado de abstracción permitiendo localizar patrones más complejos, y ofrecerá un mayor grado de fiabilidad.

El Mapa de Detección

Con el objeto de definir esta Capacidad de Detección desarrollaremos en este proyecto un *Mapa de Detección*, este incluirá la categorización de las diferentes posibilidades en la detección de ataques y eventos de seguridad.

La Auditoría de la Capacidad de Detección

A través de este mapa podremos definir un nuevo método de auditoría que nos permitirá medir la situación y necesidades de las organizaciones respecto de la efectividad de sus sistemas a la hora de detectar ataques.

El punto de vista es muy diferente de la auditoría clásica o el test de intrusión, pues no nos interesa localizar los fallos de seguridad sino la capacidad de detectar el aprovechamiento de una utilización eventual de estos fallos.

Desarrollaremos en este proyecto por lo tanto un *Procedimiento de Auditoría de la Capacidad de Detección* mediante el cual proporcionaremos de un mecanismo para valorar la situación de la organización respecto de su capacidad de detectar ataques.

3. Funcionalidad

Para entender que ofrece OSSIM podemos definir la funcionalidad del sistema de una forma gráfica y simplificada con los 9 siguientes niveles:



Comentaremos cada uno de estos niveles para describir de una forma práctica nuestro sistema:

3.1 Detectores de Patrones

La mayoría de los detectores clásicos funcionan con patrones, el ejemplo más claro es el IDS o sistema de detección de intrusos, sistema capaz de detectar patrones definidos a través de *firmas* o reglas.

Existen otra serie de detectores de patrones incluidos en la mayoría de los dispositivos como routers o Firewalls, capaces de detectar por ejemplo scaneo de puertos, intentos de spoofing, o posibles ataques por fragmentación.

Tenemos además detectores para los eventos de seguridad de un sistema operativo, casi todos ellos incluyen su propio logger como el de Unix llamado syslog, siendo capaces de alertar de posibles problemas de seguridad.

En definitiva cualquier elemento de la red, como un router, un puesto de trabajo, el Firewall, etc incluye la capacidad de detección en mayor o menor medida, en nuestro sistema nos interesa recibir los eventos de todos los sistemas críticos para de esta forma obtener uno de nuestros objetivos principales: la Visibilidad de la red.

3.2 Detectores de Anomalías

La capacidad de detección de anomalías es más reciente que la de patrones. En este caso al sistema de detección no tenemos que decirle que es bueno o que es malo, él es capaz de “aprender” por sí solo y alertar cuando un comportamiento difiera lo suficiente de lo que ha aprendido como normal.

Esta nueva funcionalidad ofrece un punto de vista diferente y complementario a la detección de patrones pues la naturaleza de los dos procesos es opuesta.

La detección de anomalías puede ser especialmente útil para prevenir por ejemplo ataques perimetrales, estos son en sí una anomalía continua, en la dirección, el sentido de las comunicaciones, y el camino que definen, en el flujo de datos, el tamaño, el tiempo, el horario, el contenido, etc.

Esta técnica ofrece una solución - hasta ahora irresoluble - de control de accesos de usuarios privilegiados como son los ataques internos, por ejemplo de empleados desleales, los cuales no implican la violación de ninguna política ni la ejecución de ningún exploit. Implican sin embargo una anomalía en el uso y la forma de uso de un servicio.

Veamos otros ejemplos en los que estos detectores serían útiles:

- Un nuevo ataque del cual no existen todavía firmas podría traspasar los sistemas de detección de patrones pero producir una anomalía clara.
- Un gusano que se ha introducido en la organización, un ataque de spamming, o el mismo uso de programas P2P, generarían un número de conexiones anómalas fácilmente detectable.
- Podremos detectar así mismo:
 - Usos de servicios anormales por origen y destino
 - Usos en horario anormal
 - Exceso en el uso de tráfico o conexiones
 - Copia anormal de ficheros en la red interna
 - Cambios en el sistema operativo de una máquina
 - Etc

Podemos pensar que como efecto negativo estos detectores generarán un número de nuevas alertas, amplificando nuestra señal y empeorando nuestro problema (nuestro objetivo es limitar el número de alertas), sin embargo si las tomamos como información adicional que acompaña a las clásicas alertas de patrones, permitirá cualificar y por lo tanto diferenciar aquellas que puedan implicar una situación de mayor de riesgo.

3.3 Centralización / Normalización

La normalización y centralización (o agregación) tiene como objetivo unificar en una única consola y formato los eventos de seguridad de todos los sistemas críticos de la organización.

Todos los productos de seguridad poseen normalmente la capacidad de gestión centralizada a través de protocolos estándar, la agregación es por lo tanto sencilla utilizando estos protocolos. En OSSIM intentaremos normalmente no utilizar agentes y utilizar las formas de comunicación naturales de los sistemas.

La normalización implica la existencia de un “parser” o traductor que conozca los tipos y formatos de alertas de los diferentes detectores, necesitaremos desarrollar un trabajo de organización de la base de datos y adaptación de la Consola Forense para homogenizar el tratamiento y la visualización de todos estos eventos.

De esta forma podremos observar en la misma pantalla y con un mismo formato los eventos de seguridad de un determinado momento, ya sean del Router, el Firewall, del IDS, o del servidor Unix.

Al tener centralizados en la misma base de datos todos los eventos de la red obtendremos una gran “visibilidad” de lo que ocurre en ella, a partir de ese momento podremos como veremos a continuación desarrollar procesos que permitan detectar patrones más complejos y distribuidos.

3.4 Priorización

La prioridad de una alerta debe ser dependiente de la Topología y el Inventario de sistemas de la organización, las razones son bastante claras como muestran estos ejemplos

- a. Si una alerta que se refiere a un ataque al servicio IIS de Microsoft llega a una máquina con sistema operativo Unix y servidor Apache, la alerta debe ser despriorizada.
- b. Si existe una conexión sospechosa de un usuario sobre un servidor, el sistema debe:
 - o Darle máxima prioridad si el usuario es externo y ataca a la base de datos de clientes.
 - o Darle prioridad baja si el usuario es interno y ataca a una impresora de red.
 - o Descartarla pues es un usuario que normalmente hace pruebas contra un servidor de desarrollo.

Llamamos priorización al proceso de contextualización, es decir la evaluación de la importancia de una alerta respecto del escenario de nuestra organización. Este escenario está descrito en una base de conocimiento sobre la red del cliente formada por:

- Inventario de Máquinas y Redes (identificadores, s. operativo, servicios, etc)
- Política de Accesos (desde donde a donde está permitido o prohibido)

Para realizar estas tareas (así como la valoración de riesgos explicada en el siguiente apartado) disponemos de un Framework donde podremos configurar:

1. Política de Seguridad. O valoración de parejas activo-amenazas según la Topología y flujo de los datos.
2. Inventario
3. Valoración de activos
4. Valoración de amenazas (priorización de alertas)
5. Valoración de Fiabilidad de cada alerta
6. Definición de Alarmas.

A través de la cualificación realizaremos una de las partes más importantes del filtrado de alertas recibidas por los detectores, la cual debe realizarse a través del proceso continuo de tuning y realimentación de la situación de nuestra organización.

3.5 Valoración del Riesgo

La importancia que debemos dar a un evento debe ser dependiente de estos tres factores:

- a. El valor del Activo al que el evento se refiere
- b. La Amenaza que representa el evento
- c. La Probabilidad de que este evento ocurra

Riesgo Intrínseco

Con ellos construimos la definición clásica de riesgo: el valor del posible impacto de una amenaza sobre un activo ponderado con la probabilidad de que este ocurra.

La valoración de riesgos se ha referido clásicamente a riesgos intrínsecos, o riesgos latentes, es decir riesgos que soporta una organización derivados del hecho de “ser” (los activos que posee para desarrollar su negocio) y “estar” (las amenazas circunstanciales que existen sobre estos activos).

Riesgo Instantáneo

En nuestro caso, debido a la capacidad de medir en tiempo real, podremos medir el riesgo asociado a la situación actual, en términos instantáneos.

En este caso el riesgo será medido como la medida ponderada del daño que produciría y la probabilidad de que este ocurriendo en este momento la amenaza.

Esta probabilidad, derivada de la imperfección de nuestros sensores, no será más que el grado de fiabilidad de estos en la detección de la posible intrusión en curso.

Llamaremos Riesgo Instantáneo a la situación de riesgo producida por la recepción de una alerta, valorada de forma instantánea como la medida ponderada entre el daño que produciría el ataque y la fiabilidad del detector que lo reporta.

OSSIM calculará el Riesgo Instantáneo de cada evento recibido que será la medida objetiva que utilizaremos para valorar la importancia que un evento puede implicar en términos de seguridad, sólo a través de esta medida valoraremos la necesidad de actuar.

Incluiremos en nuestro sistema así mismo un Monitor de Riesgos descrito posteriormente que valorará el riesgo acumulado en el tiempo de redes y grupos de máquinas relacionados en un evento.

3.6 Correlación

Definimos una función de correlación como un algoritmo que realiza una operación a través de unos datos de entrada y ofrece un dato de salida.

Pensemos en la información recogida por nuestros detectores y monitores como información específica pero parcial, dibujando pequeñas zonas del espectro de toda la información que nos interesaría tener.

Podemos pensar en la capacidad de correlación como la de aprovechar estos sistemas y a través de una nueva capa de proceso llenar otras zonas de ese espectro infinito de toda la información que podría existir de una red.

En contra de esta idea podría intentar instalarse un sistema único con un detector capaz de localizar toda la información posible de la red, pero para ello necesitaríamos una visibilidad total desde un punto único y una capacidad de almacenamiento y de memoria casi ilimitada.

Los sistemas de correlación son por tanto artificios que suplen la falta de sensibilidad, fiabilidad y la visibilidad limitada de nuestros detectores.

Entrada y Salida

En nuestra arquitectura de una forma simplificada podemos decir que tenemos dos elementos claramente diferenciados para ofrecer información a nuestras funciones de correlación:

- Los Monitores. Que nos ofrecerán normalmente indicadores.
- Los Detectores. Que nos ofrecerán normalmente alertas.

Como salida obtendremos también uno de estos dos elementos: alertas o indicadores. Nuestras funciones se habrán convertido en nuevos detectores o monitores.

3.6.1 Modelo de Correlación

OSSIM desarrolla un modelo de correlación tan ambicioso como para poder:

1. Desarrollar *patrones específicos* para detectar lo conocido y detectable
2. Desarrollar *patrones ambiguos* para detectar lo desconocido o no detectable
3. Poseer una *máquina de inferencia* configurable a través de reglas relacionadas entre sí capaz de describir patrones más complejos
4. Permitir *enlazar* Detectores y Monitores de forma recursiva para crear cada vez objetos más abstractos y capaces
5. Desarrollar algoritmos que ofrezcan una visión general de la Situación de Seguridad.

3.6.2 Métodos de Correlación

Para lograr estos objetivos utilizaremos dos métodos de correlación muy diferentes que intentaremos describir mediante sus diferencias principales:

- **Correlación mediante Secuencias de Eventos.** Focalizado en los ataques conocidos y detectables, relaciona a través de reglas que implementarán una máquina de estados, los patrones y comportamientos conocidos que definen un ataque.
- **Correlación mediante Algoritmos Heurísticos.** Tomando una aproximación opuesta implementaremos algoritmos que mediante funciones heurísticas intenten detectar situaciones de riesgo. Este método detectará situaciones sin conocer ni ofrecer detalle de los mismos, intenta suplir pues la incapacidad de los anteriores métodos y será útil para detectar ataques no conocidos y obtener una visión general del estado de seguridad para un amplio número de sistemas.

3.6.2.1 Método 1: Correlación mediante Algoritmos Heurísticos

En OSSIM implementaremos un sencillo algoritmo heurístico de correlación por acumulación de eventos con el objetivo de obtener un indicador o una fotografía del estado general de seguridad de la red.

El primer objetivo de este es recibir lo que hemos definido previamente como “riesgo instantáneo” obteniendo como resultado un valor que podríamos definir como el Nivel Acumulado de Riesgo.

Obtendremos una monitorización a alto nivel que nos servirá como “termómetro” de situaciones de riesgo sin conocer en ningún momento detalle de las características del problema.

Por hacer un símil construiremos un termómetro que será sensible y sumará la cantidad de riesgo acumulado en una ventana de tiempo, el termómetro subirá proporcionalmente a la cantidad y lo “calientes” que sean los últimos eventos recibidos, y se enfriará con el paso del tiempo en caso de no recibir nuevos eventos.

Este método de correlación quiere suplir con un punto de vista opuesto a la correlación mediante secuencias de eventos, donde intentaremos caracterizar al máximo nivel de detalle los posibles ataques.

Su interés es pues doble:

- El de ofrecer una visión global rápida de la situación.
- Detectar posibles patrones que al resto de sistemas de correlación puedan pasar por alto, ya sea por tratarse de ataques desconocidos o por falta de capacidad.

CALM

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.

La acumulación se realiza para cualquier sujeto de la red, entendiendo como tal a cualquier máquina, grupo de máquinas, segmento de red, camino.. que nos interese monitorizar.

Acumulación de Eventos

La acumulación se realiza a través de la simple suma del riesgo instantáneo de cada evento en dos variables de estado:

- **La “C” o el Nivel de Compromiso**, que mide la posibilidad de que una máquina se encuentre comprometida.
- **La “A” o el Nivel de Ataque** al que está sometido un sistema, que mide el posible riesgo debido a los ataques recibidos.

¿Por que separar estas dos variables en nuestra monitorización? En primer lugar porque caracterizan situaciones diferentes: el Nivel de Ataque indica la posibilidad de estar recibiendo un ataque, ataque que podrá o no tener éxito. El Nivel de Compromiso ofrece evidencia directa como su nombre indica de que ha habido un ataque y ha tenido éxito.

En segundo lugar la importancia de cada una de las dos variables será dependiente de la situación de la máquina. Principalmente debido a la exposición de las redes perimetrales, expuestas a multitud de ataques la mayoría de ellos automatizados y para las cuales desgraciadamente un alto valor del Nivel de Ataque ha de ser una "situación normal". Para estas redes sin embargo el indicador de Compromiso, o movimiento que pueda hacer pensar que hay un atacante alojado en ellas debe ser inmediatamente notificado y revisado.

Al contrario, hay casos en los que una máquina que por su función genera anomalías en la red como un scanner de seguridad, un servicio con puertos pasivos aleatorios, desarrollo... tendrá normalmente una C alta y una A baja.

La asignación del valor a las variable C o A de una máquina de la red se produce a través de 3 reglas:

1. Cualquier posible ataque que se produzca desde una máquina 1 a una máquina 2 aumentará la A (el nivel de ataques recibidos) de 2 y la C (el nivel de compromiso o acciones sospechosas que normalmente haría un hacker) de 1.
2. El caso de tratarse de una respuesta de ataque ("attack responses" o respuestas que pueden implicar que el ataque a tenido éxito), en este caso aumentará el nivel de C tanto en 1 como en 2.
3. En caso de ser eventos internos aumentará únicamente la C de la máquina originaria.

Acumulación en el Tiempo

CALM está pensado para la monitorización en tiempo real, por lo que nuestro interés es una ventana de tiempo en el corto plazo, es decir nos interesa la valoración de eventos de un espacio de tiempo cercano, el algoritmo debe tener una memoria en el corto plazo primando los eventos más recientes y caducando los más antiguos.

La implementación actual es a través una simple variable de recuperación en el tiempo. El sistema irá rebajando con un valor constante de forma periódica los niveles de C y A de cada máquina.

3.6.2.2 Método 2: Correlación mediante Secuencias de Eventos

El Panel de Secuencias

La idea inicial de la detección de una secuencia de patrones es sencilla pues sería simplemente realizar una lista de reglas “si ocurre el evento A y luego B y luego C, haz la acción D”.

Esto lo realizaremos a través del Panel de Secuencias, donde definiremos listas de reglas para cada secuencia de eventos que queramos definir.

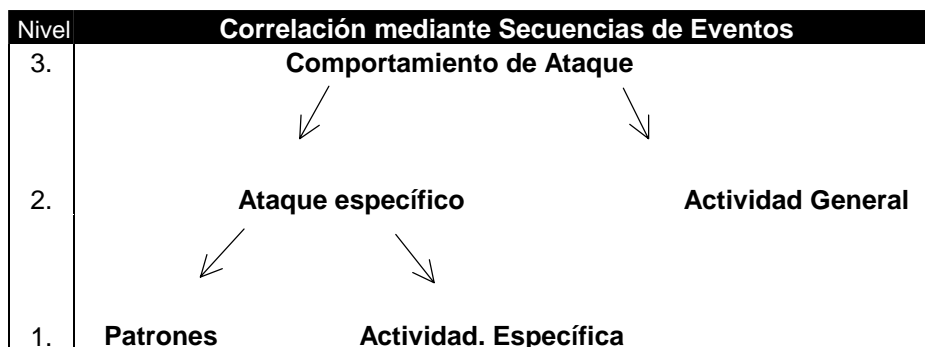
La complejidad del panel dependerá de la capacidad de abstracción que permitan sus reglas y la posibilidad de analizar diferentes entradas en nuestras funciones.

En OSSIM nuestro panel será capaz de realizar secuencias con las siguientes características:

- Posibilidad de definir orígenes y destinos variables
- Tomar como entrada tanto patrones procedentes de detectores como indicadores procedentes de monitores
- Definir el nivel de *prioridad* y *fiabilidad* de las nuevas alertas
- Utilizar variables “elásticas” o capaces de medir el grado para definir la prioridad o fiabilidad (ej. Denegación de servicio: total -> prioridad grave, 50% -> prioridad media, 15% prioridad baja).
- Arquitectura recursiva, podremos crear objetos a través de la correlación de reglas que se podrán incluir en nuevas reglas como detectores o monitores

3.6.3 Niveles de Correlación

Debido a la recursividad de nuestro modelo se podrá crear una jerarquía de niveles casi infinita, para poder centrar nuestro estudio definiremos una jerarquía de 3 niveles tal y como se muestra en el siguiente gráfico:



Recorreremos cada uno de estos niveles desordenadamente para su mejor entendimiento:

Nivel 2.1. Ataque Específico

Este nivel trata directamente con los detectores y monitores, intentaremos relacionar tanto las firmas como la actividad que se refiera a un ataque concreto, un ataque con nombre y apellidos tal y como lo conocen los detectores (por ejemplo: "compromiso mediante ftp cwd overflow").

El principal objetivo del nivel de ataque específico es el de aumentar la *fiabilidad* de las detecciones, esto es, no nos bastará con la firma de la posibilidad de un ataque, sino que buscaremos más evidencias que nos demuestren que se está produciendo el ataque o clarifique que es únicamente un intento fallido.

Esta *cualificación* es la que hará la diferencia a la hora de limitar falsos positivos y priorizar ataques reales en un sistema de detección de seguridad, ya que como hemos visto la fiabilidad de un evento afecta directamente al cálculo del riesgo.

Pensemos en un ejemplo sencillo de correlación de un detector de patrones y un monitor:

El IDS detecta mediante una firma un posible ataque de denegación de servicio mediante "synflood", la alerta de este arrancará una pregunta al Monitor de Servicio para ver si este ha sufrido un decremento de indisponibilidad y en que grado. De esta forma podremos añadir un grado de fiabilidad mayor a nuestra alerta "Denegación de servicio por Synflood".

Normalmente tendremos secuencias más complejas, donde correlacionaremos las alertas producidas por firmas con los comportamientos específicos que caracterizan un ataque. Pensemos en la detección de un Caballo de Troya, las operaciones que somos capaces de detectar a través de firmas de IDS son varias:

Connect, active, get info, access, server2client_flow, client2server_flow, response, traffic_detect

La detección de una operación *connect* probablemente no es una información de gran valor, en entornos perimetrales se reciben decenas al día, pero si detectamos cualquier otra operación y en especial de respuesta al intento de conexión deberemos enviar una alerta con prioridad alta.

Nivel 1.2. Detección por Actividad Específica

Aprovecharemos el ejemplo del Caballo de Troya para explicar el concepto de "actividad específica", pensemos en un caso simple: tras un intento de conexión, si el Caballo de Troya opera a través del puerto P, fijémonos simplemente si este puerto tiene actividad, es decir transmite datos. Si esto ocurre tendremos como antes una confirmación de que el intento de conexión probablemente ha tenido éxito, pero esta vez en vez de ser una firma de un IDS lo hemos localizado monitorizando la actividad propia del Troyano.

La Actividad Específica implica la utilización de los monitores para atender una pregunta concreta sobre la actividad asociada a un posible Ataque Específico, serán consultas que arrancará y matará el motor de correlación para un caso concreto.

Nivel 1.1. Detección Mediante Patrones

Este nivel ya se ha comentado y nos viene proporcionado directamente por los detectores de patrones, nuestro sistema de correlación será capaz de procesar cualquier alerta detectada por estos.

"Respuestas de Ataque"

Haremos un alto para sacar una conclusión de los dos puntos anteriores: la importancia de las *repuestas de ataque* ("attack responses" en los IDS) como comprobación de la existencia de un evento, o lo que es lo mismo aumento de la fiabilidad de una alerta.

Nuestro motor de correlación está diseñado para buscar continuamente estas respuestas de ataque, tras recibir la primera información de un posible ataque pondremos todo nuestro sistema a localizar evidencias de que el ataque realmente se está produciendo. Esto nos permitirá diferenciar ataques fallidos de los que realmente han tenido éxito.

Gráficamente podríamos dibujarlo para nuestro ejemplo de la siguiente forma:

Aumento de la fiabilidad para Caballo de Troya

| Sucesión de Eventos | Tipo de Alerta | Fiabilidad |
|---|---|------------|
| 1. Intento compromiso o conexión caballo troya | Firma: conexión, client2server, access, getinfo | Baja |
| | Act. Específica: flujo atacante -> víctima | |
| 2. Detección de respuesta típica de éxito del ataque o respuesta a operación de caballo troya | Firmas: response, server2client | Alta |
| | Activ. Específica: flujo víctima->atacante | |

Nivel 2.2. Detección por Actividad General

Llamaremos detección por Actividad General a las reglas destinadas a localizar ataques no conocidos o no detectables pues no tenemos las firmas o patrones que caracterizan este ataque.

La localización de estos ataques será gracias a la generación de actividad anómala por parte del atacante, para ello monitorizaremos parámetros generales de cada usuario tales como los puertos o servicios, el tráfico, el horario, etc.

Mediante esta técnica podremos caracterizar ataques con cierto detalle en algunos casos, pero generalmente detectaremos comportamientos sospechosos, con un nivel menor de precisión que en la detección de Ataques Específicos y nos moveremos muchas veces en la frontera entre lo que es un ataque, un problema de red o el mal uso por parte de los usuarios.

Ejemplos de esta detección serían:

- Detección de un gusano desconocido. Que generará un tráfico anormal, un número de conexiones atípico con puertos y destinos que hasta ahora no habían sido usados.
- Detección de acceso sospechoso. Al localizar un usuario conectado de forma persistente a un puerto de administración, que hasta ahora no lo había hecho.
- Exceso de uso de tráfico. A través de anomalías de destinos y utilización.

Nivel 3. Comportamiento de Ataque

Este tercer nivel de correlación se alimenta y es principalmente la correlación de varios Ataques Específicos o Comportamientos Generales localizados en el primer nivel.

Recordemos que la arquitectura de nuestro sistema de correlación es recursiva y en nuestras reglas podremos incluir nuevos objetos que funcionarán como detectores (enviando alertas) o monitores (ofreciendo un valor) que están formados por un conjunto de reglas del nivel inferior.

Pero la caracterización de los nuevos niveles no debemos hacerla del hecho de que los objetos de entrada sean procedentes del nivel inferior, al contrario, esto no será así siempre y podremos mezclarlos según nos convenga.

La caracterización de cada nivel debe ser debida al nivel de abstracción al que se refiera, y en este caso intentaremos localizar patrones de comportamiento que nos caractericen cual es el *objetivo*, el *camino trazado*, el *comportamiento del atacante*, el *método del mismo*. Para ello definiremos Comportamientos de Ataque o la secuencia de ataques y comportamientos desarrollada por el usuario sobre una o varias máquinas comprometidas.

Ejemplos de estos comportamientos podrían ser:

- Ataque distribuido. Al encontrar relación de varios atacantes y ataques recibidos.
- Acceso a red crítica desde Internet. Siguiendo el flujo de un ataque perimetral que desde Internet llega en varios saltos a una red crítica.
- Compromiso usuario Interno Malicioso. A través de la localización de varios comportamientos anormales de un usuario interno.

3.7 Monitores

Aunque son simples monitores de procesos realizados anteriormente, como tal son funcionalidades que vale la pena destacar:

3.7.1. Monitor de Riesgos

OSSIM posee un monitor de riesgos que hemos llamado RiskMeter y que dibujará los valores producidos por el algoritmo CALM, valores que miden el nivel de riesgo de compromiso (C) y de ataque (A) derivados de la recepción de alertas que indican la posibilidad de que una máquina ha sido comprometida, o está siendo atacada.

3.7.2. Monitor de Uso, Sesiones y Perfiles

En OSSIM le damos mucha importancia como se explica en el apartado de anomalías a la monitorización detallada de cada máquina y perfil.

Podemos diferenciar 3 tipos de monitorización para ello:

- Monitor de Uso: Nos ofrece datos generales de la máquina como el número de bytes que transmite al día.
- Monitor de Perfiles. Nos ofrece datos específicos del uso realizado por el usuario y permite establecer un perfil, por ejemplo: usa correo, pop, y http, es un perfil de usuario normal.
- Monitor de Sesiones. Nos permite ver en tiempo real las sesiones que está realizando el usuario. Nos ofrece una foto instantánea de la actividad de esta máquina en la red.

Creemos que cualquiera de estos 3 son imprescindibles para un sistema de seguridad, en caso contrario, el administrador de seguridad estará ciego ante eventos pasados, no podrá distinguir lo normal de lo anormal y no será capaz de ver su red, sería semejante a un guarda de tráfico en una carretera completamente oscura.

Aquí la frontera de seguridad se confunde con la administración de redes, pero este solape es inevitable pues la saturación de una red o el comportamiento anómalo de una máquina puede significar tanto un problema de red como un problema de seguridad.

Incluiremos en OSSIM las 3 capacidades de monitorización anteriormente expuestas a través de productos capaces de actuar como “sniffers” y ver al máximo nivel de detalle la situación de la red.

3.7.3. Monitor de Caminos

Este monitor es capaz de dibujar en tiempo real los caminos trazados en nuestra red entre las diferentes máquinas que realizan conversaciones o enlaces entre ellas.

El dibujo se realiza en un intervalo de tiempo creando un grafo cuyas ramas irán caducando en el tiempo.

El monitor obtiene sus datos de otros dos monitores: el de sesiones donde están localizados cada uno de los enlaces del momento, y del monitor de riesgo de donde obtiene el nivel de riesgo de cada máquina para dibujar cada una con un color diferente y calcular el riesgo agregado de cada uno de estos grafos.

La monitorización de enlaces tiene a su vez dos métodos:

3.7.3.1. Hard Link Analysis (TCP Link Analysis)

Mediante el cual dibujaremos únicamente sesiones TCP persistentes. Método desarrollado con la intención de localizar ataques de red que implican la intrusión de varias máquinas de forma continuada, situación típica de una intrusión perimetral.

3.7.3.2. Soft Link Analysis

Mediante el cual dibujaremos todos los enlaces percibidos en la red, tanto udp como tcp como icmp lo cual puede implicar en muchos casos un mapa de red caótico.

3.8. Consola Forense

La Consola Forense permite acceder a toda la información recogida y almacenada por el colector.

Esta consola es un buscador que ataca a la base de datos de eventos, y permite al administrador analizar a posteriori y de una forma centralizada los eventos de seguridad de todos los elementos críticos de la red.

Al contrario que el Monitor de Riesgos referido en el apartado anterior, esta consola nos permitirá profundizar al máximo detalle sobre cada uno de los eventos ocurridos en el sistema.

3.9 Cuadro de Mandos

La última de las funcionalidades es el Cuadro de Mandos, mediante él podremos ofrecer una visión de alto nivel de la situación de nuestra red respecto a seguridad.

El cuadro de mandos monitoriza una serie de indicadores que miden el estado de la organización respecto de seguridad.

Nos permitirá definir una serie de umbrales u objetivos que debe cumplir nuestra organización. Estos umbrales serán definidos de forma absoluta o relativa como un grado de anomalía.

Podremos asignar el envío de alarmas cuando se superen estos umbrales o la ejecución de cualquier procedimiento automático.

Es importante así mismo la forma de visualización de la información en este cuadro de mandos pues debe ser lo más concisa y simple posible. Para ello necesitaremos una configuración versátil que muestre únicamente la información relevante en ese momento.

El cuadro de mandos debe ser nuestro termómetro general de todo lo que ocurre en la red. A través de él enlazaremos con cada una de las herramientas de monitorización para profundizar sobre cualquier problema localizado.

Como ejemplo podríamos visualizar los siguientes datos:

- Monitorización permanente de los niveles de riesgo de las principales redes de la organización.
- Monitorización de las máquinas o subredes que superan el umbral de seguridad.
- Monitorización permanente de parámetros generales de red, sistema y niveles de servicio:
 - Throughput y Tráfico de principales redes
 - Recursos de la base de datos principal
 - Latencia de Servicios críticos
 - Número de transacciones de servicios críticos
- Monitorización de aquellos parámetros de red o niveles de servicio que superen el umbral establecido:
 - Número de correos, virus, accesos externos
 - Latencia de servicios, uso de tráfico por servicios
- Monitorización de perfiles que superen los umbrales por:
 - Uso de tráfico
 - Uso de servicios críticos
 - Uso de servicios anómalos
 - Cambios en configuración
 - Cualquier otra anomalía de comportamiento

El cuadro de mandos debe ser bajo nuestro punto de vista algo completamente personalizado y hecho a medida. Al contrario que para el resto de funcionalidades, OSSIM sólo incluirá un ejemplo base sobre el cual trabajar.

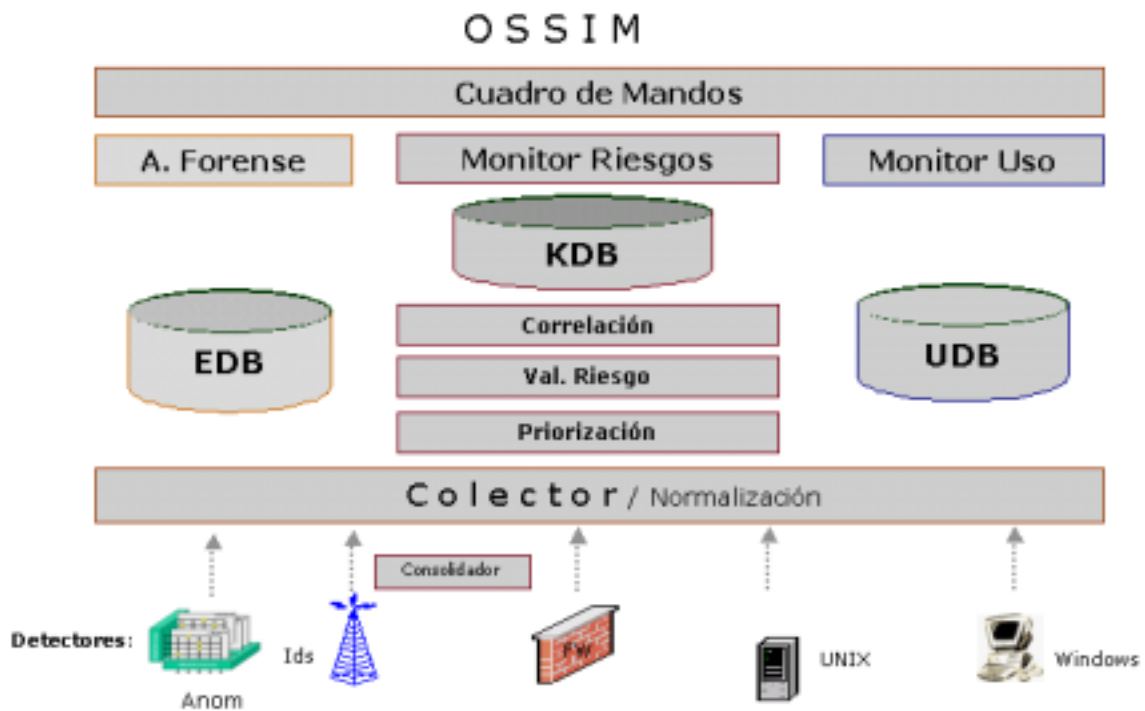
4. Arquitectura

4.1. Arquitectura General

El sistema contará como es común con dos partes diferenciadas, en ellas se desarrollan dos momentos diferentes del proceso:

- Preproceso: que se realizará en los propios monitores y detectores
- Postproceso: que se realizará en una consola centralizada

El dibujo general de la arquitectura según los procesos realizados es el siguiente:



En ella se perciben cada una de las funcionalidades anteriormente descritas. Así mismo vemos 3 bases de datos:

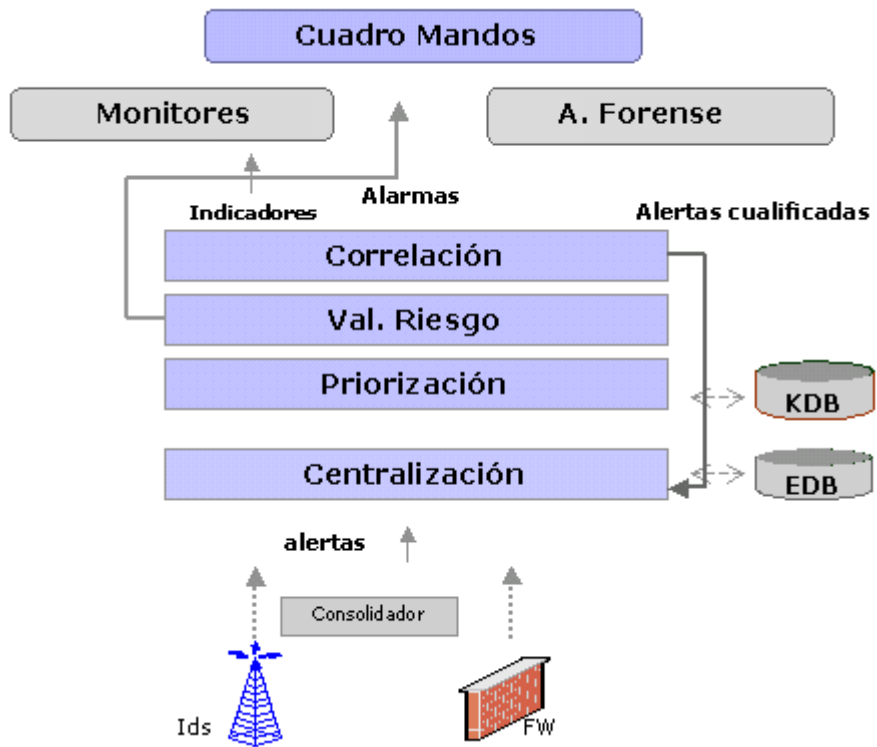
- EDB, la base de datos eventos, la más voluminosa pues alojará todos los eventos individuales recibidos de nuestros detectores.
- KDB, la base de datos del Framework, en la cual parametrizaremos el sistema para que conozca nuestra red y definiremos nuestra política de seguridad.
- UDB, la base de datos de perfiles, que almacenará todos los datos aprendidos por el Monitor de Perfiles

4.2 Flujo de los datos

Para entender la integración de cada uno de los productos haremos un recorrido del flujo desde la generación de un evento:

1. Los eventos son procesados por los detectores hasta que bien por la localización de un patrón o una anomalías se produce una alerta.
2. Las alertas son procesadas en caso de ser necesario por los consolidadores antes de ser enviadas. Estos se encargarán de enviar la información agrupada para ocupar el mínimo ancho de banda.
3. Las alertas son recibidas por el colector a través de diferentes protocolos abiertos de comunicación.
4. El parser se encarga de normalizarlas y guardarlas si procede en base de datos de eventos.
5. El parser se encarga así mismo de cualificarlas determinando su prioridad según la política de seguridad definida en el framework y los datos sobre el sistema atacado localizados en el Inventario de Sistemas.
6. El parser valora el riesgo instantáneo que implica la alerta y en caso de ser necesario envía una alarma al Cuadro de Mandos.
7. 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 fiable. Estas alertas son enviadas de nuevo al parser para su almacenamiento, priorización, valoración del riesgo, etc.
8. El monitor de riesgos visualizará periódicamente la situación de cada uno de los índices de riesgo según han sido calculados por CALM.
9. 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.
10. 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.
11. Podrá además comprobar el estado instantáneo de la máquina a través de los monitores de uso, perfiles, y sesiones.

El siguiente gráfico muestra el flujo de los datos:



4.3 Arquitectura de la Distribución

OSSIM se define así mismo como una “distribución” en vez de un producto, esto significa que su objetivo es integrar antes que desarrollar. Nuestro desarrollo irá siempre encaminado a poder “pegar” o hacer que estos productos se hablen entre ellos.

Sin embargo este desarrollo cada día es más complicado y por ello se han definido dos niveles:

4.3.1 El Núcleo

Lo que una distribución típica llamaría el Núcleo, desarrollado en el proyecto GNU de OSSIMy donde se desarrollan las siguiente tareas:

- Se define la estructura de datos
- Se ofrece los interfaces para hablar con los diferentes productos
- Se realiza lo que anteriormente hemos llamado como “postproceso”
- Se ofrece el la primera capa de administración a través de un Framework, que enlaza con los diferentes sistemas de administración
- Se implementa el Cuadro de Mandos

4.3.2 Productos Terceros

Productos terceros no desarrollados en nuestro proyecto pero integrados en la solución. De este tipo de productos hay dos posibilidades:

- Productos Open Source. Que dependiendo de los casos serán modificados y/o parcheados y la mayoría de las veces irán incluidos en la distribución.
- Productos de pago. Que lógicamente no irán en la distribución ni serán parcheados ni modificados.

Para ver la relación entre la arquitectura y consultar la sección “Road Map”.

5. Proyecto y Contactos

OSSIM es un proyecto abierto con la ambición de nutrirse de la comunidad open source y el mundo de desarrolladores e integradores de soluciones de seguridad, por ello cualquier aportación, comentario o discusión sobre el contenido de este documento será bienvenida.

Estamos desarrollando un programa de colaboraciones que incluirá todos los temas que abarca nuestro proyecto como son la participación en el desarrollo del software, la definición de la arquitectura, desarrollo de documentación y metodologías, portabilidad en diferentes plataformas, traducción, etc.

Las direcciones de contacto actuales son:

Página oficial: <http://www.ossim.net>

Página del proyecto: <http://sourceforge.net/projects/os-sim/>

Email contacto: contact@ossim.net

Email soporte: support@ossim.net

Releases: <http://freshmeat.net/projects/os-sim/>

CVS: <http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/os-sim>