



# OSSIM, una plataforma clave para la seguridad en profundidad

Angel Alonso Párrigas

Grado de dificultad



Hoy en día, en la era de la información, existe un riesgo muy grande de caer en problemas de sobre información o exceso de información. En la administración de sistemas y seguridad esto ocurre cuando tenemos decenas de máquinas y sistemas que gestionar y no somos capaces de discernir la información realmente importante de la que no lo es.

**E**n una arquitectura de red segura debe existir una segmentación correcta de los servicios y las aplicaciones mediante DMZs, Screened Subnet, VLANs y Private VLANs. Además, debe de diseñarse con diferentes capas de protección, desde el dispositivo más perimetral hasta el servidor interno que alberga la base de datos. Es lo que comúnmente se conoce como 'defense in depth' (seguridad en profundidad).

De manera práctica aplicar la seguridad en profundidad en una red sencilla como la de la figura 1 implicaría la creación de listas de accesos en el router que comunica con el exterior, una buena política de seguridad plasmada en las reglas del firewall (política de denegación de todo por defecto), monitorización de los eventos de la red mediante sistemas de detección de intrusos de red (NIDS), el bastionado de los sistemas finales que albergan los servicios y la instalación de herramientas de detección de intrusos de host (HIDS). Se podría incluso instalar sistemas de prevención de intrusos que repelieran algunos de los ataques, tanto en los hosts finales como en la red. Figura 1.

Pero tan importante como la instalación y configuración de todas estas capas de protección es monitorizar los eventos que ocurren en nuestra red/sistemas y que muchos de estos elementos hacen de manera nativa.

Es fundamental tener una red bien protegida que nos evitará muchos dolores de cabeza, pero aún estando bien protegidos siempre cabe la posibilidad de sufrir un ataque real y si se da el caso, éste debe ser detectado y contenido en la mayor brevedad posible. Es por ello que la detección mediante la monitorización juega un papel fundamental.

## En este artículo aprenderás

- qué es la correlación
- qué es la seguridad en profundidad
- monitorización de eventos de seguridad

## Lo que deberías saber

- administración de sistemas
- conocimientos básicos Scirotrig

**Listado 1. Relación de eventos**

```

Si escaneo' de IP_A a IP_B
si 'exploit' de IP_A:PUERTO_X
    a IP_B:PUERTO_Y
Si 'socket' abierto de IP_B:
    PUERTO_Y hacia IP_A
Si 'comando detectado' en IP_B.
entonces MANDAR_ALERTA();
fsi
fsi
fsi
fsi

```

**OSSIM como sistema de monitorización**

La mayoría de electrónica de red (balanceadores, routers, comunicadores LAN, firewalls, sistemas de prevención de intrusos de red, etc), al igual que los sistemas operativos y las aplicaciones que corren en los mismos, generan información en forma de logs que es de gran utilidad para analizar una violación de la seguridad o cualquier problema. Pero si disponemos de una red de cierta envergadura la cantidad

de información generada puede desbordar a cualquier Administrador de Seguridad que tendrá que revisar cantidad de logs y que al final desistirá en su tarea. Esta es una de las razones por las que es fundamental tratar al conjunto de la información generada por todos los sistemas, con el fin de ser capaces de detectar aquellos eventos relevantes y descartar aquellos sin importancia.

OSSIM es una plataforma de seguridad que permite tratar la información generada, almacenarla y priorizarla mediante técnicas de correlación. Facilita la integración de herramientas open source y herramientas propietarias, lo que permite tener una visión completa de la seguridad de la red desde una perspectiva de confidencialidad, integridad y disponibilidad.

Otra de las funcionalidades de OSSIM es la posibilidad de inventariar activos (IP, Sistema Operativo, MAC, valor del activo, etc) que fa-

cilita la tarea para un administrador de sistemas a la hora de localizar un sistema y permite tener un mayor control sobre qué ocurre en la red de un instante determinado. Este inventariado se puede realizar de manera pasiva con herramientas como p0f, de manera activa con Nmap o bien de manera manual.

Otra de las funciones de OSSIM es que permite ver la tendencia de la seguridad a lo largo del tiempo mediante gráficas y cuadros de mandos, se puede ver lo ocurrido en una ventana de tiempo con la consola forense, sacar informes estadísticos detallados, en resumen, se puede tener una visión de la seguridad desde un punto de vista técnico o un ángulo más cercano a un responsable de seguridad.

**Herramientas que integra OSSIM**

Como se ha comentado OSSIM por defecto permite trabajar con herramientas libres y propietarias.

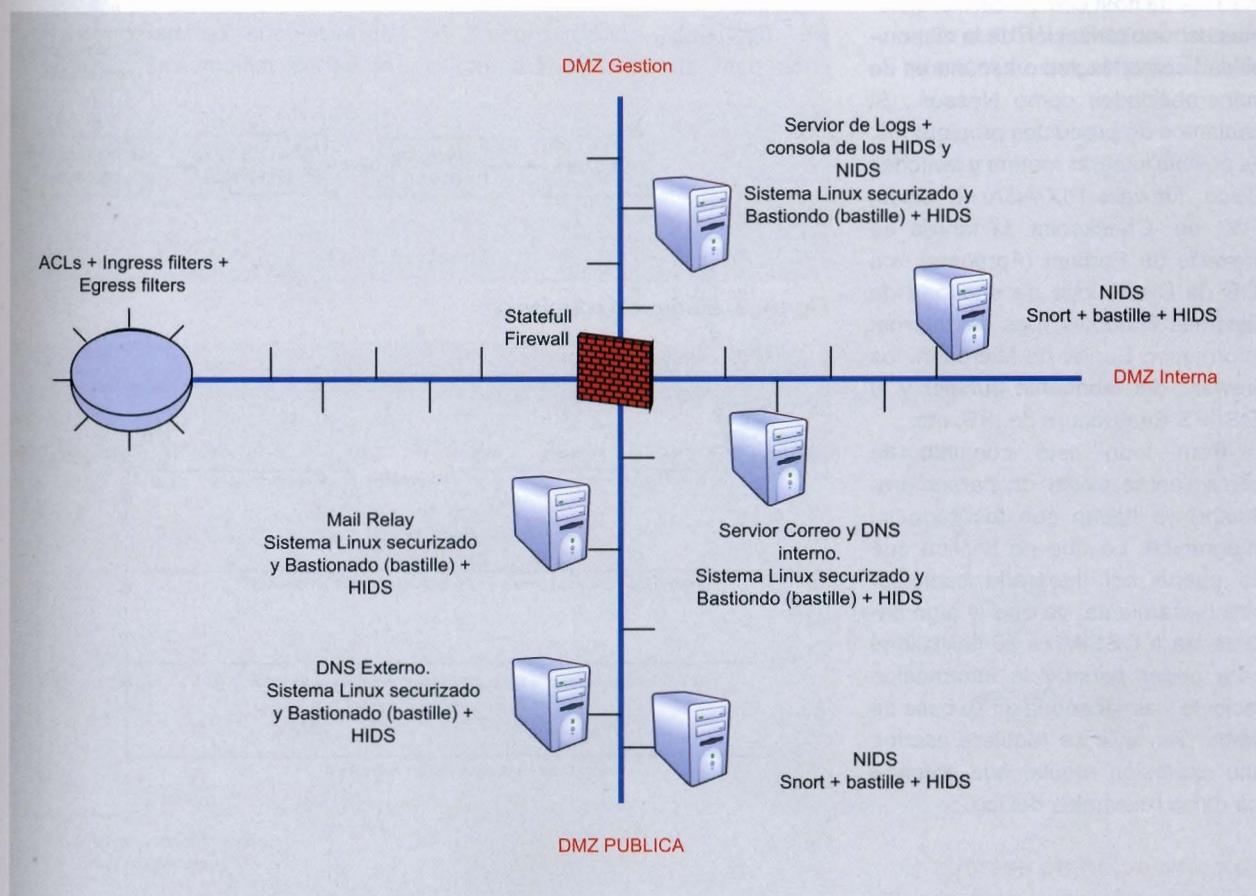


Figura 1. Defensa en profundidad con OSSIM

**Listado 2. Logos de POF**

```
--- p0f 2.0.5 resuming operations at <Tue Aug 14 15:46:02 2007> ---
<Tue Aug 14 15:46:10 2007> 192.168.40.1:24512 - Windows 2000 SP4, XP SP1
<Tue Aug 14 15:46:11 2007> 192.168.40.2:18902 - Windows 2000 SP4, XP SP1
<Tue Aug 14 15:46:12 2007> 192.168.40.3:40078 - Windows 2000 SP4, XP SP1
```

Cualquier dispositivo que permita enviar información a un servidor de logs remoto (syslog) puede servir para alimentar a OSSIM. En lo que se refiere a productos open source podemos trabajar con los logs de aplicaciones como Arpwatch, que permite ver que ocurre a nivel 2 (como cambios inusuales en direcciones MAC), logs de un IDS como Snort o Prelude que nos daría una visión de ataques a nivel de red, monitorización de flujos de tráfico mediante Ntop, sesiones TCP establecidas con Tcptrack, conexiones aceptadas o denegadas por el firewall Iptables, accesos a servidores Apache, logs de auditoria de los sistemas Unix, logs de sistemas de detección de intrusos de host como OSIRIS, sistemas de monitorización de la disponibilidad como Nagios o escáneres de vulnerabilidades como Nessus.. Si hablamos de productos propietarios, es posible integrar routers y switches Cisco, firewalls PIX/ASA de Cisco, FW1 de Checkpoint, la familia de firewalls de Fortinet (Fortigate), los IDS de Cisco, logs de auditoría de sistemas Windows, logs del Internet Information Server de Microsoft, los firewalls del fabricante Juniper y el IDS/IPS RealSecure de ISS, etc.

Para todo este conjunto de herramientas existe un parser (traductor) ya hecho que facilitaría su integración. Lo que no implica que no pueda ser integrada cualquier otra herramienta, ya que si algo caracteriza a OSSIM es su flexibilidad para poder parsear la información recibida y almacenarla en su base de datos. Tan solo se requiere escribir una expresión regular que obtenga los datos relevantes del log.

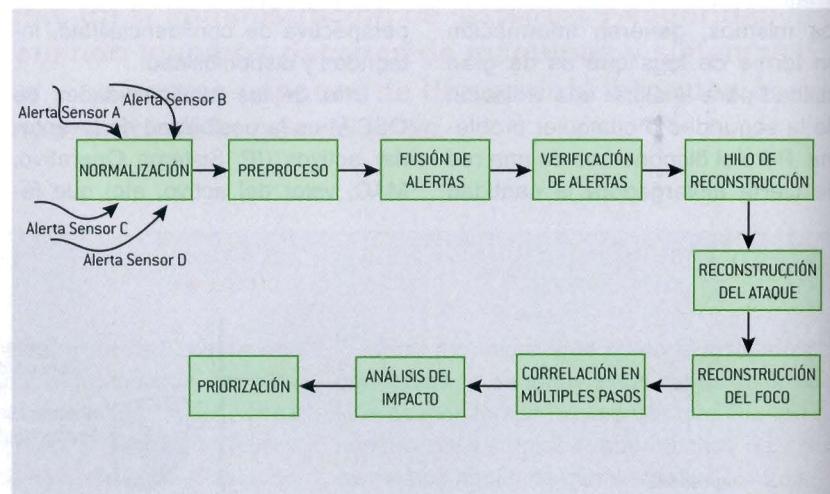
**La correlación de eventos**

La correlación se podría resumir como el análisis de los eventos

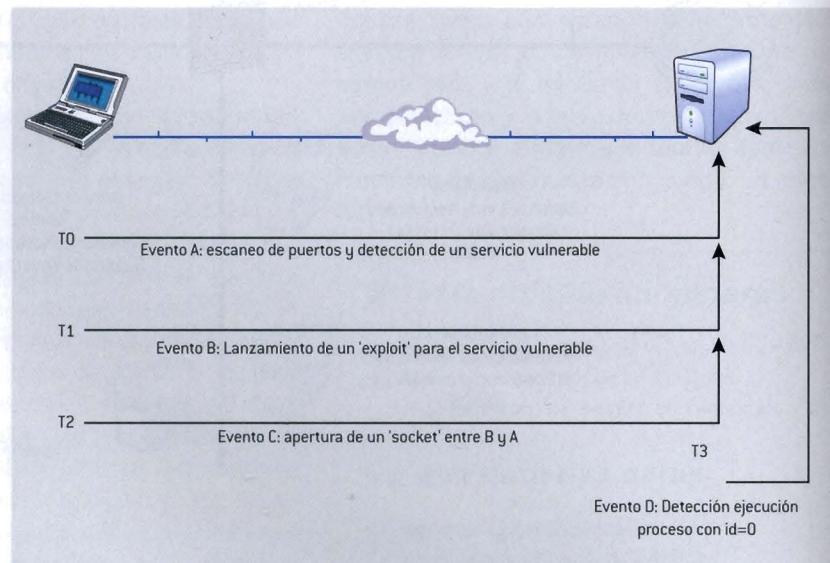
provenientes de distintos sensores o monitores con el propósito de reducir el número de falsas alertas, poder reconstruir el hilo de ejecución de un ataque, o ver el impacto real de un ataque sobre la red.

Para comprender en qué consiste la correlación y de qué manera se debe implementar, vamos a explicar de manera somera algunos de los pasos que en líneas generales debe seguir una herramienta

de correlación eficiente. El proceso de correlación que se explica a continuación está sacado de una publicación en el IEEE de varios investigadores de la Universidad de Santa Barbara (Fredrik Valeur, Giovanni Vigna, Christopher Kruegel, Richard A. Kemmerer) A *Comprehensive Approach to Intrusion Detection Alert Correlation* que se puede descargar de la web: [http://www.cs.ucsb.edu/~vigna/publications/2004\\_valeur\\_vigna\\_kruegel\\_kemmerer\\_TDSC\\_Correlation.pdf](http://www.cs.ucsb.edu/~vigna/publications/2004_valeur_vigna_kruegel_kemmerer_TDSC_Correlation.pdf). Para el lector interesado en profundizar en la correlación y su aplicación práctica en el campo de la seguridad informática, es muy recomendable la lectura del libro de los mismos autores:



**Figura 2. Etapas de correlación**



**Figura 3. Fases de un ataque**

### Intrusion Detection and Correlation: Challenges and Solutions.

Las etapas que se deben de seguir son las siguientes (se puede ver el gráfico que las resume en la Figura 2:

- Normalización: una vez se reciben las distintas alertas de los sensores hay que normalizarlas de acuerdo a un estándar o formato con el fin de que todos los elementos que componen el proceso puedan entenderlas. Esto es así ya que las alertas producidas por distintos sensores pueden estar codificadas de distinta manera. Al final de este proceso se obtendrán las alertas con sus respectivos atributos en un formato común.
  - Preproceso: esta fase se podría decir que es una depuración de la anterior, ya que lo que se pretende es llenar aquellos campos que algunos sensores dejan en blanco cuando lanzan alguna alerta.
  - Fusión de alertas: el objetivo de esta fase es combinar alertas de
- diferentes sensores (por ejemplo dos sensores de red distintos) pero que hacen referencia a un mismo ataque. En este punto juega un papel muy importante el factor tiempo junto con la información de la alerta. Un ejemplo práctico sería la fusión de un evento lanzado por un IDS como Snort y por otro IDS distinto (imaginemos el IDS Bro) pero que en realidad hacen referencia a la misma alerta.
- Verificación de la alerta: la función aquí es la de desechar las alertas que no sean alertas reales o bien que no hayan conseguido su objetivo (explotar una vulnerabilidad, reventar un servicio mediante un ataque DoS, etc).
- Reconstrucción del hilo de ataque: Es en esta fase don-

de se realiza un análisis de la sucesión de eventos que tienen como origen una máquina concreta hacia un host destino concreto. En este caso se considerarán las alertas que tienen el mismo origen y destino en un sensor dado, justo al contrario que ocurre en la fase de fusión donde.

Las alertas a tener en cuenta deben darse en sensores distintos.

- Reconstrucción de la sesión de ataque: es justo en este punto cuando se relacionan los ataques detectados en un host con los ataques de lo que se ha tenido constancia a través de la red. A priori no hay una relación directa entre qué IP o puerto puede estar relacionado con un proceso o fi-

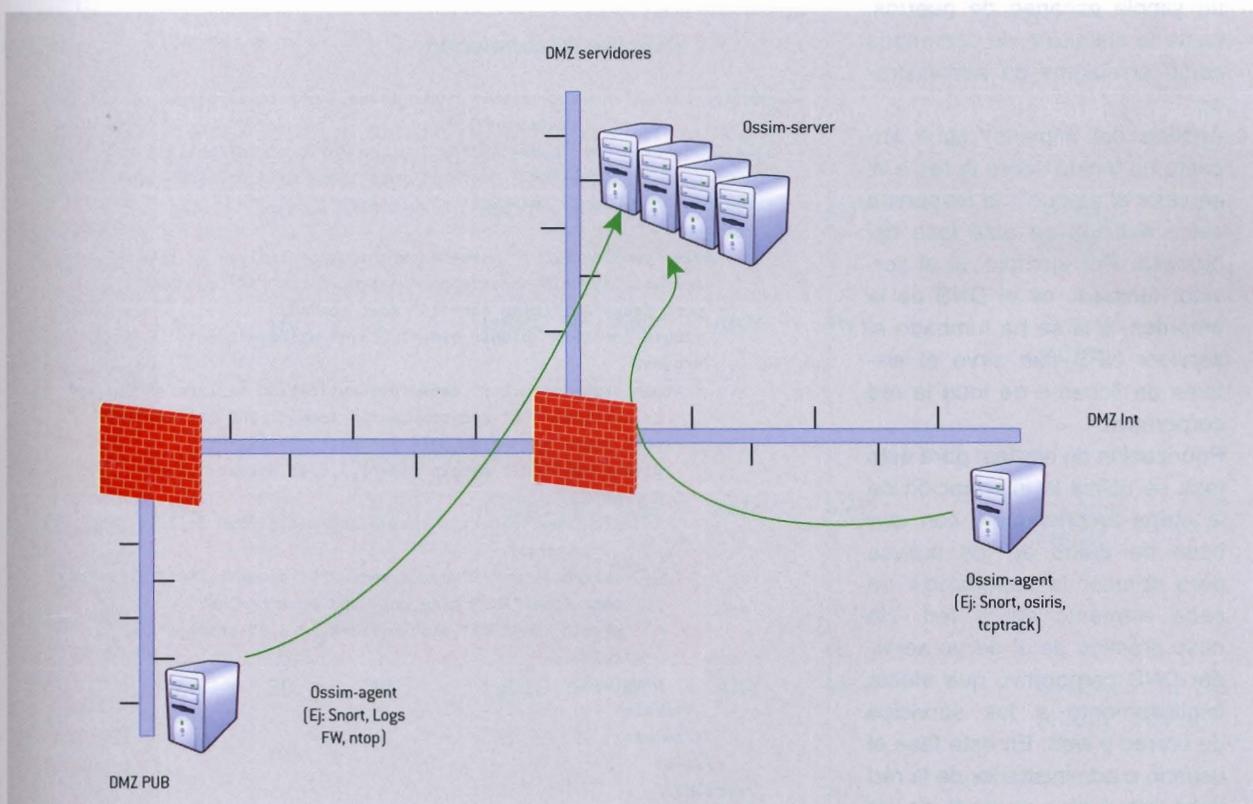


Figura 4. Sistema distribuido



chero de una máquina, por lo que la información que.

permite relacionar estos eventos es la magnitud *tiempo*. Por ejemplo, un proceso sospechoso lanzado en una máquina pocos segundos después de que un ataque haya sido detectado en la red, puede ser un ejemplo claro.

- Reconocimiento del foco del ataque: este elemento es únicamente válido para ataques de denegación distribuidos (DDoS) de varios equipos hacia uno concreto (many2one), o bien para ver si la máquina víctima de un ataque está siendo usada como pasarela para atacar otras máquinas (one2many). El factor a tener en cuenta es otra vez el tiempo, donde se define una ventana de tiempo que nos definirá un umbral, que si no se sobrepasa será indicio de alerta.
- Correlación multipaso: el objetivo es ver la evolución de un atacante, desde la primera fase de un simple escaneo de puertos, hasta la ejecución de comandos como privilegios de administrador.
- Análisis del impacto: ¿qué impacto ha tenido sobre la red o el servidor el ataque?, la respuesta viene definida en esta fase del proceso. Por ejemplo, si el servidor tumbado es el DNS de la empresa, o si se ha tumbado el servidor NFS que sirve el sistema de ficheros de toda la red corporativa.
- Priorización de alertas: para esta fase se utiliza la información de la etapa anterior junto con una base de datos de los activos para obtener la importancia de cada elemento en la red. Un caso práctico es el de un servidor DNS corporativo que afecta implícitamente a los servicios de correo y web. En esta fase el usuario o administrador de la red debe introducir a su modo de ver los valores de sus activos.

Aunque todas estas fases serían las ideales en un sistema de correlación, no tiene porque cumplirse de manera estricta ni por ese orden, si bien es cierto que muchas de las fases serán de obligado cumplimiento si se quiere tener correlación real.

OSSIM cumple varias de estas etapas mediante el mecanismo que implementa las reglas de correlación para relacionar en el tiempo eventos de distinta índole.

## OSSIM y sus funciones de correlación

Uno de los objetivos que se debe perseguir cuando existen diferentes puntos de información a analizar, y visto desde una perspectiva de correlación, es la centralización de la información en un único punto. Para conseguirlo es necesario la recolección de los logs en un punto central donde se pueda tratar a éstos y extraer los datos que realmente son interesantes. Para

realizar esta fase OSSIM se apoya en su agente, llamado ossim-agent, y que funciona como sonda a la que se envía los logs. Este agent analiza los logs y captura los datos importantes mediante parsers o traductores específicos para cada tipo de log, imaginemos que se están enviando los logs de acceso de un servidor Linux /var/log/auth.log, la información que podría ser importante sería el timestamp, el usuario de acceso y la IP origen del acceso. Cabe destacar que es posible disponer de tantos agentes como se quiera, tan solo es necesario asignar un identificador único (una IP) para diferenciar unos de otros, esto nos permite configurar un sistema de sensores totalmente distribuido y desplegarlo por tantas DMZs como se desee. Un ejemplo de una arquitectura distribuida searía el de la Figura 4.

Una vez se tiene la información relevante extraída, se procede a



Figura 5. Panel de OSSIM

### Listado 4. Directiva de correlación

```
<directive id="12" name="Possible brute force login attempt against DST_IP"
           priority="5">
  <rule type="detector" name="Authentication failure" reliability="3"
        occurrence="1" from="ANY" to="ANY" port_from="ANY" port_to="ANY"
        time_out="10" plugin_id="4002" plugin_sid="1,2,3,4">
    <rules>
      <rule type="detector" name="Authentication failure (3 times)"
            reliability="+1" occurrence="3" from="1:SRC_IP" to="ANY"
            port_from="ANY" time_out="15" port_to="ANY"
            plugin_id="4002" plugin_sid="1,2,3,4" sticky="true">
          <rules>
            <rule type="detector" name="Authentication failure (5 times)"
                  reliability="+2" occurrence="5" from="1:SRC_IP" to="ANY"
                  port_from="ANY" time_out="20" port_to="ANY"
                  plugin_id="4002" plugin_sid="1,2,3,4" sticky="true">
              <rules>
                <rule type="detector" name="Authentication failure (10
                  times)"
                  reliability="+2" occurrence="10" from="1:SRC_IP" to="ANY"
                  port_from="ANY" time_out="30" port_to="ANY"
                  plugin_id="4002" plugin_sid="1,2,3,4" sticky="true">
              </rule>
            </rules>
          </rule>
        </rules>
      </rule>
    </rules>
  </rule>
</rules>
```

|      |                      |              |                    |
|------|----------------------|--------------|--------------------|
| 1503 | <b>iptables</b>      | Detector (1) | Iptables           |
| 1504 | <b>fw1</b>           | Detector (1) | FW1                |
| 1506 | <b>realsecure</b>    | Detector (1) | Real Secure        |
| 1507 | <b>rrd_threshold</b> | Detector (1) | RRD Threshold      |
| 1508 | <b>rrd_anomaly</b>   | Detector (1) | RRD Anomaly        |
| 1509 | <b>threshold</b>     | Detector (1) | Threshold exceeded |

**Figura 6.** Ejemplo de algunas herramientas integradas

enviar a ésta a un servidor remoto (aunque se podría tener todo concentrado en un único servidor) donde se estudiará y analizará la información recibida. El proceso de OSSIM que se encarga de recibir, analizar, priorizar, correlar y almacenar la información es ossim-server

Ossim-server es un proceso complejo en cuanto a las tareas que realiza, ya que es el corazón de OSSIM.

La característica de OSSIM como correlador de secuencia de eventos, estriba en su potencia de combinar reglas que relacionan distintos elementos (dispositivos, aplicaciones, etc), direcciones IP, puertos, ventanas de tiempo, tal y como se ha descrito como precondition en

el modelo general de correlación. La idea de funcionamiento de este tipo de correlación se podría definir como la sucesión de una serie de eventos que concuerden con unos patrones definidos (ver Figura 3).

Un ejemplo sencillo sería: *si ocurre un evento A, y luego un evento B, seguido de un evento C, hacer la acción D*. Situándose en el contexto de un ataque y siguiendo la secuencia marcada en la figura 3 se puede hacer una analogía de los eventos de la siguiente manera: (Listado 1).

La idea general es relacionar eventos en un espacio de tiempo definido para cada regla, relacionando las direcciones IP origen y destino junto con sus puertos, y por

supuesto las firmas de los patrones que actúan o bien como detectores (IDSs, logs del sistemas, etc.) o como sensores (conexiones TCP abiertas, comando ejecutados con id=0, etc.).

OSSIM, como hemos comentado, dispone de cuadros de mandos y además de monitores de riesgo que miden el riesgo al que se está expuesto. Mediante unos algoritmos implementados en OSSIM, se efectúa la valoración del riesgo de manera proporcional al valor del activo (de ahí la importancia de la clasificación de los activos), la amenaza que puede representar un evento y la probabilidad de que ése evento se materialice.

También tiene un motor de anomalías, que complementa al sistema basado en patrones (la información de las firmas del IDS, la autenticación errónea de usuarios). La detección de anomalías puede detectar flujos de tráfico excesivos, cambios en el Sistema operativo, etc. Como cualquier sistema de detección ba-

**Tabla 1.** Directive 12 (Priority 5) Relación de eventos de autenticación fallidos

| Directive 12 (Priority 5) |                                    |              |           |             |          |     |            |          |        |                |            |  |
|---------------------------|------------------------------------|--------------|-----------|-------------|----------|-----|------------|----------|--------|----------------|------------|--|
|                           | Name                               | Relia-bility | Time _out | Occu-rrence | From     | To  | port_ from | port_ to | Sensor | Plugin ID      | Plugin SID |  |
| -                         | Authen-tication failure            | 3            | 10        | 1           | ANY      | ANY | ANY        | ANY      |        | Syslogd (4002) | 1 2 3 4    |  |
| -                         | Authen-tication failure (3 ti-mes) | 1            | 15        | 3           | 1:SRC_IP | ANY | ANY        | ANY      |        | Syslogd (4002) | 1 2 3 4    |  |
| -                         | Authen-tication failure (5 ti-mes) | 2            | 20        | 5           | 1:SRC_IP | ANY | ANY        | ANY      |        | Syslogd (4002) | 1 2 3 4    |  |
| +                         | Authen-tication failure (10 times) | 2            | 30        | 10          | 1:SRC_IP | ANY | ANY        | ANY      |        | Syslogd (4002) | 1 2 3 4    |  |



sado en anomalías, la cantidad de falsos positivos es elevada, por esa razón el motor de anomalías solo se toma como información complementaria.

Existe un tercer proceso en OSSIM, ossim-framework, que es la capa de presentación de la plataforma. Este proceso gestiona el interfaz donde se puede ver los eventos ocurridos, configurar la plataforma, inventariar los activos, realizar las auditorías con Nessus y un montón de operaciones adicionales.

OSSIM integra su propio gestor de incidentes, pero además, permite lanzar eventos (scripts, correos, etc) cuando se cumple una determinada condición. Es factible mandar un correo cuando se ha producido una cadena de eventos que dan fiabilidad a un incidente, lo que evita el tener que estar continuamente revisando la consola de eventos.

Otra funcionalidad de la correlación de OSSIM es la 'cross correlation' o correlación cruzada. OSSIM entre muchas de las herramientas con las que permite trabajar integra a al escáner de vulnerabilidades Nessus. Si se lanza Nessus sobre una serie de activos y detecta alguna vulnerabilidad en algún servicio, se almacena en la BBDD ese activo y la deficiencia que le acontece, si más tarde se detecta un ataque en la red hacia ese servicio vulnerable, automáticamente incrementará el nivel de riesgo y el posible nivel de compromiso. La idea de esta correlación es sencilla: si soy vulnerable a un determinado ataque y lo detecto nos encontramos ante un riesgo real.

### Ejemplo de algunas funcionalidades de OSSIM

Vamos a ver algunos ejemplos de herramientas integradas en OSSIM. Para los lectores interesados en ver de manera práctica como funciona OSSIM sin tener que instalar la plataforma, es factible descargar una versión funcional para VMware. Se puede descargar desde el enlace que se encuentra en la página del proyecto [www.ossim.net](http://www.ossim.net).

Esta versión de OSSIM integra algunas herramientas que son de gran utilidad, como es el caso de p0f, Arpwatch, Ntop, Nessus, Snort, Nagios y OSIRIS.

La Figura 5 muestra el menú de OSSIM a través del cual se puede configurar la herramienta y ver todos los eventos de nuestra red. OSSIM permite crear perfiles de usuarios distintos en función de cada rol con permisos de acceso a la plataforma diferentes en cada caso (Figura 5).

En el menú de configuración en la sección plugins se puede ver los plugins integrados en OSSIM y el identificador asignado. Este identificador es importante para poder discernir los eventos de los diferentes sensores y poder relacionar unos con otros en las reglas de correlación. La Figura 6 muestra los identificadores de algunos plugins como ejemplo.

Como hemos comentado OSSIM permite realizar un inventariado de los equipos, que se puede hacer manual o de manera automática con OSSIM, en nuestro caso de ejemplo podemos ver como se ha detectado de manera pasiva un sistema nuevo con IP 192.168.40.2 y que es un

sistema Windows. Esto se puede ver en los logs de la plataforma que están ubicados en `/var/log/ossim/`, así como en la sección de hosts en el menú de políticas (la Figura 7 muestra un ejemplo de varias máquinas – Listado 2).

Por otro lado, también detectamos nuevas direcciones MAC en nuestra red mediante la herramienta arpwatch y que se pueden visualizar en la consola de OSSIM tal y como muestra la Figura 8.

Lo que merece la pena analizar con más detenimiento son las reglas de correlación de eventos ya que en gran medida esto es una de las funcionalidades más potentes de OSSIM. Las reglas de correlación se configuran en el fichero XML ubicado en `/etc/ossim/server/generic.xml` y en el Listado 4 podemos observar una regla que hace referencia a un ataque por fuerza bruta.

Todas las directivas deben tener un identificador único, en el caso del ejemplo de la lista 4 es el número 12. Para identificar al sensor se hace referencia mediante el número de plugin (`plugin_id`) y para diferenciar eventos lanzados por un mismo sensor, se usa la etiqueta `plugin_sid`. (Una lista detallada de los eventos que lanza cada sensor se puede encontrar en el menú de configuración en la sección plugins, donde si se pincha en un plugin determinado se puede ver los eventos asociados a él – Listado 4).

Las etiquetas `from`, `to`, `port_from` y `port_to` sirven para localizar IPs y puertos (origen y destino). Por otro lado, para dar fiabilidad a las alertas, existe un parámetro, `reliability`, que permite asignar de manera manual la importancia de un evento determinado en función de otros eventos anteriores o bien de la importancia del mismo. Este parámetro es subjetivo en el sentido de que puede ponerse según cada caso y según el grado de anidamiento de la regla, por ejemplo un evento que ha disparado 5 veces seguidas (5 fallos en la introducción de la contraseña en un sistema), tendrá

|                |              |   |   |     |     |        |         |                |
|----------------|--------------|---|---|-----|-----|--------|---------|----------------|
| Servidor OSSIM | 192.168.40.2 | - | 4 | 300 | 300 | None   | None    | Servidor OSSIM |
| valario        | 192.168.40.1 | - | 3 | 300 | 300 | Server | vrossim | None           |
| Windows LabBox | 192.168.40.3 | - | 5 | 300 | 300 | None   | vrossim | None           |

Figura 7. Gestión de activos

| +[arpwatch] arpwatch: Mac address New |                               | 2007-08-14 16:04:21 | 192.168.40.2 |
|---------------------------------------|-------------------------------|---------------------|--------------|
| Plugin:                               | arpwatch (1512)               |                     |              |
| Plugin SID:                           | arpwatch: Mac address New (1) |                     |              |
| Userdata 1:                           | 00:0C:29:85:90:6B             |                     |              |
| Userdata 2:                           |                               |                     |              |
| Userdata 3:                           |                               |                     |              |

Figura 8. Detección de una MAC

más relevancia que el mismo evento ocurrido una sola vez.

Para relacionar eventos que hacen referencia a un estado anterior, supongamos una dirección IP, se debe de hacer con el número de regla y las etiquetas SRC\_IP (source IP), dst\_IP (destination IP). Para el caso que nos ocupa tenemos como ejemplo en la regla número 2 referencia a la IP de la regla 1, 1:SRC\_IP.

Por otro lado, como hemos comentado en el modelo general de correlación, es importante relacionar los eventos en una ventana de tiempo determinada, por ello, es fundamental poder configurar un tiempo de expiración de una determinada regla, y ésta es precisamente la función del parámetro timeout.

Teniendo una idea de cómo funcionan las directivas de correlación, pasemos a explicar que hace exactamente la directiva anterior.

En el paso uno comprueba si se lanza algún evento del plugin 4002 (pam\_unix de Linux) con el identificador 1, 2, 3 o 4. Estos identificadores hacen referencia a accesos erróneos (autenticación fallida) al sistema, bien porque se haya introducido un contraseña mal, bien porque se haya introducido un usuario incorrecto, o porque algún usuario del sistema haya introducido la contraseña mal al intentar hacerse con una consola root mediante el comando su. Las direcciones IPs implicadas en este caso nos dan igual, ya que puede ser cualquiera de los sistemas que estamos monitorizando. La fiabilidad de esta regla es de valor 3 sobre 10, es decir, no tiene un peso muy grande.

Pero si seguimos con el andamiento de las reglas y viendo que éste evento se repite, como en el caso de la segunda regla, la fiabilidad aumenta. En este caso porque

el mismo sensor ha lanzado otro evento en un periodo de tiempo menor de 15 segundos (estamos ya ante una fiabilidad de 4), en este caso sí se tiene en cuenta las direcciones IPs que son las implicadas en la fase número 1. Si se vuelve a dar otra vez el mismo evento lanzado por el mismo sensor, se aumentará la fiabilidad (+2) con un total de 6 y probablemente ya no estemos ante un error puntual y no se trate de un falso positivo. Para estar totalmente seguros de que estamos ante un ataque de fuerza bruta, creamos una cuarta fase de eventos donde si se detecta más eventos iguales en menos de 30 segundos, se aumentará la fiabilidad y podremos afirmar que estamos ante un ataque real.

Podemos ver el ejemplo de las reglas de correlación de manera gráfica a través del portal web de OSSIM, tal y como muestra la Tabla 9.

## Resumen

Aunque hemos visto un ejemplo sencillo de correlación para no alargar demasiado el artículo, la posibilidad de crear reglas más complejas donde intervengan sensores diferentes, eventos distintos, etc, hacen de OSSIM una herramienta muy potente a la hora de monitorizar una red.

Pero solo se han visto algunas de las propiedades de OSSIM, ya que la plataforma tiene muchas más funciones y posibles aplicaciones, desde el funcionamiento para la monitorización de la red (caída de sistemas, flujos de tráfico anómalos, etc), la violación de la seguridad, o como simple gestor de incidentes o herramienta de inventariado.

Para cualquier administrador de sistemas o de seguridad, la plataforma es de gran utilidad pues permite gestionar en un único punto todos los eventos que ocurrán en la red, evitando el tener que revisar cientos de logs con formatos distintos y que podrían limitar la visibilidad en algunos casos. ●

## Sobre el Autor

Angel Alonso Párrizas es Ingeniero en Informática e Ingeniero Técnico en Telecomunicación. Posee las certificaciones CISSP, CISM, GCFW, SSP-MPA, SSP-CNSA y CCNA. Actualmente trabaja como Ingeniero de Seguridad para la Autoritat de Certificació de la Comunitat Valencia <http://www.accv.es>, la PKI de la Generalitat Valenciana.

## En la red

### OSSIM:

- <http://www.ossim.net>

Propuesta de una arquitectura de sistemas de detección de intrusos con correlación

- <http://mural.uv.es/apan/documentos/aalonso-PFC.pdf>

A Comprehensive Approach to Intrusion Detection Alert Correlation'

- [http://www.cs.ucsb.edu/~vigna/publications/2004\\_valeur\\_vigna\\_kruegel\\_kemmerer\\_TDSC\\_Correlation.pdf](http://www.cs.ucsb.edu/~vigna/publications/2004_valeur_vigna_kruegel_kemmerer_TDSC_Correlation.pdf)

### Prelude IDS

- <http://www.prelude-ids.org/>

### OSSEC:

- <http://www.ossec.net/>

### Bro IDS:

- <http://www.bro-ids.org/>