Implantación de un sistema de detección de intrusos en un entorno universitario

Deployment of an Intrusion Detection System in a Universitary Environment

E. J. Mira, R. Montañana y F. J. Monserrat

Resumen

Este artículo describe una experiencia piloto en la que se implantó un IDS (sistema de detección de intrusos) en la Universidad de Valencia. Se probaron diferentes ubicaciones del sensor y diversas formas de conexión física. También se desarrollaron diversos programas para simplificar las labores de operación y mantenimiento diario del sistema. Se monitorizó y analizó el rendimiento del sistema durante períodos de carga de tráfico elevada. Por último se realizaron simulaciones de diversos tipos de ataques poniendo así a prueba la eficacia del sistema.

Palabras Clave: Sistemas de detección de intrusos (IDS).

Summary

This paper describes a pilot experience to deploy an IDS (Intrusion Detection System) at the University of Valencia. Different locations and physical connections of the probe were tested. Programs were also developed to assist in the daily operation and maintenance of the system. The system performance was monitored and analysed under heavy load conditions. Finally, attacks of several kinds were simulated in order to test the efficiency of the system.

Keywords: Intrusion Detection Systems

En general las universidades españolas carecen de mecanismos de seguridad que las protejan de los ataques que se producen en Internet. Debido al ambiente abierto que reina en las universidades está poco extendido el uso de cortafuegos

1.- Motivación y objetivos

La idea de instalar un IDS en la Universidad de Valencia surgió en octubre de 2001 de la mano de RedIRIS. Se quería lanzar una experiencia piloto basada en la implantación de un IDS en un nodo regional que tuviese un tráfico importante pero no excesivo. Aunque el IDS se instalaría en el nodo regional en principio la experiencia se limitaría a monitorizar el tráfico de la Universidad de Valencia, no todo el de la Comunidad Valenciana.

En general las universidades españolas carecen de mecanismos de seguridad que las protejan de los ataques que se producen en Internet. Debido al ambiente abierto que reina en ellas está poco extendido el uso de cortafuegos, que limiten el acceso por defecto y dejen abiertos sólo los servicios necesarios. En su lugar suele seguirse una política inversa de 'permitir por defecto' y filtrar sólo los servicios imprescindibles (correo y web, por ejemplo).

Una solución al problema de falta de seguridad que crea esta política 'aperturista' es la instalación de sistemas pasivos que alerten a los administradores en el momento en el que se produzca un ataque. El administrador será conocedor del ataque y dependiendo de la gravedad de éste podrá decidir si adopta medidas correctoras y si avisa al CERT asociado (en nuestro caso IRIS-CERT) y/o al responsable de la red origen del ataque.

Para facilitar esta tarea se pensó desarrollar un programa que automatizase el anuncio de alertas al CERT por medio del correo electrónico. El sistema de detección de intrusos se encargaría de elaborar cada día un informe con todas las alarmas producidas y enviarlo por correo electrónico al administrador de seguridad. El administrador respondería ese e-mail al sistema indicando qué alertas debían ser reenviadas al CERT y/o al responsable de la red origen del ataque. De esta forma el sistema se encargaría de enviar las alarmas seleccionadas a la dirección de correo adecuada.

En función de la fuente de información tenemos por una parte los IDSs basados en red y por otra los IDSs basados en host

RedIRIS proporcionó el hardware necesario, consistente en una máquina VA-Linux modelo 2230 FullOn. Para la conexión ATM se utilizó la tarjeta PCA-200E de FORE Systems.

Los objetivos planteados fueron los siguientes:

- Detectar los ataques que se reciban desde el exterior; objetivo fundamental de cualquier IDS.
- Detectar los ataques que se produzcan hacia el exterior. Esto puede deberse a usuarios malintencionados dentro de la propia universidad o, más frecuentemente, a usuarios externos que previamente han conseguido atacar una máquina de la universidad y que a continuación la aprovechan para lanzar desde ella nuevos ataques hacia afuera.
- Enviar incidencias de forma automatizada al IRIS-CERT. Se debía implementar un software para automatizar la gestión de incidencias, de forma que se pudiese avisar al CERT de la organización sobre los ataques producidos, categorizándolos de acuerdo con su nivel de importancia.
- Monitorizar la carga del sistema. Para que un IDS sea efectivo es fundamental que sea capaz de analizar todo el tráfico. Debido al análisis intensivo que se ha de realizar con cada paquetes cabe el riesgo de que en situaciones de tráfico elevado el IDS no sea capaz de 'digerirlo' en su totalidad, con lo que un ataque podría pasar desapercibido. El sistema debe disponer de herramientas que permitan detectar cuando esto ocurre.

2.- Breve introducción a los IDSs

En función de la fuente de información tenemos por una parte los IDSs basados en red, que detectan ataques capturando en modo promiscuo y analizando paquetes de una red, y por otra los IDSs basados en host, que operan sobre la información recogida desde dentro de una computadora, como pueden ser los ficheros de auditoría del sistema operativo.

En cuanto al tipo de análisis existen dos acercamientos: detección de abusos, que busca eventos que coincidan con un patrón de un ataque conocido (*firma*), y detección de anomalías, que se centra en identificar comportamientos inusuales en un host o una red.

Por último, según el tipo de respuesta, podemos encontrar IDSs que implementan respuesta pasiva y se limitan a notificar las intrusiones al responsable de seguridad e IDSs de respuesta activa que cuando detectan las intrusiones llevan a cabo automáticamente una serie de acciones preestablecidas, como incrementar el nivel de sensibilidad de los sensores o reconfigurar cortafuegos, por ejemplo.

3.- Snort

Snort es un IDS en tiempo real desarrollado por Martin Roesch y disponible bajo GPL. Se puede ejecutar en UNIX y Windows. Actualmente es el sistema de detección de intrusos más utilizado. La versión 1.9 dispone de unas 1.700 reglas y de multitud de aplicaciones para el análisis de sus alertas. Snort se basa en la librería *libpcap* para la captura de paquetes.

La arquitectura de Snort está formada por cuatro capas:

- **Decodificador de paquetes**: se encarga de tomar los paquetes que recoge la *libpcap* y almacenarlos en una estructura de datos en la que se apoyan el resto de capas.

- **Preprocesadores**: hacen un pretratado de los paquetes. Por ejemplo, el preprocesador *ip_frag* se encarga de reensamblar fragmentos IP y el *stream4* reconstruye el flujo TCP a partir de los segmentos almacenados en memoria.
- **Motor de detección**: implementa el algoritmo Boyer-Moore para la búsqueda de firmas. Este algoritmo es el más eficaz conocido para la búsqueda de un patrón en una cadena arbitrariamente larga. El problema es que en los IDSs no existe un único patrón sino varios. Se han diseñado ciertas mejoras añadiendo una estructura de datos Aho Corasick [6] lo cual teóricamente mejora el rendimiento hasta en un 500%.
- **Etapas de salida**: se utiliza cuando un ataque ha sido detectado. En este caso Snort dispone de plug-ins para el almacenamiento de la alerta en múltiples formatos: SQL (PostgreSQL, MySQL, Oracle, UnixOBDC), ASCII, XML, WinPopup, syslog, ...

El sensor debe ubicarse en el punto donde se quiera analizar el tráfico y donde la red lo permita

4.- Posibles ubicaciones del sensor

La localización de los sensores de Snort es una cuestión crucial que depende en gran medida de la infraestructura de la red en la que estemos trabajando. En el caso de la de la Universidad de Valencia podemos distinguir dos partes: la red interna y el acceso a Internet. Dependiendo del tráfico que queramos analizar nos centraremos más en una u otra.

La red interna está formada básicamente por las infraestructuras correspondientes a tres campus, más una serie de centros más pequeños. Los campus están interconectados mediante una red ATM privada formando una topología mallada que utiliza PNNI para mejorar la fiabilidad. Para el transporte de tráfico entre ellos se utiliza LAN Emulation, entre otras técnicas. Los centros medianos se conectan por líneas dedicadas de 512 Kb/s a 2 Mb/s y los pequeños por túneles VPN sobre conexiones ADSL. En cada campus/centro hay uno o varios routers que intercambian con el resto información mediante el protocolo EIGRP dentro de un sistema autónomo privado (el 65432). La UV posee la red 147.156.0.0/16 (clase B). Este espacio de direcciones se distribuye en más de cien subredes de diferentes tamaños.

Dentro de cada campus la red se basa en conmutadores Ethernet sobre los que se configuran múltiples VLANs para separar el tráfico. En cada campus hay un router que permite la interconexión local de las diferentes VLANs. La comunicación con el resto de la universidad y con Internet se realiza a través del conmutador ATM.

La conexión a Internet se realiza por medio de RedIRIS, cuyo POP (Point Of Presence) en la Comunidad Valenciana está situado en el edificio del Servicio de Informática de la Universidad de Valencia, ubicado en el campus de Burjassot. Dicha conexión se realiza mediante un PVC ATM entre el router principal de la Universidad y el de RedIRIS en el POP de la Comunidad Valenciana. Este PVC atraviesa dos conmutadores ATM, uno de la Universidad y otro de RedIRIS.

El sensor debe ubicarse en el punto donde se quiera analizar el tráfico y donde la red lo permita. A continuación describimos las posibles ubicaciones del sensor, comentando las ventajas e inconvenientes de cada una de ellas.

4.1.- PVC Multipunto

Como hemos visto antes el acceso de la Universidad de Valencia hacia el exterior se hace por medio de un PVC ATM entre el router principal y el de RedIRIS. Este PVC resulta ser un lugar idóneo donde poner el sensor para capturar una copia de todo el tráfico intercambiado con el exterior.

Los concentradores simplifican enormemente la monitorización de tráfico por su característica inherente de reenviar las tramas por todos sus puertos

Para ello hay que sustituir el PVC ATM punto a punto por uno multipunto. Dado que no es posible establecer un PVC multipunto bidireccional hay que desdoblar el PVC existente en dos, uno para cada sentido. Esto provoca algunos problemas con el tráfico multicast debido a la aparición de rutas asimétricas, como veremos más adelante.

4.2.- Hubs

Los concentradores simplifican enormemente la monitorización de tráfico por su característica inherente de reenviar las tramas por todos sus puertos.

El problema con los concentradores es que tan sólo se pueden utilizar para Ethernet 10/100Mbps y obligan al funcionamiento en modo half-dúplex. Sin embargo esta limitación de velocidad puede resultar incluso interesante puesto que nos asegura que la interfaz del IDS nunca se verá desbordada por el tráfico entrante, cosa que sí podría ocurrir en caso de enviarle tráfico de una conexión full-dúplex de la misma velocidad que la interfaz del IDS pues se sumaría el tráfico en cada sentido.

Si queremos analizar mediante un hub el tráfico en un enlace 'trunk' entre dos conmutadores el IDS deberá disponer de soporte para tramas etiquetadas según 802.1Q para que sea capaz de interpretar correctamente las tramas que recibe.

4.3.- Splitters de fibra

Otra opción es instalar splitters de fibra en las conexiones a monitorizar. En este caso la replicación se hace a nivel óptico, por lo que se requieren pocos cambios a nivel de la configuración de los equipos. Se puede aplicar tanto en conexiones ATM como Ethernet (half o full-dúplex).

En Ethernet half-dúplex el uso de un splitter es muy similar al hub antes descrito. En cambio en conexiones ATM o Ethernet full-dúplex es preciso utilizar un splitter y una interfaz física diferentes para cada sentido. Esto encarece y complica el sistema. Además el splitter supone un elemento adicional en la comunicación, con el consiguiente riesgo de avería. Por contra tiene la ventaja de que consigue una comunicación full-dúplex sin limitaciones.

5. Desarrollo del proyecto

5.1.- Proxy

La primera prueba del IDS se hizo configurando una sesión SPAN utilizando como puerto origen el del servidor proxy de la Universidad y como destino el del ordenador Linux donde se tenía instalado el Snort. La UV tiene cerrado el puerto 80 en el sentido de salida y obliga a todo el tráfico web a salir por el servidor proxy, por lo que este puerto representa una cantidad importante de tráfico.

El servidor proxy tiene dos interfaces, una hacia la Universidad y otra hacia el exterior. Para poner a prueba el rendimiento del sistema se eligió monitorizar la interfaz interna que es la que soporta un mayor volumen de tráfico.

A modo de ejemplo, para establecer una sesión SPAN con puerto origen 2/11 (servidor proxy) y destino 8/16 (IDS) en un conmutador Catalyst 5500, se introduce el siguiente comando en el conmutador:

```
set span 2/11 8/16
```

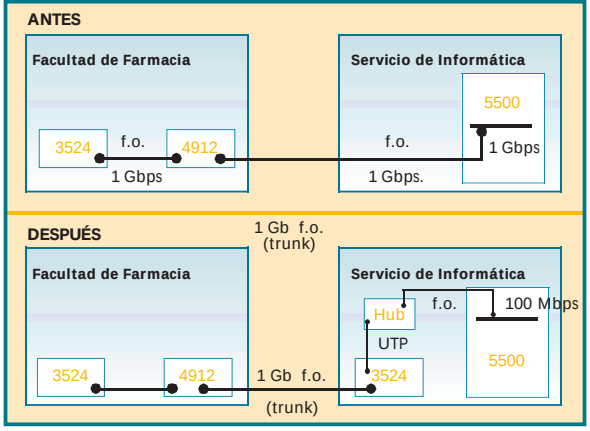
El puerto 8/16, por el que recoge datos el IDS, no necesita tener asignada ninguna dirección IP. Por otro lado tenemos el puerto 8/22, que corresponde a la interfaz de gestión del IDS; este puerto sí tiene asignada una dirección, pero no desempeña ningún papel en la función de monitorización del tráfico.

La solución adoptada para monitorizar todo el tráfico de entrada/salida de la Universidad fue la de establecer un PVC multipunto entre el router de RedIRIS y el de la Universidad

5.2.- Monitorización de VLANs

En diciembre se pasó a monitorizar una VLAN de la Facultad de Farmacia, cuya conexión con el Servicio de Informática se realizaba mediante un enlace Gigabit Ethernet. Ver la figura 1 (antes):

FIGURA 1: CAMBIOS REALIZADOS EN LA INFRAESTRUCTURA DE RED PARA LA MONITORIZACIÓN DE LA VLAN 32



Para poder establecer la sesión SPAN con el IDS hubo que convertir a 100 Mbps la conexión gigabit de la Facultad de Farmacia mediante un conmutador intermedio. Además se quería configurar dicha conexión en modo half-dúplex para asegurar que el tráfico en ambos sentidos no superara en ningún momento los 100 Mbps, capacidad máxima que podía aceptar la interfaz LAN del IDS. Dado que por otro lado era necesario realizar además una conversión de interfaz eléctrica (100BASE-TX) a óptica (100BASE-FX) se interpuso un hub que convertía de cobre a fibra y al mismo tiempo forzaba la conexión a modo half-dúplex. La configuración resultante puede verse en la figura 1 (después). Durante el período de prueba se comprobó que el tráfico total de este enlace era bastante inferior a 100 Mbps, por lo que estas modificaciones no supusieron en ningún momento una merma de rendimiento para los usuarios.

Una vez realizados los cambios descritos ya podemos establecer una sesión SPAN entre el puerto 6/3 (Facultad de Farmacia) y el 8/16 (IDS). Pero el puerto 6/3 es un enlace trunk y sólo nos interesa recibir en el IDS el tráfico perteneciente a una VLAN en particular, por lo que en este caso establecemos la sesión SPAN con un filtro de forma que el puerto 8/16 sólo reciba las tramas pertenecientes a la VLAN 32, que es la que nos interesa. Para ello utilizamos en el Catalyst el siguiente comando:

```
set span 6/3 8/16 filter 32
```

Otra opción podría haber sido la de conectar el IDS al conmutador mediante un hub con lo que no habría sido necesario configurar la sesión SPAN. Sin embargo esto habría sido más complicado ya que al tratarse de un enlace trunk el hub habría enviado las tramas con las etiquetas 802.1Q y el IDS tendría que disponer de soporte para este encapsulado y encargarse de filtrar los paquetes no pertenecientes a la VLAN 32.

5.3.- Monitorización en el PVC ATM

La solución adoptada para monitorizar todo el tráfico de entrada/salida de la Universidad fue la de establecer un PVC multipunto entre el router de RedIRIS y el de la Universidad. En un principio ambos routers eran Cisco; en abril de 2002 RedIRIS añadió a su POP un nuevo router principal de la marca

Como no es posible crear circuitos multipunto bidireccionales (full-dúplex) el PVC punto a punto entre ambos routers se ha de desdoblar en dos, uno para cada sentido

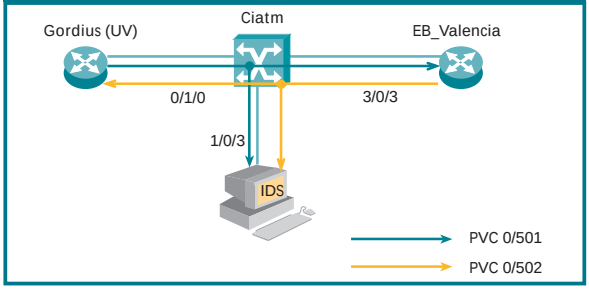
Juniper, lo cual requirió realizar algunas modificaciones en la configuración, como veremos a continuación.

5.3.1.- PVC multipunto entre routers Cisco

Como ya hemos comentamos no es posible crear circuitos multipunto bidireccionales (full-dúplex) por lo que el PVC punto a punto entre ambos routers se ha de desdoblar en dos, uno para cada sentido. La configuración resultante se muestra de manera esquemática en la figura 2.

La configuración de los dos PVCs multipunto en el conmutador ATM es una tarea sencilla. A modo de ejemplo la figura 3 muestra la configuración correspondiente al esquema de la figura 2. El conmutador utilizado es un Cisco LightStream 1010. Como puede verse la interfaz que envía las celdas se configura como raíz y las que las reciben se configuran como hojas.

FIGURA 2: ESQUEMA DE CONEXIÓN ENTRE LOS ROUTERS DE RedIRIS Y UV MEDIANTE UN PVC MULTIPUNTO



Una vez hemos configurado los PVCs en el conmutador pasamos a declararlos en los routers; esto se puede hacer utilizando para ambos la misma subinterfaz, o una diferente para cada uno. Dado que a cada una se le asocia una dirección IP diferente, si se utilizan dos subinterfaces aparecen rutas asimétricas, lo cual plantea problemas en algunos casos, en particular en los protocolos de routing multicast. En cambio si se emplea la misma subinterfaz para ambos PVCs la ruta es simétrica y además la configuración es más sencilla. Por tanto se optó por asociar ambos PVCs a la misma.

FIGURA 3: CONFIGURACIÓN DEL PVC MULTIPUNTO EN EL CONMUTADOR ATM

```
#int atm 0/1/0
  #atm pvc 0 502 cast-type p2mp-root int atm3/0/3 0 502
cast-type p2mp-leaf
  #atm pvc 0 502 cast-type p2mp-root int atm1/0/3 0 502
cast-type p2mp-leaf
#int atm 3/0/3
  #atm pvc 0 501 cast-type p2mp-root int atm0/1/0 0 501
cast-type p2mp-leaf
  #atm pvc 0 501 cast-type p2mp-root int atm1/0/3 0 501
cast-type p2mp-leaf
```

Vamos a describir brevemente la configuración que aparece en la figura 4 utilizando a modo de ejemplo el caso de *Gordius* (el de *EB_Valencia* es análogo). La subinterfaz ATM0.36 tiene asociada la dirección IP 130.206.211.182 (máscara de 30 bits) con un ancho de banda (comando `'bandwidth'`) de 45.000 Kb/s; este valor es el que se utilizará para los cálculos del protocolo de routing. Para asignar los PVCs a la subinterfaz utilizamos el comando `'atm pvc'`. Los circuitos utilizarán transporte AAL5 según el RFC 1483 únicamente para datagramas IP, sin soporte multiprotocolo (`'aal5mux ip'`). Asignamos a cada PVC un caudal SCR y PCR de 45000 Kb/s. Dado que para cada PVC ambos valores (SCR y PCR) coinciden se trata de un servicio CBR. Además coinciden con el previamente declarado en el comando bandwidth. La declaración map-list que aparece más abajo nos indica que el PVC 15 (el de VPI/VCI 0/502) se 'mapea' con la dirección IP 130.206.211.181, que es precisamente la 'compañera' de la de esta subinterfaz. Por tanto el PVC 0/502 se utilizará para enviar tráfico hacia ese destino y será el PVC saliente, mientras que cabe deducir que el PVC 0/501 se utilizará para el entrante (aunque para estar seguros de esto tendríamos que ver la configuración de *EB_Valencia*).

5.3.2.- PVC multipunto entre Juniper y Cisco

FIGURA 4:
CONFIGURACIÓN DE PVCs MULTIPUNTO EN *GORDIUS*

```
#int ATM0/0.6 multipoint
  #mtu 1500
  #bandwidth 45000
  #ip address 130.206.211.182 255.255.255.252
  #atm pvc 15 0 502 aal5mux ip 45000 45000 32
  #atm pvc 16 0 501 aal5mux ip 45000 45000 32
  #map-group ip-rediris
  #exit
#map-list ip-rediris
  #ip 130.206.211.181 atm-vc 15 broadcast
  . . .
#ip route 0.0.0.0 0.0.0.0 130.206.211.181
```

Como ya hemos comentado, en abril de 2002 RedIRIS añadió un nuevo router principal marca Juniper al POP de Valencia. A partir de entonces el PVC de la Universidad de Valencia terminaba en dicho router, en vez de en el Cisco EB-Valencia. La configuración utilizada en EB-Valencia no funcionaba en el Juniper. La diferencia fundamental era que el Juniper no permitía asociar más de un PVC con una misma subinterfaz. Para resolver el problema se creó en el Juniper una subinterfaz nueva asociada con el PVC entrante (el 0/502); dado que esta subinterfaz sólo se utilizaba para recibir tráfico y no era visible en la red se le asignó una dirección IP privada.

En abril de 2002 RedIRIS añadió un nuevo router principal marca Juniper al POP de Valencia. A partir de entonces el PVC de la Universidad de Valencia terminaba en dicho router, en vez de en el Cisco EB-Valencia

En principio la creación de la segunda subinterfaz resolvió el problema creado por el PVC multipunto, pero cuando en septiembre de 2002 se puso en marcha el servicio de routing multicast en RedIRIS la existencia de dos subinterfaces en el Juniper creaba rutas asimétricas que impedían el funcionamiento del protocolo PIM-SM debido a que fallaba el mecanismo conocido como RPF check (Reverse Path Forwarding check).

La solución a este problema fue la creación de un túnel que permitiera una ruta simétrica. Una vez creado el túnel se definirían rutas multicast en ambos routers para forzar que el tráfico fuera siempre por el túnel. Así se resolvía el problema del fallo en el RPF check y el PIM-SM volvía a funcionar. En el router de RedIRIS la ruta multicast era sencilla, pues se trataba de encaminar hacia el túnel los paquetes multicast de toda la red de la Universidad de Valencia (147.156.0.0/16). En la Universidad de Valencia lo lógico era declarar una ruta multicast por defecto hacia el túnel, pero esto tenía la desagradable consecuencia (lógica por otra parte) de enviar por el túnel todo el tráfico multicast que llegaba al router principal de UV, impidiendo por completo la distribución de tráfico hacia el interior de la Universidad. El tráfico multicast del exterior llegaba hasta el router de entrada pero nunca pasaba de él. Es decir, a efectos del tráfico multicast la información de routing EIGRP del sistema autónomo de UV estaba siendo ignorada. Esto se debe a que para este tráfico tienen preferencia las rutas multicast, aun cuando tengan una máscara de menor longitud.

Se barajaron varias soluciones posibles al problema, pero algunas de ellas resultaban difíciles o casi imposibles de mantener. Por ejemplo una era, en vez de utilizar la ruta por defecto, declarar en el router principal de la Universidad rutas estáticas que encaminaran por el túnel una a una todas las redes de Internet, a excepción de la 147.156.0.0/16. Pero aun utilizando profusamente agregación de rutas mediante CIDR la lista de rutas a declarar era enorme. Otra posibilidad habría sido dejar la ruta por defecto e incluir una multicast estática para cada subred de la Universidad, pero esto hubiera requerido añadir una nueva ruta cada vez que se crea una subred, y se habrían perdido –en lo que al tráfico multicast se refiere– las ventajas que comporta el routing dinámico.

La solución más sencilla y eficaz a este problema era jugar con las distancias administrativas. Por defecto una ruta estática (uni o multicast) tiene una distancia administrativa de 1, mientras que una aprendida por EIGRP la tiene de 90. Asignando a la ruta multicast por defecto que estamos creando una distancia administrativa superior (por ejemplo 200) la información de rutas dinámicas aprendida

Cuando la memoria reservada para reconstruir el flujo TCP de las conexiones se llena Snort elimina de memoria de forma masiva los nodos que no va a utilizar

por EIGRP prevalece siempre, tanto para tráfico unicast como multicast. De esta forma la distribución multicast dentro de la red de la universidad se realiza normalmente, ya que la ruta por defecto se utilizará solamente cuando EIGRP no tenga conocimiento de la red de destino. Además debemos asignar a la ruta unicast por defecto una distancia administrativa superior a la de la multicast (por ejemplo 201) pues de lo contrario el tráfico utilizaría esta ruta primero y no haría uso del túnel. Para mayor claridad mostramos a continuación la parte relevante de la configuración utilizada en el router principal (*Gordius*) de la Universidad de Valencia:

FIGURA 5:
CONFIGURACIÓN MULTICAST EN *GORDIUS*

```
#interface Tunnel7
  #description tunel multicast hacia RedIRIS
  #ip unnumbered Loopback0
  #ip pim sparse-mode
  #ip multicast boundary 10
  #ip sdr listen
  #tunnel source 147.156.148.113
  #tunnel destination 130.206.211.253
  . . .
#ip morute 0.0.0.0 0.0.0.0 Tunnel7         200
#ip route  0.0.0.0 0.0.0.0 130.206.211.181 201
  . . .
```

6.- Resultados

6.1.- Pruebas de rendimiento

En este apartado veremos los resultados al medir el rendimiento de Snort mediante la aplicación 'IDS Benchmark' que hemos desarrollado. Las pruebas se han realizado durante junio y julio de 2002 con el IDS monitorizando la red completa de la Universidad (circuito ATM multipunto) y se han utilizado dos configuraciones del sistema distintas. La primera ha sido ejecutar Snort de la forma más sencilla, mostrando la salida de las alertas y los logs en un fichero binario, y en la segunda configuración se le ha añadido la base de datos MySQL como back-end.

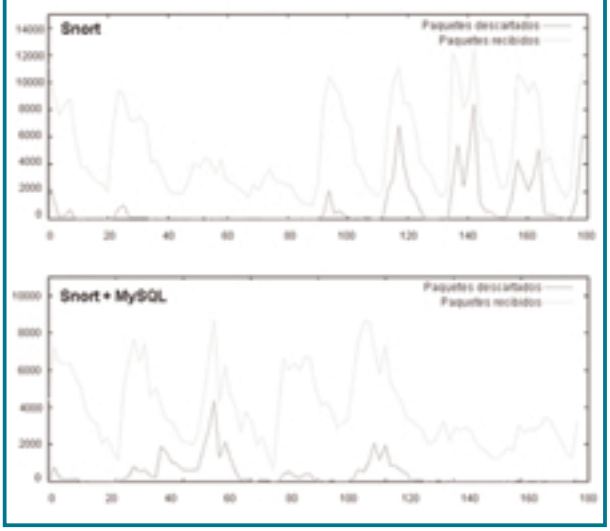
6.1.1.- Snort

En la primera prueba, realizada durante la tercera semana de junio de 2002, se ha lanzado el proceso Snort con el plugin tcpdump, que escribe en un fichero binario (flag '-b') las alertas y los logs producidos en formato tcpdump. Además hemos lanzado Snort con el flag '-A fast', lo que indica que las alertas deben ser lo más breves posibles. Según los diseñadores de Snort éste es el modo de funcionamiento que mejor rendimiento ofrece.

En la figura 6 podemos comparar el tráfico entrante con el procesado por Snort. En el eje de las abscisas tenemos el tiempo medido en horas, y en el de las ordenadas los paquetes por segundo. En la gráfica, que corresponde a una semana, pueden verse claramente los picos que corresponden a los días y los valles que indican las noches. Vemos también que por la noche no hay ningún exceso de carga, pero que durante el día Snort pierde tráfico, y esta pérdida no es regular, no podemos fijar un máximo de carga en pps para el sistema.

Por ejemplo, para las horas 100 y 120 (días 5 y 6) en las que entró un volumen de tráfico similar, Snort no tuvo el mismo comportamiento. Esto es así porque el comportamiento de un IDS es dinámico y depende del tipo de tráfico que entra y del conjunto de reglas. Como vimos en el punto anterior en Snort existen varios conjuntos de reglas y algunos producen más carga computacional que otros, en especial el conjunto *shellcode.rules*. Dependiendo del tipo y contenido del tráfico entrante, Snort tendrá más o menos trabajo. Es más, la carga no solo depende del tráfico externo, sino también del propio estado de Snort. Cuando la memoria reservada para reconstruir el flujo TCP de las conexiones se llena Snort elimina de memoria de forma masiva los nodos que no va a utilizar. Esta es una operación costosa y durante este tiempo puede producirse descarte de paquetes.

FIGURA 6: GRÁFICA COMPARATIVA ENTRE EL TRÁFICO ENTRANTE Y EL DESCARTADO EN SNORT



En teoría, al tener que convertir sus paquetes en inserciones a una base de datos SQL, Snort estará ocupado durante más tiempo en otras tareas que no son las de procesar paquetes, por lo que el descarte será mayor

6.1.2.- Snort+MySQL

En esta prueba, realizada durante la cuarta semana de junio de 2002, le añadimos a Snort una base de datos para almacenar las alertas que se producen. En teoría, al tener que convertir sus paquetes en inserciones a una base de datos SQL, Snort estará ocupado durante más tiempo en otras tareas que no son las de procesar paquetes, por lo que el descarte será mayor. En la figura 6 vemos la comparación entre tráfico entrante y descartado.

Si comparamos esta gráfica con la anterior parece que el número de paquetes descartados (es decir la diferencia entre los recibidos y procesados) al utilizar Snort+MySQL es menor que cuando se utilizaba Snort sólo, aunque en teoría tendría que ser mayor pues el IDS tiene que hacer más trabajo por cada paquete.

La explicación es la siguiente: durante la semana en la que se utilizó Snort+MySQL el tráfico fue inferior al de la semana anterior, en la que no se usó MySQL. Como puede verse durante la prueba de Snort el tráfico llegó a los 13,000 pps y en la prueba de Snort+MySQL no alcanzó los 9.000 pps, por lo que aunque la configuración Snort sea más rápida, al tener más tráfico que procesar es normal que descarte un porcentaje de paquetes mayor. Como el criterio de evaluación del rendimiento se basa en la carga y ésta no es igual en ambas pruebas, el cálculo del número medio de paquetes por segundo procesados nos puede dar una idea del rendimiento de cada configuración (ver tabla 1).

	Snort	Snort+MySQL
Media de PPS recibidos	5060.45	4014.29
Media de PPS procesados	4254.12	3586.01
Paquetes procesados (%)	84,1%	89,3%

Tabla 1: Estadísticas del rendimiento de Snort y Snort+MySQL

En el total de 4.560.476 alarmas recogidas durante junio de 2002 del PVC ATM, la mayoría fueron generadas por reglas que no son consideradas como ataques directos

6.2.- Simulación de ataques

La simulación de ataques se hizo utilizando la configuración de Snort con MySQL como back-end. Durante cuatro días se lanzaron cada cinco minutos tres ataques de exploit distintos. Los resultados de estas pruebas se muestran en la tabla 2. En esta tabla vemos los días en los que se lanzaron los ataques, los ataques realizados y las horas a las que tuvieron lugar los fallos de detección. En la columna de la derecha vemos el porcentaje de falsos negativos de un total de 1032 pruebas por exploit lanzado.

Vemos como es en jueves y viernes cuando se producen falsos negativos. Esto lo corroboran los resultados del apartado anterior, donde veíamos que los descartes de paquetes se producían solo en días laborables, nunca en fin de semana.

Ataque	Jueves	Viernes	Sábado	Domingo	Falsos negativos
Wu-ftp *exploit*	·	12,40h, 12,47h	·	·	0.2%
RPC.statd *exploit*	10,58h, 17,50h	8.56h	·	·	0.3%
Apache *DoS*	·	·	·	·	0.0%

Tabla 2: Resultados de la simulación de ataques

6.3.- Ranking de alertas

En el total de 4.560.476 alarmas recogidas durante junio de 2002 del PVC ATM, la mayoría fueron generadas por reglas que no son consideradas como ataques directos, por ejemplo pings, peticiones a servicios no muy frecuentes (finger), actividad de NetBIOS, actividad de software p2p, etc.

Del resto de alertas haremos dos distinciones:

- Alertas de tráfico sospechoso: son indicios de que algo está ocurriendo, ya sea porque el tráfico sea anormal o porque se acceda a servicios o ficheros que se consideran delicados.
  - accesos http a scripts utilizados para la puesta a punto de servidores web (whoisraw, test-cgi, win-c-sample.exe).
  - accesos externos a servicios delicados (mountd, nfsd).
  - respuestas a ataques (identificador de usuario root como consecuencia del comando 'id', listado de directorios por http).
  - tráfico TCP/IP corrupto (puerto TCP o UDP 0, fragmentos muy pequeños, ICMP anómalos).
  - fallos de login en telnet o FTP.
- Alertas de ataques: ataques claramente recogidos en las reglas de Snort, pero de naturaleza diversa.
  - ataques no intencionados por medio de gusanos.
  - recogida de información por escaneos.
  - ataques directos (ataques web, DoS y DDoS y Exploits).

Veamos en aproximadamente un mes cual ha sido el balance en cuanto a esta clasificación:

- Alertas de tráfico sospechoso: 89.96% (4.560.476 en total).

- Alertas de ataques:10.04% (458.177 en total).

El reparto de las alertas de ataques lo podemos ver en la tabla 3:

Tipo de ataque	Porcentaje	Total
Gusanos	92.62%	424.403
Escaneos	6.85%	31.402
Ataques web, DoS y DDoS, Exploits	0.53%	2.372

Tabla 3: Reparto de las alertas de ataques

Los ataques más peligrosos pueden ser quizá los exploits. Afortunadamente estos son los menos frecuentes en nuestro caso. Pero vemos que en aproximadamente un mes se han producido 190 alertas, lo que da unos 6 ataques diarios de este tipo.

Esta información puede ser utilizada por los administradores para evaluar el riesgo que tiene actualmente la Universidad, y elaborar así una política de seguridad completa, que podría ser más restrictiva cuanto mayor fuese el riesgo.

Como resultado del proyecto se ha implantado con éxito un IDS de altas prestaciones en un entorno universitario, lo que permite a los administradores conocer cuando está siendo atacada su red y les aporta información valiosa para determinar la naturaleza de los ataques

8.- Trabajo futuro

Como trabajo futuro para próximos proyectos podemos destacar lo siguiente:

- Desarrollo de una interfaz gráfica al sistema que permita:

  - realizar consultas a la base de datos por medio de los scripts de consulta desarrollados.
  - la manipulación de la base de datos para eliminar falsas alarmas y reducir así su tamaño. configurar Snort desde el propio interfaz.

- Estudio de la variación en la carga de Snort en función del tráfico entrante y del conjunto de firmas: nos puede ayudar a conocer con más exactitud el comportamiento dinámico que tiene Snort y adecuar su configuración a las necesidades de la organización.
- Desarrollo de un programa inteligente que active o desactive reglas en función de la carga: se podría hacer una clasificación del conjunto de reglas y dependiendo de la carga y del descarte de paquetes que se está produciendo, activarlas todas o sólo las más prioritarias.

9.- Conclusiones

Como resultado del proyecto hemos obtenido las siguientes conclusiones:

1.- Se ha implantado con éxito un IDS de altas prestaciones en un entorno universitario, lo cual permite a los administradores conocer cuando está siendo atacada su red y les aporta información valiosa para determinar la naturaleza de los ataques.
2.- El IDS necesita monitorización dada su imposibilidad de detectar cuando un ataque ha tenido éxito o no.

Snort es un software IDS de gran aceptación, potente y gratuito, multiplataforma y de código abierto. Su mayor inconveniente es que se encuentra en un desarrollo todavía temprano

3.- Cuando se monitorizan accesos de alta capacidad (>100 Mb/s) el rendimiento del hardware puede llegar a ser un cuello de botella en el análisis de reglas. Para ello se ha desarrollado una herramienta para medir el rendimiento de la máquina que permite a los administradores estimar la potencia de la máquina necesaria para desarrollar la función de IDS dependiendo de la cantidad de tráfico a monitorizar.
4.- Snort es un software IDS de gran aceptación, potente y gratuito, multiplataforma y de código abierto. Su mayor inconveniente es que se encuentra en un desarrollo todavía temprano y muchas de las características avanzadas de que disponen los IDSs comerciales no se encuentran todavía implementadas en Snort.

10.- Bibliografia

[1] S. Northcutt, D. McLachlan, J. Novak. "Network Intrusion Detection: An Analyst's Handbook (2nd Edition)". New Riders Publishing. Septiembre, 2000.
[2] M. Cooper, S. Northcutt, M. Fearnow, K. Frederick. "Intrusion Signatures and Analysis". New Riders Publishing. Enero, 2001.
[3] P. E. Proctor. "Practical Intrusion Detection Handbook". Prentice Hall. Agosto, 2000.
[4] T.H. Ptacek, T. N. Newsham. ``Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection''. Secure Networks, Inc. Enero 1998. http://secinf.net/info/ids/idspaper/idspaper.html
[5] SNORT -The Open Source Network Intrusion Detection System- http://www.snort.org
[6] M. Roesch. "Snort Presentation in The Black Hat Conference. Las Vegas'01", http://www.blackhat.com/presentations/bh-usa-01/MartyRoesch/bh-usa-01-Marty-Roesch.ppt
[7] M. Sobierey. "Michael Sobirey's Intrusion Detection Systems page". 2000. http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html
[8] J. Allen. "State of the Practice of Intrusion Detection Technologies". Carnegie Mellon University. 2000. http://www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html

Emilio José Mira Alfaro
(emial@alumni.uv.es)
Rogelio Montañana Pérez
(rogelio.montanan@uv.es)
Universidad de Valencia
Servicio de Informática
Francisco Jesús Monserrat Coll
(francisco.monserrat@rediris.es)
Equipo de seguridad IRIS-CERT
RedIRIS