Snort como herramienta administrativa

Andrés Acosta
Leonardo Rodríguez M.

Resumen
Snort como una herramienta de sniffer puede ser utilizado en forma ilegal capturando tráfico, en el cual se puede filtrar información privada de una empresa. Este artículo pretende dar a entender que su uso se puede orientar a una forma administrativa para asegurar una red, de tal forma que esta herramienta nos informe sobre hechos o eventos ocurridos dentro de unas reglas preestablecidas por el administrador de la red. Así mismo, describe algunas características de Snort, como su ubicación dentro del esquema de red y la creación de reglas, controlando así el tráfico entrante y saliente.
Snort como IDS busca aumentar la seguridad, vigilar y examinar el tráfico de la red en busca de datos sospechosos, además de detectar los primeros instantes de un ataque que pueden comprometer de manera importante la seguridad de la red.

Palabras clave
Snort, IDS (Sistema detector de intrusos), Sniffer.

Abstract
Alliances is one of the key factors for the organization's sustainable competitive advantage. The knowledge transference process, as one of the results of these strategic alliances, is configured as a knowledge interactive exchange between participants of the same alliances. In this article, the case study is the process developed by the Corporación Universitaria Minuto de Dios (UNIMINUTO) in order to transfer and become owner of the project CUPI2 of Universidad de los Andes – UNIANDES. By the present means, a contextualization management knowledge study will
take place, the characterization of the actors involved in the process and the study of the relation which is the origin of the transference process of the project CUPI2.

Key words
Knowledge Appropriation, CUPI2 Project, Knowledge transference, Knowledge, higher Education.

I. Introducción

En la actualidad, el control de redes se ha convertido en una tarea tediosa para sus administradores; a diario, surgen nuevos problemas al tratar de controlar el tráfico entrante y saliente de la información digital; dentro de las compañías se hace indispensable disponer de herramientas administrativas para generar procesos correctos, en el momento de hacer fluir información por canales digitales.

Sin embargo, para detener ciertos eventos que impiden el normal funcionamiento de estos sistemas es necesario conocer qué está ocurriendo dentro de un canal de datos. Función que cumple *snort*, además de ayudar al administrador a través de alertas creadas por defecto en el *software* o las configuradas por él mismo.

Este artículo busca dar a conocer la forma correcta de implementar snort y sus reglas, según lo desee el administrador; para permitir el flujo de paquetes bajo una extensión y una ruta específica, según las necesidades de la compañía.

II. Antecedentes

Para empezar a hablar de *snort* (Baker, A., Caswell, B., 2004), *sniffer* de paquetes y detector de intrusos basado en red, se debe saber y entender el funcionamiento de un IDS (Sistema de Detección de Intrusos), toda vez que esta herramienta nace de tales tipos de programas.

Un IDS es un tipo de *software* de gran utilidad para la auditoría y administración de redes, utilizado para detectar accesos no autorizados a una estación de trabajo o a una red, a través de un análisis detallado del tráfico de la misma, el cual es comparado con una base de datos de firmas o reglas de ataques conocidos, que impliquen cualquier tipo de actividad sospechosa o maliciosa sobre la red.

Tipos de IDS

HIDS (Host IDS): este actúa de forma local, es decir, sobre un único host con el objetivo principal de protegerlo, recopilando información de ficheros, recursos o logs, para posteriormente analizarlos y encontrar posibles hechos que atenten contra las firmas. Este tipo de IDS ofrece una gran ventaja y es que en el procesamiento es mucho menor comparado con un NIDS.

NIDS (Net IDS): este tipo de IDS, actúa en una red de igual manera que HIDS capturando todo el tráfico de una red de igual manera que lo puede hacer un *sniffer*, para luego analizar los paquetes capturados y así detectar primeras fases de posibles ataques a las redes. Por ejemplo, ataques de denegación de servicios, escanear puertos o intentos de entrar en un equipo de cómputo, analizando el tráfico de la red en tiempo real.

Arquitectura de un IDS: aunque no es un estándar generado por alguna entidad certificadora, la arquitectura de IDS se compone de los siguientes parámetros:

Recolección de datos: a través de los logs de registro de los dispositivos de red se pueden recopilar datos.

Parámetros: configuración de las reglas que determinan acciones particulares de amenazas o fallas de seguridad en la red.

Filtros: comparan datos obtenidos en la parte de recolección de datos con los parámetros.

Detector de eventos: función del IDS para alertar al administrador sobre actos inusuales en el tráfico de la red.

Dispositivo generador de alarmas: según la configuración que el administrador le proporcione al IDS, este está en capacidad de alertar mediante correo electrónico o vía sms.

Dónde colocar un IDS: lo más normal es colocar este dispositivo de acuerdo con las necesidades de la red. Por ejemplo, en el caso de un control total de la red, es conveniente colocar un IDS en cada tramo de red o simplemente instalarlo en un dispositivo, por donde pasará todo el tráfico de la red. Pero, esto entra en contradicción porque capturaría todo el trafico entrante y saliente, lo cual generaría un gran número de falsos positivos. La otra posibilidad es instalarlo detrás del cortafuegos o recurrir a la opción más fácil, esto es, ubicar uno delante y otro detrás del cortafuegos.

Snort como herramienta administrativa

Como ya se ha dicho, *snort* es considerado como un IDS y por ser un sistema detector de intrusos, contiene un motor de búsqueda y análisis basado en firmas que obtiene de eventos previamente analizados en logs, que genera alarmas de posibles amenazas obtenidas al escanear el tráfico de la red.

Dentro de *snort*, se pueden encontrar todos los ítems relacionados en la sección anterior como componentes de la arquitectura de un IDS.

Por su funcionalidad, *Snort* se ha convertido en una herramienta de fácil adquisición, toda vez que se encuentra funcionando bajo licencia GPL, y además se

ejecuta de forma correcta en plataformas Windows, UNIX /Linux.

Por otra parte, contiene una serie de patrones y reglas predefinidas a ataques conocidos y vulnerabilidades frecuentes, como lo pueden ser: implantación de backdoor, ataques de denegación de servicios (DoS), finger, ftp, ataques web, escaneos de puertos TCP y UDP (Nmap), entre otros.

Así mismo, provee en su sito web (http://www.*snort*.org) actualizaciones de las firmas de ataques encontrados por los administradores de red. Debido a su fácil configuración y administración, snort está en capacidad de dar sobre aviso al administrador si los equipos que se encuentran dentro de su red están siendo victimas de un sencillo *ping*, hasta el más complejo de los ataques; todo esto a través de firmas.

Configuración de snort

El primer paso de la configuración es la instalación, aunque es un procedimiento relativamente sencillo; cabe aclarar que para ejecutar *snort* en forma correcta es necesario que contenga una librería de bajo nivel para el acceso de redes; esta herramienta se llama *WinCap*, que en la actualidad se encuentra en su versión 3.1.

Una vez se ha realizado con éxito la instalación, es importante definir qué tipo de sistema se quiere monitorear. Puede ser una red o simplemente un *host* (computador de trabajo). En este caso se utiliza la configuración y los ejemplos basados en el análisis de información obtenida del escaneo de un simple *host*.

Por tal razón, es que en el momento de la configuración es de gran importancia modificar una de las variables de configuración, en la que se especifica qué tipo de sistema se quiere analizar, porque el *snort,* por defecto, trae la siguiente variable:

Var HOME_NET any

Especifica que se monitorea más de un *host* dentro de la una red. Ahora, se debe modificar esa línea con el objetivo de especificar una única máquina o una red clase C o, en su defecto, una parte de la red; es decir, solo varios equipos que pueden ser los más vulnerables.

1. Una red clase C: **var HOME_NET 192.168.2.0/24**

2. Host específico: **var HOME_NET 192.168.2.3/32**

3. Varios Host: **var HOME_NET 192.168.2.2/32, 192.168.2.4/32, 192.168.2.8/32, 192.168.2.9/32, 192.168.2.10/32**

Para el ejemplo del presente artículo se activará la herramienta para el análisis de un solo host, con base en la siguiente variable:

var HOME_NET ip de la máquina a análizar/32

Creación de reglas con *snort*

En este punto es importante aclarar el tipo de reglas que se pueden encontrar para configurar el *snort*.

1. *Subscribes:* este tipo de reglas son de ac ceso exclusivo para usuarios reconocidos y son registradas y avaladas por Snort. org.

2. *Registered:* son reglas para acceso general, pero de igual manera están respaldadas por la misma entidad.

3. *Unregistered:* son las reglas que snort establece o carga por defecto, en el momento de su instalación.

El lenguaje usado por snort es flexible y potente, basado en una serie de normas que sirven de guía para la escritura de las reglas.

Reglas

Las reglas snort (Security focus complete snort-based IDS architecture, s.f.) se pueden dividir en dos secciones lógicas: cabecera de la regla y opciones.

1. La cabecera contiene la acción de la regla en sí, protocolo, IPs, máscaras de red, puertos origen y destino del paquete o dirección de la operación.

2. La sección opciones contiene los mensajes y la información necesaria para la decisión a tomar por parte de la alerta en forma de opciones.

En resumen:

|| CABECERA – Acción – Protocolos involucrados – Direcciones IP – Números de puerto – Dirección de la operación || OPCIONES – Mensaje – Opciones de decisión ||

Parámetros más usados dentro de la línea de comando

Tanto para *snort*, como para otros aplicativos, el uso de los mismos puede ser administrable a través de un entorno gráfico o por medio de una línea de comando.

A continuación, se enumerarán algunos de los parámetros más usados y necesarios al momento de utilizar *snort*, por medio de una consola DOS:

1. *snort -W*: lista las interfaces de red disponibles sobre las cuales se va a llevar a cabo el estudio.

2. *snort –v*: este parámetro activa el modo verbal; sin este no se visualizaría en texto claro para el humano.

3. *snort –i*: sirve para seleccionar cualquiera de las interfaces listadas con el parámetro –W, para iniciar el proceso.

4. *snort –d*: sirve para visualizar cabeceras que contienen datos que pasan por las diferentes interfaces de red.

5. *snort –e*: muestra paso por paso y en detalle los análisis realizados.

6. *snort –c*: archivos dentro de los cuales snort contiene su configuración.

7. *snort* –l: para especificar en qué ruta quedarán los logs y las alarmas generadas durante el proceso del monitoreo.

Configuración de reglas (Redes-linux.es, 2008)
Para el uso de las reglas que por defecto se cargan en el sistema para generar alertas, snort debe tener dentro de sus archivos de configuración un fichero. rules, dentro del cual almacenará reglas generadas por el administrador, descargadas de Internet, las cuales han sido facilitadas por otros administradores y generadas por defecto.

Si se cuenta con otro archivo que contenga reglas, llamado, por ejemplo, "reglas.rules", a través del siguiente comando se puede hacer el llamado al archivo, para que interactúe en forma simultánea con el archivo original, creado por defecto por snort.

El comando es:

Include $RULE_PATH/reglas.rules
Así mismo, es posible generar varios archivos con reglas que podrían ser clasificadas de acuerdo con el tipo y la prioridad que el administrador le de a las mismas:

config <directiva>:nombre, descripción, prioridad

Otras consideraciones importantes, al modificar la configuración son:

1. Cabecera: las cabeceras en este tipo de configuración siguen los siguientes parámetros.

<acción><protocolo><ip origen y máscara de subred><puerto> <ip destino><puerto>

2. Opciones: en las opciones existe un sinfín de ellas, en este artículo solo se nombraran las mas conocidas:

a) msg. Es un tipo mensaje que contiene las firmas de los ataques.

b) itype. Para tráfico tipo ICMP.

c) Icode. Para código ICMP.

d) Flags. Flag TCP (A, S, F, U, R Y P).

e) Sid. ID de la sigla.

f) Classtype. Estas se encuentran en el archivo snort.confg.

g) Content. Busca patrones según el tráfico de la red y las reglas ya creadas.

Con lo anterior, es posible concluir que según el parámetro seleccionado, el administrador puede crear reglas a su antojo, para que el sistema le alerte de posibles ataques y de las firmas que ellas llevan según el daño que quiera hacer.

Ejemplos de reglas por defecto
Las reglas *snort* se ubican en ficheros reglas.rules.

A continuación, se presentará parte del contenido de un fichero, el cual generalmente contienen firmas, acciones que de forma casi inequívoca identifican un ataque, lo cual puede contener archivos ejecutables, mensajes, segmentos de códigos, puertos para atacar, entre otros.

alert tcp any 110 - > any any (msg:"Virus – SnowWhite Trojan Incoming"; content: "Suddlendtly"; sid:720; classtype:misc-activity;rev3;)

alert tcp any 110 - > any any (msg:"Virus – Possible pif Worm"; content: ".pif"; nocase; sid:721; classtype:misc-activity;rev3;)

alert tcp any 110 - > any any (msg:"Virus – Possible NAVIDAD Worm"; content: "NAVIDAD.EXE"; nocase; sid:722; classtype:misc-activity;rev3;)

alert tcp any 110 - > any any (msg:"Virus – Possible MyRomeo Worm"; content: "myjuliet.chm"; nocase; sid:724; classtype:misc-activity;rev3;)

alert tcp any 110 - > any any (msg:"Virus – Possible MyRomeo Worm"; content: "I Love You"; sid:726; classtype:misc-activity;rev3;)

alert tcp any 110 - > any any (msg:"Virus – Possible MyRomeo Worm"; content: "ble bla"; nocase; sid:725; classtype:misc-activity;rev3;)

alert tcp any 110 - > any any (msg:"Virus – Possible MyRomeo Worm"; content: "Sorry… Hey you!"; sid:727; classtype:misc-activity;rev3;)

alert tcp any 110 - > any any (msg:"Virus – Possible MyRomeo Worm"; content: "my picture from shakebeer"; sid:728; classtype:misc-activity;rev3;)

alert tcp any 110 - > any any (msg:"Virus – Possible pif Worm"; content: ".pif; nocase; sid:721; classtype:misc-activity;rev3;)

Estos ficheros.rules se almacenan en el directorio raíz de Snort (por defecto).

Los anteriores son ejemplos de tipos de alertas, los cuales son creados por snort, por defecto, y contienen reglas, la mayoría de ellos firmas reconocidas por antivirus y contienen la misma estructura.

III. Conclusiones

- Tal como se trató de evidenciar en el anterior documento, la decisión de instalar snort en una red puede realizarse teniendo en cuenta el tráfico que se quiere vigilar: paquetes que entran, paquetes que salen, dentro de un cortafuegos, fuera del cortafuegos; y en realidad su ubicación depende de los criterios de los administradores.

- Los IDS son una herramienta muy importante en la prevención de ataques, constituyen una primera barrera que nos puede ayudar a corregir fallas de seguridad o a recopilar información relacionada con un posible futuro ataque.

- En este artículo se ha definido snort como una herramienta capaz de crear alertas, de acuerdo con las necesidades que el administrador contemple dentro de la organización. Esta es una de las herramientas más poderosas en el campo de la administración de alertas de ataques.

IV. Referencias

[1] Baker, A., Caswell, B., (2004). POOR, *Baker*. Snort 2.1 Intrusion Detection. State Unites. 721h, segunda edición.

[2] Snort.org, (2008). *Snort Documents*. Recuperado el 15 de marzo de 2008, de http://www.snort.org/docs/

[3] *Security focus complete snort-based IDS architecture* (s.f.). Recuperado el 20 de marzo de 2008, de htto://www.securityfocus.com/infocus/1640

[4] Redes-linux.es. *Redes-Linux documentación*, Manuales. Recuperado el 20 de marzo de 2008, de http://beta.redes-linux.com/manuales/seguridad/snort_as s!.pdf

Andrés Acosta: *Diplomado de Seguridad Informática, I, 2008, Corporación Universitaria Minuto de Dios, (Uniminuto). Estudiante de VI semestre de Tecnología en redes de computadores y seguridad informática, Corporación Universitaria Minuto de Dios, Uniminuto.*
jimfloyd89@gmail.com

Leonardo Rodríguez *Martín. Diplomado de Seguridad Informática, I, 2008, Corporación Universitaria Minuto de Dios, (Uniminuto). Cursa cuarto semestre de Licenciatura en Electrónica en la Universidad Pedagógica Nacional; "Redes de Computadores y Seguridad Informática", Universidad Minuto de Dios.*
leonardorod79@gmail.com