SNORT

Detector de intrusiones ligero para redes.

Curso de Administración de Sistemas y Servicios Telemáticos
Martín de la Herrán Brickmanne
DNI 50879032-L

SNORT

Detector de intrusiones ligero para redes.

SNORT

Detector de intrusiones ligero para redes.

Resumiendo lo más posible la descripción, SNORT es un programa multiplataforma diseñado para analizar el tráfico que circula a través de una red, compararlo con una serie de patrones predeterminados, y generar las alarmas correspondientes.

1.- Introducción

Desde la aparición de internet hasta nuestros días, las redes informáticas han crecido a una velocidad tremenda, y junto con su crecimiento han aparecido multitud de "crackers" con propósitos no siempre benignos. Para defenderse de dichos ataques, que en principio asumiremos externos, hay ante todo que identificarlos. En este trabajo vamos a centrarnos en ataques provienientes de una red IP, y que por lo tanto cumplen forzosamente una condición instrínseca: todos ellos son detectables desde cualquier ordenador que esté conectado a la entrada de la red que queremos proteger.

Podemos hacer un cierto paralelismo entre los distintos programas de detección de intrusiones, como el SNORT, y agentes de aduana en un aeropuerto. Los programas de detección de intrusiones, Network Intrusion Detection Systems (NIDS en adelante), examinan todos y cada uno de los paquetes que reciben los distintos interfaces de red de una máquina. Una vez adquirido, el paquete es analizado para ver si el tráfico recibido cumple alguna de las reglas establecidas por el administrador. En tal caso, el programa se encarga de realizar alguna acción, como anotar el tráfico, hacer una copia local de los paquetes recibidos o hacer sonar un alarma.

2.- Utilidad

SNORT es pues un programa capaz de cumplir las siguientes funciones:

A. Escuchar en modo promiscuo en un interfaz de red, es decir, atendiendo tanto a los paquetes dirigidos a dicho interfaz como a los que no le correspondería escuchar, y mostrar o registrar localmente en el disco duro dicho tráfico. En este modo de operación se asemeja al programa "tcpdump" y es de hecho muy similar a él, hasta el punto de que comparte el sistema de registro en el disco de tráfico.

B. Escuchar en modo promiscuo en un interfaz de red, pero atender a una serie de reglas más avanzadas que las de tcpdump, lo cual puede servir para analizar y depurar tráfico y protocolos de red.

C. Como herramienta para el análisis avanzado del tráfico de red, obviamente también en modo promiscuo, y utilizando una base de reglas sofisticada para detectar y avisar sobre tráfico no deseado que esté circulando por nuestra red.

SNORT no se limita exclusivamente al protocolo IP+TCP/UDP/ICMP, aunque sobresalga su especialización en ellos. También es capaz de escuchar otros protocolos como ethernet, fddi, arp, rarp, decnet, lat, sca, moprc y mopdl.
Para ser capaz de filtrar y analizar el tráfico lo mejor posible, snort lleva una serie de módulos que le permiten:

- recomponer el tráfico fragmentado
- recomponer las cadenas de caracteres enviadas a través de servicios como telnet, ftp, smtp, http, pop3 e imap, lo que le permite explorar el tráfico desde el nivel de aplicación y no sólo del nivel de red.
- entender el tráfico http, como p.ej. la expansión de caracteres " " = "%20"
- entender el tráfico rpc
- reconocer el tráfico generado por el BackOrifice, independientemente de los puertos que use
- detectar escaneos remotos en busca de puertos abiertos

Además, está diseñado de modo que sea fácil incorporarle nuevos módulos para expandir su funcionalidad.

3.- Entorno

Snort es un programa escrito bajo licencia Gnu GPL, por lo que es libremente distribuible. Al estar escrito en C y con todo el código fuente liberado, funciona en múltiples plataformas como linux, windows, *BSD, Solaris, MkLinux, HP-UX, MacOS X y BeOS. Esto le diferencia de los sistemas integrados comerciales, cuyo precio normalmente es prohibitivo, y que requieren de hardware especializado. Snort funciona bien en cualquier PC doméstico, siempre y cuando no se le exijan velocidades de proceso demasiado elevadas para la CPU de la máquina.

Así pues, snort se puede instalar en casi cualquier ordenador u ordenadores de nuestra red, afectando muy poco al funcionamiento de ésta, mientras que permite detectar fácilmente el tráfico sospechoso. Su pequeño tamaño, licencia abierta y código multiplataforma le convierten en una herramienta útil para todos los administradores de red. Esa es la razón por la que se le da el calificativo de ligero (lightweight), en comparación los grandes NIDS comerciales.

4-. Uso personal práctico

Mi experiencia personal con snort se sitúa en el ámbito del firewall que he puesto a la salida del router ADSL de mi casa, antes de los dos ordenadores familares. Utiliza el sistema operativo OpenBSD 2.9, y he estado probando snort en varias ocasiones. Pese a estar filtrando un gran número de puertos con mi firewall (y que por lo tanto no llegan al snort, pues "ipf", el programa de firewall del OpenBSD 2.9 precede a snort, estas son algunas de las alarmas que ha generado el snort en mi ADSL, basándome en las reglas de www.whitehats.com:

- Intento de overflow: IDS488/misc_identd-overflow_SuSE

[**] [1:0:0] IDS488/misc_identd-overflow_SuSE [**]
[Classification: system integrity attempt] [Priority: 11]
08/24-08:02:15.657493 62.81.156.181:35782 -> 192.168.0.253:113
TCP TTL:58 TOS:0x0 ID:49551 IpLen:20 DgmLen:1500 DF
A* Seq: 0x3A5EE176 Ack: 0xBDD66DFC Win: 0x16D0 TcpLen: 32
TCP Options (3) => NOP NOP TS: 58071660 1930206606
[Xref => http://www.whitehats.com/info/IDS488]

- Tests de existencia de agujeros remotos como el BackOrifice:

[**] IDS397/trojan_trojan-BackOrifice1-scan [**]
08/02-23:07:10.547650 63.197.54.27:2517 -> 192.168.1.70:31337
UDP TTL:110 TOS:0x0 ID:8344 IpLen:20 DgmLen:47
Len: 27

[**] IDS397/trojan_trojan-BackOrifice1-scan [**]
07/28-14:00:31.343145 62.2.142.36:4406 -> 192.168.1.70:31337
UDP TTL:111 TOS:0x0 ID:13682 IpLen:20 DgmLen:47
Len: 27

- Autentificaciones fallidas a mi ftp:

[**] IDS364/ftp_ftp-bad-login [**]
06/22-19:15:29.378104 212.109.196.81:21 -> 192.168.0.253:40930
TCP TTL:40 TOS:0x10 ID:59645 IpLen:20 DgmLen:62 DF
AP Seq: 0xA60E75BB Ack: 0x60176F55 Win: 0x4470 TcpLen: 20

- Multitud de intentos de conexión a samba, tanto españoles

(este host pertenece a la red de telefónica)
[**] [1:0:0] IDS177/netbios_netbios-name-query [**]
[Classification: information gathering attempt] [Priority: 8]

09/01-17:00:52.741894 213.96.141.35:1263 -> 192.168.0.253:137
UDP TTL:116 TOS:0x0 ID:3643 IpLen:20 DgmLen:78
Len: 58
[Xref => http://www.whitehats.com/info/IDS177]

como extranjeros

cs666824-67.austin.rr.com
[**] [1:0:0] IDS177/netbios_netbios-name-query [**]
[Classification: information gathering attempt] [Priority: 8]
08/24-02:51:44.795386 66.68.24.67:1164 -> 192.168.0.253:137
UDP TTL:109 TOS:0x0 ID:13442 IpLen:20 DgmLen:78
Len: 58
[Xref => http://www.whitehats.com/info/IDS177]

pc132.cpool6.quickclic.net
[**] IDS177/netbios_netbios-name-query [**]
07/31-05:50:02.093897 24.244.201.132:62520 -> 192.168.1.70:137
UDP TTL:105 TOS:0x0 ID:27529 IpLen:20 DgmLen:78
Len: 58

- Comandos de "traceroute" con ipopts (hecho por mi mismo desde la universidad)

[**] IDS238/scan_Traceroute IPOPTS [**]
06/28-15:05:49.727722 138.100.17.15 -> 192.168.0.253
ICMP TTL:244 TOS:0x0 ID:31944 IpLen:60 DgmLen:124
IP Options (2) => RR EOL
Type:0 Code:0 ID:53795 Seq:2 ECHO REPLY

- Tráfico poco común:

[**] [1:0:0] IDS7/misc_SourcePortTraffic-53-tcp [**]
[Classification: suspicious miscellaneous traffic] [Priority: 1]
08/23-07:51:51.967810 210.183.235.129:53 -> 192.168.0.253:53
TCP TTL:241 TOS:0x0 ID:19942 IpLen:20 DgmLen:40
******S* Seq: 0x41CA6D73 Ack: 0x4B3AE2CE Win: 0x28 TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS7]

Y bastantes más. En principio ninguno de los ataques ha sido fructífero, y muchos de los ataques detectables no han llegado hasta el snort puesto que normalmente tenía mi firewall activado filtrando una gran candidad de puertos. Igualmente cabe notar que sólo he tenido activado el snort en periodos cortos.

Hay que tener en cuenta las falsas alarmas, según lo específico que sea el patrón de tráfico que hayamos detectado. Cuando se utilizan puertos altos aleatorios, como usan muchos protocolos a la hora de transmitir datos (ftp, dcc, programas Peer2Peer, etc),

existe la posibilidad de hacer saltar falsas alarmas. Así pues una conexión desde el puerto 80 de una máquina 31337 no siempre es indicativa de la existencia del BackOrifice, pero tráfico del puerto 2517 al 31337 sí resulta más sospechoso. Por otro lado, una conexión al demonio de identd con parámetros específicos (como el primer intento de intrusión de la lista que he pueso más arriba) sí indica claramente un intento de aprovecharse de una vulnerabilidad del sistema (intento inútil en mi caso).

A la hora de generar alarmas, he utilizado un programa externo a snort para generar un informe en formato html sobre el tráfico sospechoso encontrado por snort. El programa en concreto que he utilizado es “snortnarf”, un programa gratuito escrito en perl que se puede descargar de la web de snort. Existen multitu de programas de este tipo, todos ellos bien referenciados en www.snort.org

Dicho informe, en formato html y comprimido con zip, va adjunto a la copia electrónica de esta práctica.

5.- Ejemplos de reglas de snort

Las reglas de snort se pueden crear a mano o descargarlas de internet. Para casos específicos se pueden forjar a medida, pero en el caso de querer utilizar snort para la detección general de intrusiones, existen básicamente dos sitios en internet donde descargar ficheros muy completos detallando tráfico indeseable.
- www.snort.org, la página oficial del programa, que incluye un conjunto de reglas bastante completo
- www.whitehats.com, página web dedicada a noticias sobre hacking y herramientas gratuitas para la seguridad, tiene una base de datos bastante grande sobre los distintos patrones de tráfico utilizados en herramientas NIDS. De ahí se pueden descargar reglas actualizadas tanto para snort como para otros programas similares.

El tipo de tráfico detectable con estas reglas incluye:

- Intentos de aprovecharse de desbordamientos de pila de los distintos demonios que proveen servicios y pueden comprometer a una máquina.
- Escaneos de puertos en busca de protocolos abiertos. No sólo podemos configurar la sensibilidad de la alarma, sino que avisa siempre de intentos de conexión llamados "stealth", en los que no se llega a completar la conexión TCP pero si que queda revelado si el puerto está abierto o cerrado.
- Abusos de CGI de páginas web, como los famosos codered y nimda que han estado circulando recientemente por la web.
- Intentos de conexion al protocolo samba de windows, en busca de recursos abiertos, impresoras, contraseñas, y toda la parafernalia conocida de los generalmente inseguros windows y su protocolo de compartición de ficheros.
- Análisis de respuestas a diferentes patrones de tráfico, en busca de identificar el sistema operativo de una máquina. Con esto los "crackers" intentan saber que

sistema operativo usa una máquina para poder refinar posteriormente sus ataques. Varias herramientas, como "nmap" utilizan estas técnicas, y snort es capaz de detectar la mayoría de los intentos de identificación de sistema operativo.

- y un largo etcétera.

Algunos ejemplos de reglas para el snort, que no explicaremos en detalle pero son bastante claras, son:

- Conexión al ftp de usuarios con cualquier nombre, como "warez", muy utilizado en el entorno del pirateo de programas:

alert TCP $EXTERNAL any -> $INTERNAL 21 (msg: "IDS327/ftp_ftp-user-warez"; flags: A+; content: "user warez"; nocase; classtype: syst em-attempt; reference: arachnids,327;)

- Intentos de aprovecharse de una de las multiples vulnerabilidades del IIS, en concreto la de "directory traversal", donde el IIS permite a conexiones remotas moverse por todo el arbol de ficheros del servidor ya que no coprueba (comprobaba si se le ha aplicado el parche) correctamente las cadenas unicode:

  (la cadena es "..|25|c1|25|1c", que equivale a "../" transformado; el contenido entre "||" indica que es hexadecimal)

alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS432/web-iis_http-iis-unicode-traversal"; flags: A+; uricontent: "..|25|c1|25|1c"; nocase; classtype: system-attempt; reference: arachnids,432;)

- conexiones varias a samba:

alert UDP $EXTERNAL any -> $INTERNAL 137 (msg: "IDS177/netbios_netbios-name-query"; content: "CKAAAAAAAAAAAAAAAAAAAAAAAAAAAA|00 00 |"; classtype: info-attempt; reference: arachnids,177;)

6.- Instalación

Snort se instala muy fácilmente en los distintos ssitemas operativos unix. En Debian, no hay mas que buscar el paquete "snort"

apt-get install snort

Aunque debido a la búsqueda de estabilidad de Debian, puede compensar bajarse las últimas fuentes del programa en su página web.

Otras versiones de unix, también incluyen paquetes preforjados, como los RPM de Redhat, y paquetes en FreeBSD y OpenBSD. Aparte, compilar el programa es muy sencillo, ya que sus fuentes no ocupan más que unos pocos cientos de Kb. Snort suele instalarse en /usr/bin o /usr/local/bin. Incluye lógicamente sus documentos de "man" y algunos ficheros asociados.

En su modo de funcionamiento normal, snort se ejecuta como:

/usr/local/bin/snort -A full -c /etc/snort.conf -i eth0

en este caso apunta toda la información posible sobre los paquetes que haya encontrado, en base a las reglas encontradas en /etc/snort.conf, y escucha en el interfaz eth0.

Una vez arrancado, snort anota los paquetes sospechosos que ha encontrado de diferentes formas:

a) Normalmente genera ficheros en el directorio /var/log/snort. El fichero "alert" incluye los distintos patrones de tráfico encontrados. El fichero "portscan" incluye información sobre los intentos de escaneo de puertos encontrado, y, si se le ha dicho a snort que guarde los paquetes sospechosos encontrados (con p.ej. -A full) crea un árbol de directorios según la numeración ip de los paquetes de origen.
b) También puede generar alarmas via syslog.
c) Enviar la info a través de sockets unix
d) Enviarla a máquinas windows vía mensajes Winpopup y samba
e) Guardar la info en bases de datos (Mysql/PostgreSQL/Oracle/ODBC) o en formato XML.

Todos estos distintos modos de avisar del tráfico encontrado permite a los administradores de sistema actuar en consecuencia con el modelo de seguridad de su red.

A la hora de personalizar el fichero snort.conf, que normalmente nos habremos bajado o de snort.org o de whitehats.com, configuramos distintos parámetros que son mayoritariamente autoexplicativos, como, en mi caso:

Variables para utilizar en las reglas siguientes

var INTERNAL [192.168.0.0/16] < mi red interna
var EXTERNAL !$INTERNAL < la red externa
varDNS_SERVERS
[62.81.0.1/32,62.81.16.197/32,212.49.128.65/32,62.81.0.33/32,62.81.16.129/32]
servidores dns

Configuración de los distintos módulos:

preprocessor defrag
habilita el reensamblado de paquetes fragmentados

preprocessor stream2: timeout 23, ports 21 23 25 80 110 143, maxbytes 16384
reconstruye cadenas de caracteres a nivel de aplicación en los puertos indicados. Aquí se pueden añadir todos los puertos que sepamos utilizan tráfico de ese tipo.

preprocessor telnet_decode
reconstruye los comandos telnet utilizados

preprocessor http_decode: 80 2301
indica en que puertos hay tráfico web; aquí añadiríamos todos los puertos donde tuviesemos sevidores web.

preprocessor rpc_decode: 111
localización y descodificación del protocolo rpc

preprocessor bo: -nobrute
si queremos analizar _todo_ el tráfico en busca de tráfico proveniente de backorifice quitamos esta opción. Sin embargo, comprobar todo el tráfico de este modo es muy costoso en cuanto a CPU y solo es recomendable en el caso de que tengamos sospechas de la existencia de servidores ocultos.

preprocessor portscan: $INTERNAL 5 5 portscan
sensibilidad del detector de escaneo de puertos, en número de conexiones en cuantos segundos.

preprocessor portscan-ignorehosts: $DNS_SERVERS
que ordenadores puede ignorar el portscan por ser especialmente ruidosos y generar muchas conexiones rápidas. Es normal incluir los servidores de DNS ya que suelen originar muchas conexiones/segundo, y podemos añadir más máquinas.

Posteriormente ya solo queda arrancar el snort como indicado más arriba, y tratar los datos que produce en /var/log/snort

7.- Uso como capturador de paquetes

Snort se puede también utilizar para capturar paquetes, de modo parecido a tcpdump. El funcionamiento es casi idéntico a dicho programa y queda fuera del ámbito de éste trabajo, pero podemos citar algun ejemplo de utilización:

snort -i eth0 -b -l logdir

Copiará toda la información que escuche el interfaz eth0 en formato binario-tcpdump a un fichero llamado snort-“dia”@”hora”.log del directorio “logdir”. Posteriormente se puede analizar el tráfico con el mismo snort, o con otras herramientas como *ethereal*, *tcpshow*, o similares. Un sitio web muy interesante donde han hecho uso de snort de este modo es http://project.honeynet.org/ , donde se ofrece a los visitantes web practicar y

analizar ataques externos, todos ellos capturados normalmente en formato tcpdump. Snort puede procesar la información (con la opción –r “fichero tcpdump”) y mostrar el nivel de aplicación decodificado (opción –d) .

Igualmente se puede filtrar selectivamente que tráfico capturamos:

snort -i eth0 -b -l logdir host 192.168.1.70

sólo capturará la información procedente o destinada a la ip 192.168.1.70. El filtrado de información de este modo es muy versátil y comparte la sintaxis de tcpdump, pudiendo determinar puertos, protocolos, direcciones de origen y/o destino, etc.

8.- Para más información y referencias:

- www.snort.org

Web del programa, ejemplos, reglas e infinidad de documentación. Destacan los siguientes documentos, que cubren snort mucho más en profundidad:
http://www.snort.org/docs/writing_rules/
http://www.snort.org/docs/lisapaper.txt
http://www.snort.org/docs/idspaper/

- www.whitehats.com

Sitio web dedicado al hacking, seguridad, y programas gratuitos sobre seguridad y redes. Hospeda una gran base de datos sobre huellas digitales (cadenas) sospechosas e indicadoras de intrusión.

- www.tcpdump.org

Página sobre el programa tcpdump y su librería, libpcap, que permite capturar los paquetes que circulen a través de un interfaz de red.

- www.ethereal.com

Programa para el análisis de tráfico capturado en un fichero

- www.tcpshow.org

Programa para analizar la información a nivel aplicación decodificada de tráfico tcp capturado en ficheros tipo tcpdump

Martín de la Herrán Brickmanne