

Snort

Introducción a sistemas IDS

Un IDS (*Intrusion Detection System*) es una herramienta de seguridad que intenta detectar o monitorizar los eventos ocurridos en un determinado sistema informático o red en busca de intentos de accesos que comprometan la seguridad de los sistemas. Estos accesos pueden ser ataques por Hackers, denegaciones de servicios, finger, ataques web, CGI, escaneos de puertos, nmap, etc.. Los IDS buscan patrones predefinidos característicos de actividades sospechosas en la red o host. Tienen la capacidad de analizar los paquetes de la red en busca de actividad sospechosa, por lo que esto aumenta la capacidad de prevención y alerta anticipada dado que el ataque es detectado al inicio del mismo.

Existen varios tipos de IDS:

- HIDS (*Host IDS*). Protege a un único pc o host. Funcionan de forma local analizando la actividad del sistema en archivos de logs, ficheros, recursos, buscando posibles anomalías. La interfaz funciona en modo no promiscuo y la carga del procesador es mucho menor que los otros IDS. Monitorizan gran cantidad de eventos analizando actividades con gran precisión determinando de esta manera que procesos y usuarios se involucran en una determinada acción.
- NIDS (*Net IDS*). Protege a la red. Captura y analiza paquetes de red como un sniffer de red. Posteriormente analiza los paquetes capturados buscando patrones que supongan algún ataque. La interfaz de red funciona en modo promiscuo para poder analizar todo el tráfico de red y este análisis se lleva a cabo en tiempo real. Además de trabajar a nivel de TCP/IP lo hacen también a nivel de aplicación.
- DIDS (*Distributed IDS*). Es un sistema basado en la estructura de cliente-servidor compuesto por una serie de NIDS que actúan como sensores centralizando la información de posibles ataques en una unidad central que puede almacenar o recuperar los datos en una base de datos centralizada. Esto trae una ventaja que es que cada NIDS se puede fijar en las reglas de control

especializándose para cada segmento de red.

Existe otro tipo de clasificación basado en el tipo de respuesta:

- IDS Pasivo: el sensor detecta una posible intrusión, almacena la información y envía una señal de alerta que se almacena en la base de datos. No actúan sobre el atacante.
- IDS Reactivo: responde a la actividad sospechosa reprogramando el cortafuegos para que bloquee el tráfico que proviene de la red del atacante. Puede actuar cerrando la conexión o mediante una respuesta predefinida por el administrador de red. SNORT puede funcionar tanto como un sistema reactivo como pasivo.

Un IDS está formado por :

- a) La fuente de recogida de datos. Estas fuentes pueden ser un log, dispositivo de red, o como en el caso de los HIDS, el propio sistema.
- b) Reglas que contienen los datos y patrones para detectar anomalías de seguridad en el sistema.
- c) Detectores de eventos anormales en el tráfico de red.
- d) Filtros que comparan los datos recogidos de la red o de logs con los patrones almacenados en las reglas.
- e) Dispositivo generador de informes y alarmas.

Una de las claves más importantes para el correcto uso de los IDS es en donde vamos a colocar los IDS. Es posible que coloquemos el IDS antes del Firewall para estar seguros de capturar todo el tráfico de red, pero estaremos generando una cantidad alta de falsas alarmas. Las alarmas generadas por el IDS no tienen necesariamente que pasar por el firewall. Para estar seguros de no generar gran cantidad de falsas alarmas es conveniente colocarlo detrás del firewall, así de esa forma solo monitoriza el tráfico entrante que el firewall permitió pasar y la posibilidad de falsas alarmas es muy inferior. Algunos no recomiendan instalarlo en el mismo firewall ya que es posible que se escape del análisis parte del tráfico de red. Si la red es demasiado grande, se podría instalar un IDS en cada

host.

Características de Snort

Snort es una herramienta de detección de intrusiones open source que puede usarse para monitorear redes TCP/IP y detectar una amplia variedad de tráfico sospechoso así como ataques externos. Es utilizado para detectar una gran cantidad de ataques como ser buffer overflows, escaneo de puertos, ataques CGI, denegaciones de servicios, etc. Puede funcionar como sniffer (al estilo tcpdump), como detector de intrusos monitorizando todo un dominio de colisión, y como packet logger registrando los paquetes que circulan por la red. Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida. Provee al administrador de una gran cantidad de datos para tomar la decisión correcta al momento que se lleva a cabo la actividad sospechosa. Es un detector liviano de intrusiones ya que puede ser instalado en los host sin interrumpir la actividad normal de los equipos. Implementa un lenguaje de creación de reglas flexible y sencillo. Cuando un paquete coincide con algún patrón establecido en las reglas de configuración se guarda en un log. De esa forma se sabe cuando, donde y como se produjo el ataque. Los logs pueden ser almacenados en archivos de texto como en bases de datos (MySQL, Postgresql, ORACLE, ODBC), en ASCII o XML. Snort utiliza la librería libcap y tcpdump como registros de paquetes.

Snort está bajo licencia GPL y funciona tanto en Linux como en Windows. Además de disponer de una gran cantidad de filtros predefinidos, cuenta con actualizaciones constantes ante casos de ataque, barridos o vulnerabilidades que vayan siendo detectadas a través de los distintos boletines de seguridad. La base de datos de ataques se actualiza constantemente y se puede añadir o actualizar vía internet. Un usuario puede crear “Firmas” de ataques y enviarlos a la lista de Snort así todos los usuarios pueden beneficiarse.

En las últimas versiones se ha implementado una característica nueva denominada FlexRep. Esto permite que dada una conexión que emita tráfico malicioso se bloquee con un DROP mediante el envío de un paquete con el flag RST activa. No solo corta las conexiones sino que también se puede

personalizar el tipo de respuesta.

Instalación y uso de Snort

La instalación se puede realizar desde los repositorios de la distribución. Para el caso de Ubuntu (Debian) es posible instalarlo con apt-get desde la consola:

```
pablo@laptop:~$ sudo apt-get install snort
```

Una vez instalado los paquetes editamos el archivo de configuración /etc/snort/snort.conf. Este archivo viene bastante comentado por lo que es muy facil realizar modificaciones. En primera instancia no sería necesario realizar modificaciones ya que los parametros estandar se aplican a cualquier red mediana. Podemos afinar algunas características acordes a nuestra red como ser:

```
var HOME_NET <red>
```

```
var EXTERNAL_NET !$HOME_NET
```

```
output log_tcpdump: <nombre de archivo>
```

y otras con las cuales podemos afinar nuestro sensor. Dentro de este archivo tambien estan los include de las reglas, las cuales se almacenan en /etc/snort/rules

```
pablo@neurona:/etc/snort/rules$ ls -l
total 1248
-rw-r--r-- 1 root root 4709 2006-11-09 00:39 attack-responses.rules
-rw-r--r-- 1 root root 17026 2006-11-09 00:39 backdoor.rules
-rw-r--r-- 1 root root 3002 2006-11-09 00:39 bad-traffic.rules
-rw-r--r-- 1 root root 7212 2006-11-09 00:39 chat.rules
-rw-r--r-- 1 root root 6786 2006-11-09 00:39 ddos.rules
-rw-r--r-- 1 root root 63449 2006-11-09 00:39 deleted.rules
-rw-r--r-- 1 root root 5867 2006-11-09 00:39 dns.rules
-rw-r--r-- 1 root root 5290 2006-11-09 00:39 dos.rules
-rw-r--r-- 1 root root 471 2006-11-09 00:39 experimental.rules
-rw-r--r-- 1 root root 26548 2006-11-09 00:39 exploit.rules
-rw-r--r-- 1 root root 3353 2006-11-09 00:39 finger.rules
-rw-r--r-- 1 root root 20491 2006-11-09 00:39 ftp.rules
-rw-r--r-- 1 root root 15618 2006-11-09 00:39 icmp-info.rules
-rw-r--r-- 1 root root 4488 2006-11-09 00:39 icmp.rules
-rw-r--r-- 1 root root 12579 2006-11-09 00:39 imap.rules
-rw-r--r-- 1 root root 2430 2006-11-09 00:39 info.rules
-rw-r--r-- 1 root root 199 2006-11-09 00:39 local.rules
```

```

-rw-r--r-- 1 root root 17159 2006-11-09 00:39 misc.rules
-rw-r--r-- 1 root root 2866 2006-11-09 00:39 multimedia.rules
-rw-r--r-- 1 root root 1075 2006-11-09 00:39 mysql.rules
-rw-r--r-- 1 root root 284703 2006-11-09 00:39 netbios.rules
-rw-r--r-- 1 root root 3895 2006-11-09 00:39 nntp.rules
-rw-r--r-- 1 root root 176913 2006-11-09 00:39 oracle.rules
-rw-r--r-- 1 root root 1383 2006-11-09 00:39 other-ids.rules
-rw-r--r-- 1 root root 3953 2006-11-09 00:39 p2p.rules
-rw-r--r-- 1 root root 5323 2006-11-09 00:39 policy.rules
-rw-r--r-- 1 root root 1228 2006-11-09 00:39 pop2.rules
-rw-r--r-- 1 root root 8578 2006-11-09 00:39 pop3.rules
-rw-r--r-- 1 root root 5061 2006-11-09 00:39 porn.rules
-rw-r--r-- 1 root root 51378 2006-11-09 00:39 rpc.rules
-rw-r--r-- 1 root root 2920 2006-11-09 00:39 rservices.rules
-rw-r--r-- 1 root root 4092 2006-11-09 00:39 scan.rules
-rw-r--r-- 1 root root 4727 2006-11-09 00:39 shellcode.rules
-rw-r--r-- 1 root root 22138 2006-11-09 00:39 smtp.rules
-rw-r--r-- 1 root root 4915 2006-11-09 00:39 snmp.rules
-rw-r--r-- 1 root root 17310 2006-11-09 00:39 sql.rules
-rw-r--r-- 1 root root 4260 2006-11-09 00:39 telnet.rules
-rw-r--r-- 1 root root 2560 2006-11-09 00:39 tftp.rules
-rw-r--r-- 1 root root 1211 2006-11-09 00:39 virus.rules
-rw-r--r-- 1 root root 10229 2006-11-09 00:39 web-attacks.rules
-rw-r--r-- 1 root root 100925 2006-11-09 00:39 web-cgi.rules
-rw-r--r-- 1 root root 9795 2006-11-09 00:39 web-client.rules
-rw-r--r-- 1 root root 9166 2006-11-09 00:39 web-coldfusion.rules
-rw-r--r-- 1 root root 9484 2006-11-09 00:39 web-frontpage.rules
-rw-r--r-- 1 root root 38547 2006-11-09 00:39 web-iis.rules
-rw-r--r-- 1 root root 94965 2006-11-09 00:39 web-misc.rules
-rw-r--r-- 1 root root 35801 2006-11-09 00:39 web-php.rules

```

Snort lo podemos activar utilizando alguna de las siguientes opciones:

```

,,_ -*> Snort! <*-

```

```

o" )~ Version 2.3.3 (Build 14)

```

```

"" By Martin Roesch & The Snort Team: http://www.snort.org/team.html
(C) Copyright 1998-2004 Sourcefire Inc., et al.

```

USAGE: snort [-options] <filter options>

Options:

- A Set alert mode: fast, full, console, or none (alert file alerts only)
- "unssock" enables UNIX socket logging (experimental).
- b Log packets in tcpdump format (much faster!)
- c <rules> Use Rules File <rules>
- C Print out payloads with character data only (no hex)
- d Dump the Application Layer

- D *Run Snort in background (daemon) mode*
- e *Display the second layer header info*
- f *Turn off fflush() calls after binary log writes*
- F <bpf> *Read BPF filters from file <bpf>*
- g <gname> *Run snort gid as <gname> group (or gid) after initialization*
- h <hn> *Home network = <hn>*
- i <if> *Listen on interface <if>*
- I *Add Interface name to alert output*
- k <mode> *Checksum mode (all,noip,notcp,noudp,noicmp,none)*
- l <ld> *Log to directory <ld>*
- L <file> *Log to this tcpdump file*
- m <umask> *Set umask = <umask>*
- n <cnt> *Exit after receiving <cnt> packets*
- N *Turn off logging (alerts still work)*
- o *Change the rule testing order to Pass\Alert\Log*
- O *Obfuscate the logged IP addresses*
- p *Disable promiscuous mode sniffing*
- P <snap> *Set explicit snaplen of packet (default: 1514)*
- q *Quiet. Don't show banner and status report*
- r <tf> *Read and process tcpdump file <tf>*
- R <id> *Include 'id' in snort_intf<id>.pid file name*
- s *Log alert messages to syslog*
- S <n=v> *Set rules file variable n equal to value v*
- t <dir> *Chroots process to <dir> after initialization*
- T *Test and report on the current Snort configuration*
- u <uname> *Run snort uid as <uname> user (or uid) after initialization*
- U *Use UTC for timestamps*
- v *Be verbose*
- V *Show version number*
- w *Dump 802.11 management and control frames*
- X *Dump the raw packet data starting at the link layer*
- y *Include year in timestamp in the alert and log files*
- z *Set assurance mode, match on established sessions (for TCP)*
- ? *Show this information*

<Filter Options> are standard BPF options, as seen in TCPDump

Uh, you need to tell me to do something...

Para utilizar Snort en modo IDS hacemos:

```
snort -devl /var/log/snort -h 10.100.0.0/24 -c /etc/snort/snort.conf
```

Para utilizar Snort en modo Sniffer:

```
snort -dev
```

Para utilizar Snort en modo packet logger

```
snort -devl /var/log/snort
```


=====
=+

07/16-09:30:00.100198 10.100.0.252:3080 -> 10.100.0.25:2012

TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF

***A*R** Seq: 0x0 Ack: 0x68814548 Win: 0x0 TcpLen: 20

=====
=+

07/16-09:30:02.102200 10.100.0.25:2013 -> 10.100.0.252:3080

TCP TTL:128 TOS:0x0 ID:20735 IpLen:20 DgmLen:48 DF

*****S* Seq: 0xD398E752 Ack: 0x0 Win: 0xFFFF TcpLen: 28

TCP Options (4) => MSS: 1460 NOP NOP SackOK

=====
=+

07/16-09:30:02.102228 10.100.0.252:3080 -> 10.100.0.25:2013

TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF

***A*R** Seq: 0x0 Ack: 0xD398E753 Win: 0x0 TcpLen: 20

=====
=+

07/16-09:30:02.615324 10.100.0.25:2013 -> 10.100.0.252:3080

TCP TTL:128 TOS:0x0 ID:20736 IpLen:20 DgmLen:48 DF

*****S* Seq: 0xD398E752 Ack: 0x0 Win: 0xFFFF TcpLen: 28

TCP Options (4) => MSS: 1460 NOP NOP SackOK

=====
=+

07/16-09:30:02.615353 10.100.0.252:3080 -> 10.100.0.25:2013

TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF

***A*R** Seq: 0x0 Ack: 0xD398E753 Win: 0x0 TcpLen: 20

=====
=+

07/16-09:30:03.052746 10.100.0.25:2013 -> 10.100.0.252:3080

TCP TTL:128 TOS:0x0 ID:20737 IpLen:20 DgmLen:48 DF

*****S* Seq: 0xD398E752 Ack: 0x0 Win: 0xFFFF TcpLen: 28

TCP Options (4) => MSS: 1460 NOP NOP SackOK

=====
=+

07/16-09:30:03.052774 10.100.0.252:3080 -> 10.100.0.25:2013

TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF

***A*R** Seq: 0x0 Ack: 0xD398E753 Win: 0x0 TcpLen: 20

=====
=+

07/16-09:30:04.214435 ARP who-has 10.100.0.25 tell 10.100.0.252

07/16-09:30:04.214576 ARP reply 10.100.0.25 is-at 0:8:A1:6F:E7:9B

=====
=====

Snort received 12 packets
Analyzed: 12(100.000%)
Dropped: 0(0.000%)

=====
=====

Breakdown by protocol:
TCP: 10 (83.333%)
UDP: 0 (0.000%)
ICMP: 0 (0.000%)
ARP: 2 (16.667%)
EAPOL: 0 (0.000%)
IPv6: 0 (0.000%)
IPX: 0 (0.000%)
OTHER: 0 (0.000%)
DISCARD: 0 (0.000%)

=====
=====

Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0

=====
=====

Snort exiting

Snort en modo NIDS

Para activar este modo utilizamos:

```
snort -devl /etc/snort/snort.conf -h 10.100.0.0/24 -c /etc/snort/snort.conf
```

Ademas de las opciones que fueron detalladas anteriormente, le podemos agregar la opcion -D para que corra como servicio. Tambien podemos agregar la opcion -A para configurar las alertas (solo se mostrarán algunas de ellas):

```
snort -A full -devl /etc/snort/snort.conf -h 10.100.0.0/24 -c /etc/snort/snort.conf
```

-A *FAST* es el modo rapido que nos devolverá información mas elemental de la captura

-A *FULL* es el modo completo el cual ademas registrará informacion completa de las

cabeceras de los paquetes registrados.

-A *NONE* desactiva las alarmas

-A *SMB* Realiza llamadas al cliente de SMB y envia mensajes de alerta a los host Windows

(Winpopup)

Reglas Snort

Snort utiliza un ligero y simple lenguaje de reglas que es flexible y poderoso. Las reglas se pueden dividir en dos secciones: Cabecera de la regla y opciones. La cabecera de la regla contiene la acción de la regla en sí, protocolo, IP, mascarar de red, puertos de origen y destino, destino del paquete y dirección de la operación. En las opciones definimos mensajes y la información necesaria para la decisión a tomar por parte de la alerta en forma de opciones.

La estructura de la regla es la siguiente:

Cabecera – acción – protocolo involucrado – dirección IP – número de puerto – dirección de la operación (opciones – mensaje – opciones de decisión)

Ejemplo:

alert tcp any any -> 192.168.1.0/24 111 (content:"!00 01 86 a5!";msg: "mountd access";)