

Sistemas de Detección de Intrusos



Antonio Campoy Torrecillas
Ana Alicia Vílchez Ceballos

1.¿Qué son?

Un Sistema de Detección de Intrusos (IDS) es una herramienta de seguridad que se emplea para detectar y monitorizar accesos no autorizados a un dispositivo o red.

Suelen utilizar Sniffers para obtener datos relacionados con el tráfico de red, de modo que si encuentran alguna anomalía en dicha operación, alertarán al dispositivo para el que se encuentra trabajando.

Para quien no lo sepa: un Sniffer es un programa que captura los paquetes que se transmiten a través de una red, actuando en modo promiscuo (en el cual en la capa de enlace de datos no son descartados los paquetes no destinados a la dirección MAC de la tarjeta de red).

El IDS no sólo analiza qué tipo de tráfico es, sino que también revisa el contenido y el comportamiento del mismo.

También poseen una base de datos de patrones de ataques de modo que en caso de encontrar alguna anomalía, serán comparados con los resultados del análisis hecho previamente.

2.Tipos de IDS

Clasificados según su alcance de protección disponemos de los siguientes tipos:

-HIDS(Host IDS): Protege contra un servidor o PC. Se basa en la monitorización de eventos del sistema para detectar las posibles modificaciones que han podido hacer los intrusos.

-NIDS(Net IDS): Actúan en la red del sistema. Se basan en Sniffers que hemos configurado previamente para provocar una alerta en caso de que detecten cierto tipo de actividades. Estos Sniffers actúan en modo promiscuo para capturar todo el tráfico.

-DIDS(Distributed IDS): Es una evolución del anterior. Se basan en una arquitectura cliente-servidor, formada por un conjunto de NIDS, cada NID puede ser configurado con distintas reglas. Estos envían la información a una base de datos central.

También podemos clasificarlos por su tipo de respuesta:

- Pasivos: este tipo de IDS recopila y analiza los datos, ante un posible ataque nos notifica con una alerta, pero no hace nada por evitar el ataque.

- Activos: este tipo de IDS sí que genera una respuesta ante un posible ataque. Trata de evitarlo aplicando una serie de reglas definidas previamente.

3. ¿Por qué instalar un IDS?

Cuando tenemos un servidor de cara a internet debemos tomar todas las medidas necesarias para asegurarnos de que nuestros datos están a salvo. Un IDS es una parte de estas medidas y nos puede ayudar en las siguientes tareas:

- Prevenir ataques que otras medidas de seguridad no son capaces de detectar.

- Detectar las acciones previas a un ataque.

- Recolectar información de cómo se ha producido un ataque.

4. Suricata



Suricata es un Sistema de Detección de Intrusos de código abierto, el proyecto empezó a finales de 2009 y sigue en marcha. Es un IDS del tipo NIDS(Net IDS). El motor de suricata analiza el tráfico en tiempo real, también incorpora IPS(Sistema de Prevención de Intrusos) y procesamiento de paquetes offline. Suricata utiliza un lenguaje extenso y potente para definir las reglas de detección de paquetes.

4.1 Características de Suricata

Multi-Threaded Processing: Como bien imagináis por el título de este apartado, Suricata permite la ejecución de varios procesos de forma simultánea. Se puede asignar por tanto un número de threads para cada core del sistema.

A continuación se presentan los submódulos del IDS:

- Capturar paquetes.
- Seguir el flujo de secuencias y conexiones.
- Comparar el flujo con las firmas de “ataques famosos” que se encuentran registradas en el sistema.
- Procesar los eventos y la salida de alertas.

Cada una de estas tareas podrán ser procesadas de forma simultánea en función del número de cores que posee el sistema.

Automatic Protocol Detection: además de incorporar reglas específicas para los protocolos IP, TCP, UDP e ICMP, Suricata incorpora reglas para otros protocolos como FTP, HTTP, TLS y SMB. Esto nos proporciona opciones para definir reglas distintas e independientes para los distintos protocolos de red.

Performance Statistics: Las estadísticas y el análisis de rendimiento se vuelcan en el archivo `‘/var/log/suricata/stat.log’`, de esta forma podemos revisarlos posteriormente.

HTTP Log Module: Suricata guarda todas las peticiones HTTP, independientemente de las alertas, estas son guardadas en el archivo `‘/var/log/suricata/http.log’`. De esta forma podremos revisarlas en cualquier momento . Estas peticiones son almacenadas por defecto en formato Log Apache, pero podemos configurarlo para guardarlas con otro tipo de formato.

Fast Log Module: Suricata almacena en el archivo `‘var/log/suricata/fast.log’` el log de los ataques que realizan contra nuestro servidor, la mayoría de veces estos son rechazados sin que seamos conscientes de ello.

4.2 Instalación de Suricata

Antes de proceder a instalar Suricata, deberemos actualizar e instalar una serie de paquetes en nuestro sistema para que Suricata funcione correctamente:

```
apt-get update
```

```
apt-get install libpcap3-dev libpcap3-dbg libpcap3-dev autoconf automake libtool libpcap-dev  
libnet1-dev libyaml-dev zlib1g-dev libcap-ng-dev libmagic-dev libjansson-dev  
libjansson4
```

Para obtener el IDS en la última versión, nos vamos al siguiente enlace:

<http://www.openinfosecfoundation.org/download/>

y descargamos el tar que mejor nos venga (siempre recomendando la última versión).

Una vez descargado lo descomprimos y nos movemos al directorio. Por ejemplo si la versión descargada es la 3.1.1, se puede realizar de la siguiente forma:

```
tar -zxf suricata-3.1.1.tar.gz  
cd suricata-3.1.1/
```

Una vez dentro, si queremos instalar el IDS con la configuración por defecto, podemos ejecutar el makefile llamado instal-conf:

```
make install-conf
```

Si queremos añadir las reglas que trae el IDS, podemos ejecutar el makefile llamado `install-rules`:

```
make install-rules
```

Las reglas que se han descargado al ejecutar el makefile se encuentran en el directorio `/etc/suricata/rules`

Con este paso ya tendríamos instalado Suricata con las reglas creadas por sus desarrolladores y la configuración por defecto.

4.3 Crear Reglas en Suricata

Al instalar Suricata ya hemos descargado diferentes tipos de reglas que vienen por defecto, sin embargo en ciertas ocasiones conviene modificarlas o incluso crear nuestras propias reglas.

Las reglas en Suricata se guardan en la carpeta `rules`:

```
/etc/suricata/rules
```

En ese directorio encontraremos todos los archivos de reglas que tienen extensión `'rules'`. Ahora para crear nuestras propias reglas solo debemos crear un nuevo archivo en esta carpeta al que le pondremos la extensión `'rules'`. Por ejemplo nosotros creamos una regla para detectar conexiones ssh así que lo llamamos `'ssh.rules'`, pero el nombre anterior a `.rules` puede ser cualquiera.

Una vez creado el archivo, deberemos comprender un poco la sintaxis de las reglas, que es la siguiente:

```
alert PROTOCOLO IP_SOURCE PORT_S -> IP_DEST PORT_D (msg:"algo";  
flow: established,to_server; app-layer-protocol:PROTOCOLO; sid:XXXXXXXX;  
rev:Y;)
```

Veamos para qué sirven los campos más importantes de las reglas:

-Protocolo: indicamos el tipo de protocolo que utiliza el paquete que vamos a inspeccionar. Suelen ser TCP o UDP, pero como hemos visto antes en las características Suricata trabaja con bastantes tipos de protocolo.

-IP_SOURCE: aquí va la IP de quién manda, podemos utilizar “any” si no queremos indicar ninguna IP concreta, es lo que se suele utilizar. Además en el archivo de configuración de suricata “etc/suricata/suricata.yaml” se definen variables de las ips:

```
# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricatayaml

##
## Step 1: inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

    HTTP_SERVERS: "$HOME_NET"
    SMTP_SERVERS: "$HOME_NET"
    SQL_SERVERS: "$HOME_NET"
    DNS_SERVERS: "$HOME_NET"
    TELNET_SERVERS: "$HOME_NET"
    AIM_SERVERS: "$EXTERNAL_NET"
    DNP3_SERVER: "$HOME_NET"
    DNP3_CLIENT: "$HOME_NET"
    MODBUS_CLIENT: "$HOME_NET"
```

En estas variables podemos definir, como vemos, las IPs de nuestra red, y podrán ser utilizadas a la hora de definir las IPs en las reglas, por ejemplo podríamos usar ![\$HOME_NET] para referirnos a cualquier IP distinta de las definidas ahí.

-PORT_S: aquí indicamos el puerto por el que salió del paquete desde el emisor, también podemos usar “any” para referirnos a cualquiera.

-IP_DEST: aquí ponemos la IP de destino del paquete. Podemos utilizar lo visto en IP_SOURCE. Normalmente ahí pondremos nuestra red, que vendría definida en [\$HOME_NET], también podríamos utilizar “any”.

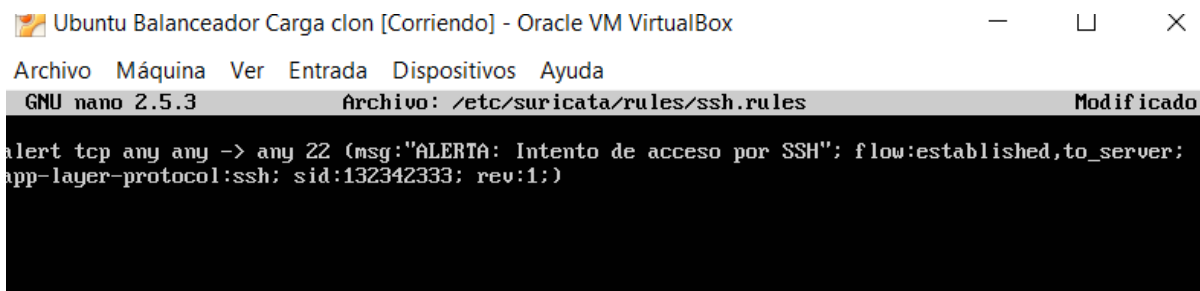
-PORT_D: aquí viene el puerto de destino del paquete analizado, tendremos que conocer el protocolo del que creamos la regla de alerta para saber esto.

-msg: aquí pondremos el mensaje que se mostrará en la alerta. Cuando se produzca este mensaje se grabará en “var/log/suricata/fast.log”.

-SID: es el identificador de la regla, debe ser un número de 9 dígitos, y no debe coincidir con ningún otro SID previamente definido.

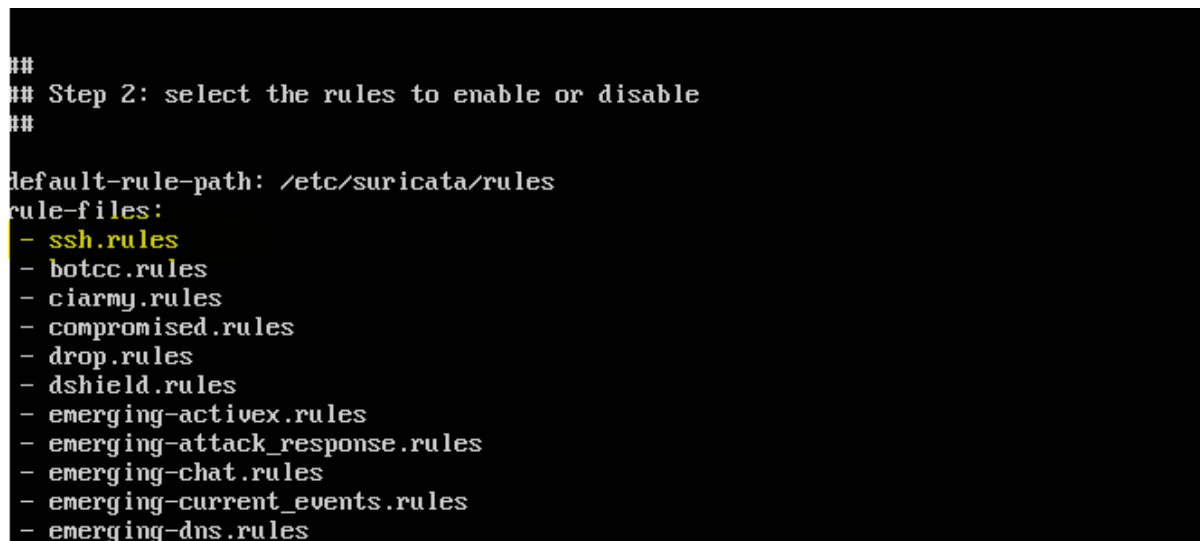
-REV: prioridad de la regla.

Ahora ya conocemos los campos más importantes para definir una regla. Para la demostración, nosotros hicimos una regla que nos alertaría de cualquier conexión SSH a nuestra máquina:



```
Ubuntu Balanceador Carga clon [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 2.5.3      Archivo: /etc/suricata/rules/ssh.rules      Modificado
alert tcp any any -> any 22 (msg:'ALERTA: Intento de acceso por SSH'; flow:established,to_server;
app-layer-protocol:ssh; sid:132342333; rev:1;)
```

Ahora una vez creada la regla deberemos añadir el nombre de nuestro archivo .rules en la lista de los archivos de reglas del archivo de configuración("etc//suricata.yaml"):



```
##
## Step 2: select the rules to enable or disable
##

default-rule-path: /etc/suricata/rules
rule-files:
- ssh.rules
- botcc.rules
- ciarmy.rules
- compromised.rules
- drop.rules
- dshield.rules
- emerging-activex.rules
- emerging-attack_response.rules
- emerging-chat.rules
- emerging-current_events.rules
- emerging-dns.rules
```

Una vez hecho esto solo debemos arrancar Suricata, y si todo está correcto, debería funcionar. De haber algún error se nos mostraría cuando iniciemos Suricata en el terminal.

4.4 Script creado para la ejecución de Suricata

Cuando empezamos a configurar Suricata, poco a poco fuimos comprendiendo cómo se aplicaban sus diferentes funcionalidades. Conseguimos que detectara los accesos por SSH con la regla que definimos y que las alertas las almacenará en el archivo fast.log. Sin embargo no pudimos obtener la forma de que la alerta apareciera “en vivo”, es decir, que se mostrará en el terminal mientras el programa se encontraba en modo activo esnifando el tráfico de red.

Es por eso que decidimos crear un script que además de lanzar Suricata, nos diera la última alerta almacenada en fast.log cada vez que se producía alguna. A continuación se muestra el script:

```
GNU nano 2.5.3          Archivo: SuricataScript          Modificado
#!/bin/bash

/usr/bin/suricata -c /etc/suricata/suricata.yaml -i enp0s3 &

tam=`cat /var/log/suricata/fast.log | wc -c`
while :
do
    tam2=`cat /var/log/suricata/fast.log | wc -c`

    if [ "$tam2" != "$tam" ]
    then
        tail -n 2 /var/log/suricata/fast.log
        tam=$tam2
    fi
done
```

Como se puede observar, el script lanza el programa y guarda el tamaño de fast.log. En el bucle infinito, en caso de que el tamaño del archivo cambie, mostrará con tail las dos últimas líneas del archivo que almacena las alertas.

5. Enlaces consultados

A la hora de realizar el trabajo hemos visitado muchas páginas web, estas son las que nos han sido más útiles:

-Sistemas de Detección de Intrusos:

<https://www.linux-party.com/index.php/6000-el-sistema-de-deteccion-de-intrusos--snort---windows-y-linux->

<http://deim.urv.cat/~pfc/docs/pfc375/d1126516530.pdf>

https://www.dspace.espol.edu.ec/bitstream/123456789/19502/1/Diapositivas_tesina.pdf

<http://rediris.es/cert/doc/pdf/ids-uv.pdf>

-Suricata:

<https://suricata-ids.org/>

<http://linuxpitstop.com/install-suricata-ids-on-ubuntu-16-04/>

<https://seguridadyredes.wordpress.com/2011/02/22/ids-ips-suricata-entendiendo-y-configurando-suricata-parte-i/>