

IES Gonzalo Nazareno

Sistemas detección de intrusos

Juan Javier Rodríguez Guisado

1.Introducción

Los **sistemas de detección de intrusos** o **IDS** (*Intrusion Detection System*) son programas que monitorizan la actividad del sistema con el fin de detectar accesos o acciones no autorizados a una máquina o a una red.

A la hora de implantar un sistema de detección de intrusos se deben tener en cuenta dos posibilidades, hardware o software (siendo posible una combinación de los dos) La opción de optar por hardware es más usual cuando el tráfico de red es muy alto.

Podemos diferenciar dos tipos de IDS, HIDS y NIDS

- *HIDS* – Host Intrusion Detection System: Está basado en el propio host donde está instalado, busca actividad sospechosa en la propia máquina analizando el sistema constantemente.
- *NIDS* – Network Intrusion Detection System: Está basado en una red, busca actividad sospechosa analizando el tráfico de red (Incluido el tráfico local)

Estos sistemas, en caso de encontrar anomalías pueden tomar medidas o sólo conservar la información del sistema, a esto se les llaman **sistemas pasivos** o **sistemas reactivos**:

- *Sistemas pasivos*: Cuando detectan una actividad similar a un acción o acceso no autorizado lo registra en una base de datos.
- *Sistemas activos*: Cuando detectan una actividad similar a una acción o acceso no autorizado el sistema responde bloqueando la posible intrusión y guardándolo en la base de datos.

2. Instalación de Ossec



Ossec es un HIDS muy potente, además es Open Source, para instalarlo no tenemos más que ir a su página oficial y descargar la última versión disponible, la 2.4 en mi caso. Este HIDS está disponible para GNU/Linux, Windows(sólo agente),
* BSD, Solaris, Mac OS...

Página de descargas de Ossec:

<http://www.ossec.net/main/downloads>

Requisitos previos para instalar Ossec:

- Tener instalado un compilador de C
- Tener instalado un servidor de correo o disponer de uno.

Para instalarlo en nuestra máquina linux (Debian o derivados), tenemos que hacer los siguientes pasos. (Hay una forma más manual de instalarlo, pero usaremos el script que automatiza toda la instalación)

```
#tar -xvf ossec-hids-2.4.tar.gz -C /opt
```

```
#cd /opt/ossec-hids-2.4/
```

```
#./install.sh
```

Lo que acabamos de hacer es descomprimir el archivo, entrar en la carpeta donde lo descomprimos y ejecutar el script de instalación. Este script nos preguntará el idioma en el que deseamos instalar ossec, tendremos que introducir el código de idioma, que en nuestro caso es “es” que pertenece a “español”.

A continuación nos preguntará que tipo de instalación deseamos hacer, en nuestro caso vamos a hacer la instalación de servidor. Escribimos ‘servidor’ y seguimos con la instalación. La instalación de ossec en tipo servidor sólo se puede hacer en sistemas operativos GNU/Linux.

Nota: El modo local hace que el host se proteja a el mismo únicamente, se diferencia del modo servidor porque el servidor puede gestionar todo el proceso de los agentes (clientes) centralizandolo y descargando actividad de las demás máquinas de las cuales recibe los logs y eventos de sistema

Tras esto, la instalación configurará las variables de entorno del sistema que usará y te preguntará el directorio de instalación de Ossec (y muestra uno por defecto “/var/ossec”) podemos introducir uno manualmente o dejar el que esta por defecto.

Ahora la instalación empezará a configurar el sistema ossec y nos hará algunas preguntas:

- Si deseamos recibir notificaciones por email (necesitaremos usar un servidor de correo electrónico, puedes instalar uno en la misma máquina por ejemplo **postfix**)
- Si queremos hacer uso del servidor de integridad del sistema, obviamente es una parte importante del HIDS y es recomendable hacer uso del mismo, este demonio es el encargado de monitorizar y reportar cambios en los archivos de sistema.
- Si queremos activar el sistema de detección de rootkits, un rootkits es un malware con el objetivo de ganar privilegios en el sistema, por lo tanto deberíamos de activarla.
- Si deseamos activar la respuesta activa del HIDS, las respuestas activas nos permiten definir unas acciones que se ejecutarán de forma automática para evitar una posible intrusión o ataque tanto usando el /etc/host.deny o iptables (o ipfilter en caso de BSD o Solaris)
- Ips a la lista blanca, la “lista blanca” es un archivo donde se definimos unas IP en las que “confiamos”, por defecto se agrega la puerta de enlace de la máquina, y te pregunta si deseas añadir alguna IP más a “lista blanca”.
- Si deseas activar las alertas desde un syslog remoto

Una vez respondemos todo, el instalador compilará durante unos minutos, una vez compile nos dirá como iniciar y como parar Ossec.

```
#!/var/ossec/bin/ossec-control start  
#!/var/ossec/bin/ossec-control stop
```

El archivo de configuración de Ossec se encuentra en */var/ossec/etc/ossec.conf*

Como último el instalador nos dice que para conectar a un agente al servidor, debemos agregarlos (a los agentes) a el servidor, podemos hacerlo desde el programa “manage_agents” (*/var/ossec/bin/manage_agents*)

Instalar la interfaz Web

Ossec cuenta con una aplicación web para hacer más cómodo el análisis del HIDS, podemos descargarla de la página oficial.

```
#wget http://www.ossec.net/files/ui/ossec-wui-0.3.tar.gz
```

Lo vamos a montar sobre apache, así que tendremos que instalar apache, php y el módulo de apache que soporta php (se instala por defecto si instalamos php después de apache) en nuestra máquina.

```
#aptitude install apache2 php5
```

Movemos la carpeta de la interfaz web a la carpeta web de apache y ejecutamos el script de instalación, en este tendremos que crearnos un usuario para loguearnos en la interfaz web.

```
#mv ossec-wui-0.3 /var/www/ossecui
```

```
#!/var/www/ossecui/setup.sh
```

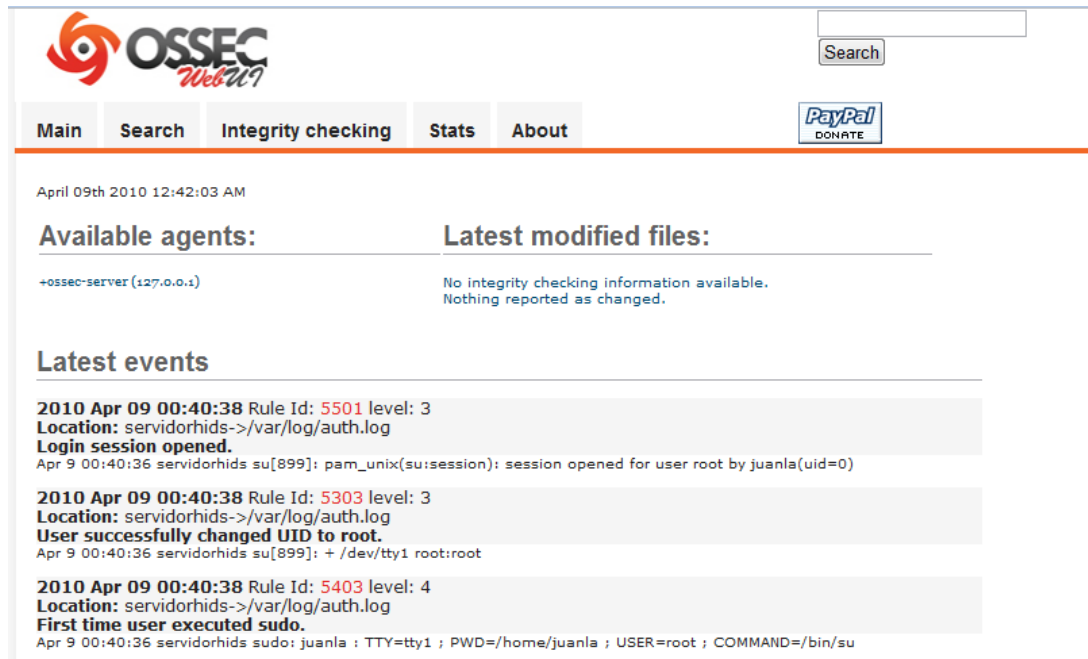
Ahora editamos el archivo */etc/group*, buscamos la línea donde aparece ossec, será algo así como “ossec:x:1001:” y a continuación añadimos el nombre de usuario de apache, lo normal es que sea “www-data”, quedando así “ossec:x:1001:www-data”

Una vez echo damos permisos a la carpeta *tmp de la interfaz web (/var/www/ossecui/tmp)* y reiniciamos apache para que los cambios tengan efecto:

```
/var/www/ossecui#chmod 770 tmp/  
/var/www/ossecui#chgrp www-data tmp/  
#!/etc/init.d/apache2 restart
```

Ya podemos acceder a la interfaz web de Ossec escribiendo la URL pertinente

<http://localhost/ossecui>



Captura de pantalla de la página de administración web de ossec

Instalar los agentes

Para instalar los agentes, hacemos del mismo modo que al instalar el servidor, pero en el paso que nos pregunta cómo vamos a instalar elegimos agente en vez de servidor.

Sólo nos quedará conectar los agentes a el servidor.

Conectar agentes con servidor

La comunicación entre agente – servidor , va cifrada y autenticada, por esto para conectar un agente con un servidor tenemos que crear una clave en el servidor, y exportarla hacia los clientes.

Ossec cuenta con una aplicación específica para añadir agentes `"/var/ossec/bin/manage_agents"` los pasos que hay que seguir para hacerlos son los siguientes:

En el servidor (repetir por cada agente que vallamos añadir):

`#/var/ossec/bin/manage_agents`

```
*****
* OSSEC HIDS v2.3 Agent manager. *
* The following options are available: *
*****
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
```

Menú principal en la máquina servidor

Este programa nos mostrará un menú con 4 opciones, añadir un agente, extraer clave de un agente, listar agentes y eliminar un agente. Obviamente vamos a añadir un agente (o los que fueran necesarios) Para ello, usaremos la opción A, añadir un agente, introduciremos la IP del agente, un nombre para ese agente y una clave de identificación (001,002.. por defecto).

```
Choose your action: A,E,L,R or Q: A
- Adding a new agent (use '\q' to return to the main menu).
  Please provide the following:
    * A name for the new agent: agente1
  ** Name 'agente1' already present. Please enter a new name.
    * A name for the new agent: agente
    * The IP Address of the new agent: 192.168.1.50
    * An ID for the new agent[003]: 003
Agent information:
  ID:003
  Name:agente
  IP Address:192.168.1.50
```

Introducimos los datos de nuestro agente en la máquina servidor

Una vez añadido el agente, necesitamos su clave pública, para ello usaremos la segunda opción (E) que nos mostrará los agentes, y al poner su identificador nos dará la clave pública.

```
Choose your action: A,E,L,R or Q: E
Available agents:
  ID: 001, Name: agente1, IP: 192.168.1.132
  ID: 002, Name: agente2, IP: 192.168.1.88
Provide the ID of the agent to extract the key (or '\q' to quit): 001
Agent key information for '001' is:
MDAxIGFnZW50ZTEgMTkyLjE2ODQxLjEzMiBkMzh1ZmQwMDR1NjA4ZmMzZjd1NDZjZTN1NzYzZTk2ODI4
Nzk5MTIxOTMxNDRkYjM5YjNjNjg2NTNmZWJkMTM0
```

Extrayendo la clave de uno de los agentes

El siguiente paso se realizará *en el agente o agentes*, abrimos el mismo programa que en el servidor `/var/ossec/bin/manage_agents`, nos daremos cuenta de que en este sólo tenemos 2 opciones disponibles, Importar clave pública y eliminar, nosotros vamos a importar la clave pública, elegiremos la opción (I) y pegaremos la clave que nos proporcionó el servidor.

```
root@agente1:/home/agente1# /var/ossec/bin/manage_agents

*****
* OSSEC HIDS v2.3 Agent manager.      *
* The following options are available: *
*****
  (I)mport key from the server (I).
  (Q)uit.
Choose your action: I or Q: I

* Provide the Key generated by the server.
* The best approach is to cut and paste it.
*** OBS: Do not include spaces or new lines.

Paste it here (or '\q' to quit): MDAxIGFnZW50ZTEgMTkyLjE2ODQxLjEzMiBkMzh1ZmQwMDR1NjA4ZmMzZjd1NDZjZTN1NzYzZTk2ODI4Nzk5MTIxOTMxNDRkYjM5YjNjNjg2NTNmZWJkMTM0

Agent information:
  ID:001
  Name:agente1
  IP Address:192.168.1.132
```

Añadiendo la clave del servidor en el agente

Una vez añadidos todos los agentes de la misma manera, tenemos que reiniciar el servidor:

```
#/var/ossec/bin/ossec-control restart
```

Desinstalar ossec

```
rm -rf /var/ossec
rm -f /etc/init.d/ossec
rm -f /etc/ossec-init.conf
```

Instalar ossec en una máquina agente con Windows

Vamos a hacer lo mismo ahora para Windows XP, para ello nos descargamos la versión para windows de la página oficial. Lo ejecutamos y instalamos el agente y los complementos que queramos.



Al finalizar la instalación nos pedirá la ip del servidor ossec y el código de autenticación. Ponemos la IP y el código de autenticación y guardamos. Esta aplicación que se ejecuta automáticamente al finalizar la instalación la podemos encontrar en **Inicio - Programas - Ossec - Ossec Agent**

Ahora nos conectamos mediante ssh a nuestro servidor y seguimos el procedimiento de arriba para extraer la clave de el agente (Opción E) y la ponemos.



Y con esto ya tenemos añadido un agente Windows

3. Iniciar Ossec y los agentes y comprobar su funcionamiento

Una vez instalado, debemos iniciar los demonios de ossec en las máquinas (al reiniciar se inician automáticamente con ella) para iniciarlos a mano escribimos:

```
#/var/ossec/bin/ossec-control restart
```

Vamos a probar el funcionamiento intentando atacar de un agente a otro mediante conexiones ssh inválidas, ossec lo detectará como un ataque de fuerza bruta (analizando las secuencias de ataque y comparándolas con las de su base de datos) y automáticamente tomará medidas.

Nos enviará un correo electrónico (si hemos instalado y configurado un servidor de correo) con la información de la alerta (dependiendo de como lo configures será más flexible o menos flexible a la hora de enviar alertas):

```
Message 2:
From ossecm@servidorhids Fri Apr 9 12:16:48 2010
X-Original-To: root@localhost
To: <root@localhost>
From: OSSEC HIDS <ossecm@servidorhids>
Date: Fri, 09 Apr 2010 12:16:47 +0200
Subject: OSSEC Notification - (agente1) 192.168.1.132 - Alert level 10

OSSEC HIDS Notification.
2010 Apr 09 12:16:34

Received From: (agente1) 192.168.1.132->/var/log/auth.log
Rule: 5720 fired (level 10) -> "Multiple SSHD authentication failures."
Portion of the log(s):

Apr 9 12:16:36 agente1 sshd[1640]: Failed password for root from 192.168.1.188
port 41528 ssh2
Apr 9 12:16:35 agente1 sshd[1640]: Failed password for root from 192.168.1.188
port 41528 ssh2
Apr 9 12:16:29 agente1 sshd[1640]: Failed password for root from 192.168.1.188
port 41528 ssh2
Apr 9 12:16:22 agente1 sshd[1638]: Failed password for root from 192.168.1.188
port 41527 ssh2
Apr 9 12:16:19 agente1 sshd[1638]: Failed password for root from 192.168.1.188
```

Añadirá una línea al archivo /etc/hosts.deny con la IP del presunto atacante:

```
root@agente1:/home/agente1# cat /etc/hosts.deny |grep -v ^$ |grep -v ^#
ALL:192.168.1.188
```

Si mostramos las reglas iptables del sistema (iptables -L) veremos denegado al host atacante:

```
root@servidorhids:~# iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source                destination
DROP      all  --  192.168.1.151          anywhere

Chain FORWARD (policy ACCEPT)
target    prot opt source                destination
DROP      all  --  192.168.1.151          anywhere

Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination
```

En el log de la máquina agente1(atacante) se escribe en el log lo siguiente:

```
Fri Apr 9 12:16:36 CEST 2010 /var/ossec/active-response/bin/firewall-drop.sh add - 192.168.1.188 1270808194.201042 5720
```


Fri Apr 9 12:16:36 CEST 2010 /var/ossec/active-response/bin/host-deny.sh add -
192.168.1.188 1270808194.201042 57204.Archivo de configuración del servidor Ossec

El archivo de configuración (/var/ossec/etc/ossec.conf) se organiza en varias partes.

La primera parte, la parte global, es la que define el servidor de correo y la configuración del mismo (si usamos).

Podríamos activar la notificación por email configurando esos campos:

```
<global>
  <email_notification>yes</email_notification>
  <email_to>root@localhost</email_to>
  <smtp_server>127.0.0.1</smtp_server>
  <email_from>ossecm@servidorhids</email_from>
</global>
```

A continuación podemos ver las reglas que usa ossec, quitar o añadir las, estás reglas se encuentran en /var/ossec/rules/:

```
<rules>
  <include>rules_config.xml</include>
  <include>pam_rules.xml</include>
  <include>sshd_rules.xml</include>
  ...
  <include>asterisk_rules.xml</include>
  <include>ossec_rules.xml</include>
  <include>attack_rules.xml</include>
  <include>local_rules.xml</include>
</rules>
```

Un ejemplo de el archivo de reglas de ssh (el que vimos actuando antes) es:

```
<group name="syslog,sshd,">
  <rule id="5700" level="0" noalert="1">
    <decoded_as>sshd</decoded_as>
    <description>SSHD messages grouped.</description>
  </rule>
```

Como podemos ver se definen la clave de la regla, el nivel (para organizar la importancia), el nombre y una descripción (ya que es lo que nos proporcionará por email)

La siguiente parte del archivo de configuración es la encargada de definir las opciones del sistema de integridad de archivos, donde se definen los directorios que se deben de omitir o los que se deben de comprobar, cada que tiempo (definido en segundos)..

```
<syscheck>
  <!-- Frequency that syscheck is executed - default to every
22 hours -->
  <frequency>79200</frequency>

  <!-- Directories to check (perform all possible
verifications) -->
```

```

<directories
check_all="yes">/etc,/usr/bin,/usr/sbin</directories>
<directories check_all="yes">/bin,/sbin</directories>

<!-- Files/directories to ignore -->
<ignore>/etc/mtab</ignore>
<ignore>/etc/mnttab</ignore>
...
<!-- Windows files to ignore -->
<ignore>C:\WINDOWS\System32\LogFiles</ignore>
<ignore>C:\WINDOWS\Debug</ignore>
...

```

A continuación podremos ver cómo está configurado el sistema de detección de rootkits, esos archivos de texto definen normas para poder detectar rootkits, trojanos, etc.. :

```

<rootcheck>
  <rootkit_files>/var/ossec/etc/shared/rootkit_files.txt</rootkit_files>
  <rootkit_trojans>/var/ossec/etc/shared/rootkit_trojans.txt</rootkit_trojans>
  <system_audit>/var/ossec/etc/shared/system_audit_rcl.txt</system_audit>
  <system_audit>/var/ossec/etc/shared/cis_debian_linux_rcl.txt</system_audit>
  <system_audit>/var/ossec/etc/shared/cis_rhel_linux_rcl.txt</system_audit>
  <system_audit>/var/ossec/etc/shared/cis_rhel5_linux_rcl.txt</system_audit>
</rootcheck>

```

El siguiente “tramo” del archivo de configuración corresponde a los host de la lista blanca (los definimos con la instalación, en nuestro caso en la lista blanca sólo está nuestro propio host, dominio y puerta de enlace:

```

<global>
  <white_list>127.0.0.1</white_list>
  <white_list>^localhost.localdomain$</white_list>
  <white_list>192.168.1.1</white_list>
</global>

```

También podemos ver algunos tramos que definen que el syslog esta activado:

```

<remote>
  <connection>syslog</connection>
</remote>

```

Algunos scripts que son la base de la respuesta activa(para el host.deny, iptables..) Este por ejemplo ejecuta el script host-deny.sh, y tiene un tiempo predefinido (al tiempo la entrada del host deny es borrada, eso se define en la configuración de respuesta activa):

```

<command>
  <name>host-deny</name>
  <executable>host-deny.sh</executable>
  <expect>srcip</expect>
  <timeout_allowed>yes</timeout_allowed>
</command>

```

La configuración de respuesta activa (que usa los comandos que hemos visto) en este caso el regla host-deny que se corresponde con el comando host-deny:

```
<!-- Active Response Config -->
<active-response>
  <!-- This response is going to execute the host-deny
    - command for every event that fires a rule with
    - level (severity) >= 6.
    - The IP is going to be blocked for 600 seconds.
  -->
  <command>host-deny</command>
  <location>local</location>
  <level>6</level>
  <timeout>600</timeout>
</active-response>
```

Ahora podemos ver que archivos se monitorizán localmente:

```
<!-- Files to monitor (localfiles) -->

<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/messages</location>
</localfile>
...
```

5. Archivo de configuración del agente ossec.

Este archivo es mucho más simple que el de servidor, la principal diferencia es este tramo:

```
<client>

  <server-ip>192.168.1.155</server-ip>

</client>
```

Y que solo tiene las entradas para monitorizar rootkits, los logs de sistema y el sistema de integridad de ficheros.

6. Instalación de Snort.

Snort es otro Sistema de detección de intrusos, esta vez de tipo **NIDS** (trabaja en red a diferencia de ossec, que tiene un comportamiento más “interno”), por lo tanto trabaja escaneando y analizando los paquetes que circulan en la red, identifica posibles ataques según el comportamiento de los mismos,



dispone de unas reglas de actualización diaria que hacen de él una herramienta importante en el mundo de la seguridad informática. La manera más básica de instalar snort permite la archivación de logs de systema, pero vamos intentar guardar esos logs en una base de datos y verlos desde algún administrador web.

Requisitos

Para una instalación sin problemas debemos tener los siguientes paquetes funcionando en nuestro sistema, como partimos de una distribución de Ubuntu 10.4 limpia, vamos a instalarlos uno a uno.

```
#aptitude install nmap
#aptitude install nmap
#aptitude install apache2
#aptitude install php5
#aptitude install php5-mysql
#aptitude install php5-gd
#aptitude install libpcap0.8-dev
#aptitude install libpcap3-dev
#aptitude install g++
#aptitude install mysql-server
#aptitude install libmysqlclient16-dev
```

Comenzando la instalación

Nos podríamos instalar snort desde los mismos repositorios de ubuntu, pero no tendríamos la última versión corriendo en nuestro sistema, así que la descargaremos de su página oficial y lo compilaremos para su instalación. Actualmente la última versión es la 2.8.6. Podemos crear un directorio temporal para tener todos los archivos que usaremos para la instalación:

```
#mkdir snort
#cd snort
#wget http://dl.snort.org/snort-current/snort-2.8.6.tar.gz
#tar xvf snort-2.8.6.tar.gz
#cd snort-2.8.6
```

Ahora compilamos el programa, le indicaremos que instale snort en la ruta /usr/local/snort

```
#./configure --prefix=/usr/local/snort
#make
#make install
```

El programa debería de compilarse e instalarse sin problemas, una vez instalado vamos a crear un usuario y grupo para snort y un directorio para logs, donde daremos permisos al usuario que crearemos:

```
#mkdir /var/log/snort
#groupadd snort
#useradd -g snort snort
#chown snort:snort /var/log/snort
```

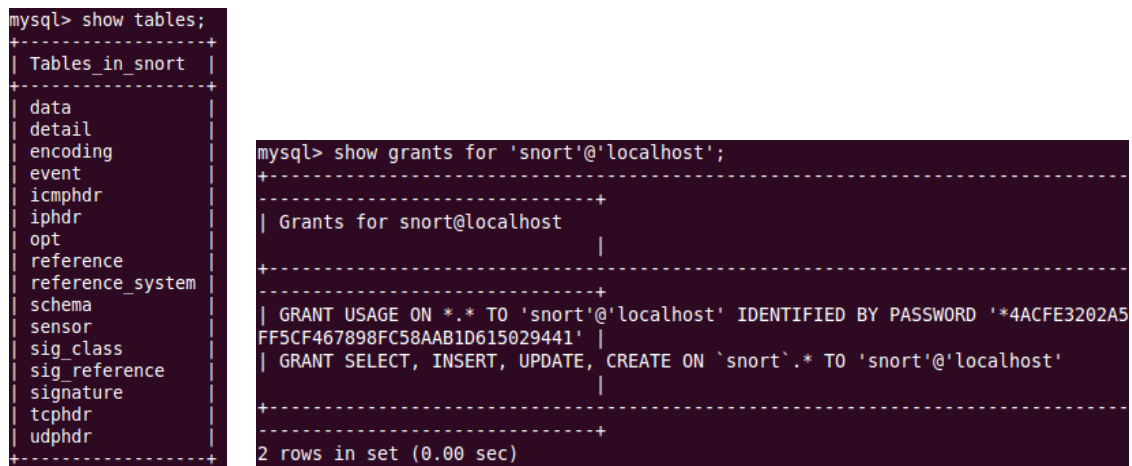
Creando la base de datos (MySQL)

El siguiente punto de instalación **es crear una base de datos** (MySQL en nuestro caso), como hemos instalado anteriormente el servidor no deberíamos de tener problemas:

```
#echo "create database snort" | mysql -u root -p
#mysql -u root -p -D snort < ./schemas/create_mysql
```

Debemos recordad nuestra contraseña de mysql, para no ejecutar la base de datos de snort con el usuario root de mysql (por seguridad) vamos a crear un usuario específico llamado snort:

```
echo "grant create, insert, select, delete, update on snort.* to snort@localhost identified by 'contraseña'" | mysql -u root -p
```



```
mysql> show tables;
+-----+
| Tables_in_snort |
+-----+
| data             |
| detail           |
| encoding         |
| event            |
| icmphdr          |
| iphdr            |
| opt              |
| reference         |
| reference_system |
| schema           |
| sensor           |
| sig_class        |
| sig_reference    |
| signature        |
| tcphdr           |
| udphdr           |
+-----+

mysql> show grants for 'snort'@'localhost';
+-----+
| Grants for snort@localhost |
+-----+
| GRANT USAGE ON *.* TO 'snort'@'localhost' IDENTIFIED BY PASSWORD '*4ACFE3202A5FF5CF467898FC58AAB1D615029441' |
| GRANT SELECT, INSERT, UPDATE, CREATE ON `snort`.* TO 'snort'@'localhost' |
+-----+
2 rows in set (0.00 sec)
```

Añadiéndole las reglas

Lo siguiente que necesitamos para continuar con la instalación de snort son las **reglas**, sin las reglas no podremos hacer funcionar el programa y de echo no nos serviría de nada, estas reglas se pueden descargar de la página oficial de snort siendo un usuario suscrito (con derecho a actualizaciones), se actualizan a diario. Por suerte, si eres un nuevo usuario puedes descargar unas reglas y durante los 30 primeros días actualizarlas.

Para descargar las reglas a nosotros nos bastaría con registrarnos como nuevo usuario y pinchar en el enlace “Get Rules” y descargar las reglas que pertenecen a nuestra versión, las descomprimos en nuestro directorio de instalación de snort y vamos a crear una carpeta donde vamos a copiar las reglas.

```
#wget http://dl.snort.org/reg-rules/snortrules-snapshot-2860.tar.gz
#tar xvf snortrules-snapshot-2860_s.tar.gz -C /usr/local/snort
#mkdir /usr/local/snort/lib/snort_dynamicrules
#cp /usr/local/snort/so_rules/precompiled/Debian-Lenny/i386/2.8.6.0/*
/usr/local/snort/lib/snort_dynamicrules
```

Con esto ya tenemos instalado snort en nuestra máquina, preparado la base de datos y le hemos puesto las reglas correspondientes a su versión.

Instalación de barnyard

Para comenzar con el final de la instalación vamos a instalar Barnyard2 (un fork de barnyard que se adaptó a la nueva estructura de snort, sirve para agilizar el análisis de archivos de snort, envío y reenvío de datos, interacción con la base de datos, plugins, alertas..) un programa que ayuda a snort a trabajar. Para ello lo descargamos de web oficial:

```
#wget http://www.securixlive.com/download/barnyard2/barnyard2-1.8.tar.gz
#tar xvf barnyard2-1.8.tar.gz
#cd barnyard2-1.8
```

Ahora vamos a compilarlo de modo que convierta la base de datos de “formato snort” a formato “snort-barnyard”

```
#!/configure --with-mysql
#make
#make install
```

Copiamos el archivo de configuración de barnyard a la carpeta /etc/ de snort, crearemos un directorio y archivo de favoritos para conocer donde pararon o se reiniciaron los eventos de barnyard2 (waldo):

```
#cp etc/barnyard2.conf /usr/local/snort/etc/
#mkdir /var/log/barnyard2
#touch /var/log/snort/barnyard2.waldo
#chown snort.snort /var/log/snort/barnyard2.waldo
```

Ahora vamos a configurar barnyard2 modificando su archivo de configuración (/usr/local/snort/etc/barnyard2.conf) Donde deberíamos de cambiar el valor de las directivas siguientes:

```
config reference_file: /etc/snort/reference.config
config classification_file: /etc/snort/classification.config
config gen_file: /etc/snort/genmsg.map
config sid_file: /etc/snort/sidmsg.map
```

Por estos:

```
config reference_file: /usr/local/snort/etc/reference.config
config classification_file: /usr/local/snort/etc/classification.config
config gen_file: /usr/local/snort/etc/genmsg.map
config sid_file: /usr/local/snort/etc/sidmsg.map
```

Y descomentar las siguientes y modificarlas de la siguiente manera (Estado original:)

```
#config hostname: thor
#config interface: eth0
#output database: log, mysql, user=root password=test dbname=db host=localhost
```

Estado modificado por nosotros:

```
config hostname: localhost
config interface: eth1
output database: log, mysql, user=snort password=contraseña dbname=snort host=localhost
```

Bien, para finalizar vamos a configurar las tarjetas de red, para ello editaremos el archivo /etc/network/interfaces dejándolo así (en el caso mínimo que tengamos 2 tarjetas de red):

```
auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

```
auto eth1
iface eth1 inet manual
up ifconfig eth1 up
```

Nota: el “up ifconfig eth1 up” es para que se levante sola cada inicio de sistema

Como último paso vamos a editar el archivo de configuración de snort indicándole donde lo hemos instalado (dimos otra ruta de instalación que no era la de por defecto)

```
#nano /usr/local/snort/etc/snort.conf
```

Cambiaremos las siguientes líneas:

```
dynamicpreprocessor directory /usr/local/lib/snort_dynamicpreprocessor/
dynamicengine /usr/local/lib/snort_dynamicengine/libs_f_engine.so
dynamicdetection directory /usr/local/lib/snort_dynamicrules
```

a estas:

```
dynamicpreprocessor directory /usr/local/snort/lib/snort_dynamicpreprocessor/
dynamicengine /usr/local/snort/lib/snort_dynamicengine/libs_f_engine.so
dynamicdetection directory /usr/local/snort/lib/snort_dynamicrules
```

Ya hemos finalizado la instalación de snort en nuestro sistema.

Vamos a iniciarlo:

```
# /usr/local/snort/bin/snort -u snort -g snort -c /usr/local/snort/etc/snort.conf -i eth1
```

```
+-----+
[ Number of patterns truncated to 20 bytes: 977 ]

--== Initialization Complete ==--

    ,,_-   -*> Snort! <*-
o"  )~   Version 2.8.6 (Build 38)
    ""   By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team

Copyright (C) 1998-2010 Sourcefire, Inc., et al.
Using PCRE version: 7.8 2008-09-05
```

De este modo hemos iniciado snort como IDS, ya que tenemos más modos de iniciación (sólo snifer “snort -v”...) y empieza a capturar paquetes, podemos ver las alertas de varias formas, desde alguna interfaz web de las muchas que hay (snort report, sam, acidbase...) o sencillamente desde los logs.

```
#cat /var/log/snort/alert
```

```
[**] [1:12286:2] WEB-CLIENT PCRE character class double free overflow attempt [**]
[[Classification: Attempted User Privilege Gain] [Priority: 1]
05/30-19:02:50.990212 174.120.39.58:80 -> 192.168.1.128:54053
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:2800
***AP*** Seq: 0x415C5FDF Ack: 0xF01E4DE3 Win: 0x40B0 TcpLen: 20
[Xref => http://docs.info.apple.com/article.html?artnum=306174][Xref => http://cve.mitre.org/cgi/s
```

Entradas como estas son las que nos encontraremos en nuestro log de snort.

Además podemos guardar el log en formato tcdump, para abrirlo con tcdump y wireshark.

Instalación de SnortReport

Snort report es una interfaz web para snort, que proporciona unas gráficas y unas alertas a tiempo real generadas desde la base de datos mysql por snort.

Así podremos administrar de forma más interpretable nuestros datos de snort e incluso desde una máquina remota (ya que es via web). Bien, para instalarlo tenemos que seguir los siguientes pasos. Descargamos snortreport.

```
#wget http://www.symmetrixtech.com/ids/snortreport-1.3.1.tar.gz
```

```
#tar xzvf snortreport-1.3.1.tar.gz -C /var/www
```

```
#mv /var/www/snortreport-1.3.1 /var/www/snortreport
```

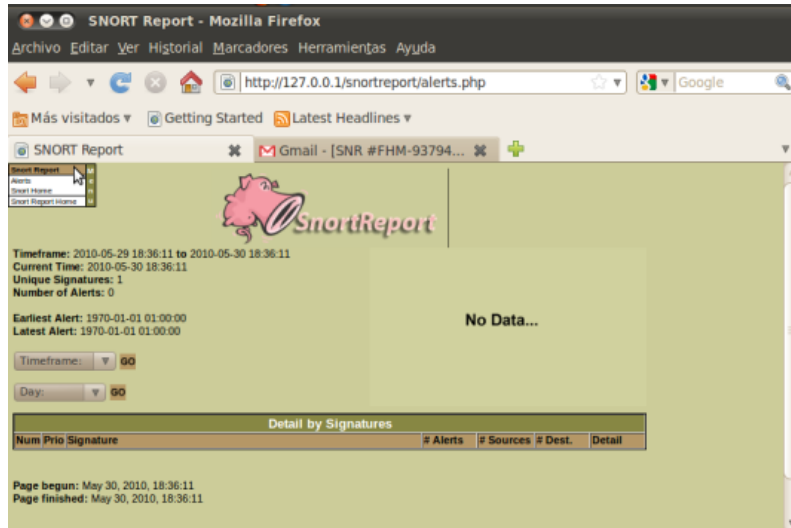
Ahora vamos a configurar nuestro archivo de configuración de snortreport

```
#nano /var/www/snortreport/srconf.php
```


Buscamos esta línea y la cambiamos por la que proceda en nuestro caso:
\$pass = "contraseñadelabasededatos";

Esta es la instalación más simple de SnortReport, se puede configurar de forma que añadamos nmap, nbtscan a la interfaz web, el uso de plugins para crear gráficas y muchas más cosas.

Ya podremos acceder a la interfaz web desde el navegador:



Nota: No he conseguido que meta datos en la base de datos **MySQL**, lo he instalado tanto de los repositorios como compilándolo, en debian y ubuntu y no ha acabado introduciendo datos en la base de datos, sólo en los logs.

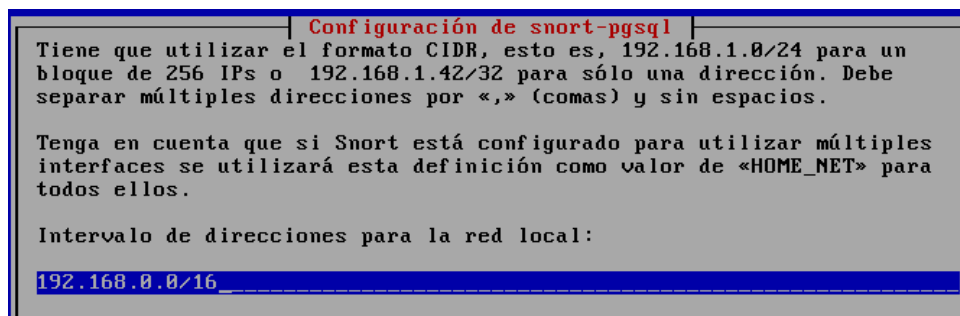
Como no he llegado a saber si el error es mio o algún bug o cualquier problema de algún paquete probé a instalarlo en Debian con postgres y desde apt, que es una instalación totalmente distinta.

Instalación en Debian Lenny desde apt

Voy a ser más breve con la instalación en Debian Lenny, ya que hemos visto lo que vamos a hacer antes. La instalación de snort con soporte para postgres es tan sencilla como:

```
#apt-get install snort-pgsql
```

Nos preguntará a que red queremos monitorizar (por defecto nos pone 192.168.0.0/16)



Una vez definida la red, nos preguntará por nuestra base de datos (que aún no tenemos creada) aceptemos o no nos dará error al final (y aunque la tengamos configurada) así que vamos a aceptar, terminará de instalarse y nos dirá que hubo un error (la instalación de la base de datos)

Vamos a crear y configurar la base de datos en postgres, para ello lo deberíamos tener instalado “#apt-get install postgresql”

```
#su postgres
#$ createdb snort_db
$ zcat /usr/share/doc/snort-pgsql/create_postgresql.gz | psql -d snort_db
$ psql -d snort_db
snort_db=# CREATE USER snort WITH PASSWORD 'password' ;
snort_db=# GRANT ALL ON DATABASE snort_db TO snort ;
snort_db=# \d
(con este comando comprobamos que las tablas están creadas)
snort_db=# GRANT ALL ON TABLE “tablas de snort” TO snort;
snort_db=# \q
```

Con esto la base de datos y el usuario estan creados, ahora vamos a cambiar un parámetro de postres para que no nos de problemas al conectarnos a la base de datos en el archivo /etc/postgresql/8.3/main/pg_hba.conf

Buscamos la línea:

```
host all all 127.0.0.1 255.255.255.255 ident sameuser
```

Y la cambiamos por:

```
host all all 127.0.0.1 255.255.255.255 trust
```

Una vez echo esto, comprobamos que las 2 interfaces de red estan levantadas, y levantamos snort con el siguiente comando:

```
#snort -i eth1 -c /etc/snort/snort.conf
```

```
ecoding Ethernet on interface eth1
atabase: compiled support for ( postgresql )
atabase: configured to use postgresql
atabase: user = snort
atabase: password is set
atabase: database name = snort_db
atabase: host = 127.0.0.1
atabase: sensor name = unknown:eth1
atabase: sensor id = 2
atabase: schema version = 107
atabase: using the "log" facility
```

Entre líneas al iniciarse veremos que se conecta a la base de datos sin problemas.

```

Action Stats:
ALERTS: 36
LOGGED: 36
PASSED: 0
=====
database: Closing connection to database "snort_db"
Snort exiting

```

Para parar snort pulsamos Control + C y veremos si ha capturado algo o no, también comprobamos que cierra la conexión con la base de datos porque ha terminado de loguear.

Desde esta instalación podemos ver las alertas también en /var/log/snort/alerts (como hemos visto arriba)

Podemos cambiar o ver los parámetros de configuración por defecto de snort en /etc/default/snort, como son usuario, grupo, parámetros de iniciación de modo demonio, ruta de logs..

Otros archivos de log en formato tcpdump son los que guarda snort de cada sesión de escaneo, a los que asigna un nombre así:

```

debiansnort:~# ls /var/log/snort/
alert                                tcpdump.log.1276969723  tcpdump.log.1276970262
tcpdump.log.1276969332  tcpdump.log.1276970215  tcpdump.log.1276970285
tcpdump.log.1276969474  tcpdump.log.1276970238  tcpdump.log.1276970466

```

Podemos ver el contenido de los mismos con wireshark o tcpdump de la siguiente forma:

\$tcpdump -tttt -X -r /var/log/snort/tcpdump.log.1276969332

```

debiansnort:~# tcpdump -tttt -X -r /var/log/snort/tcpdump.log.1276969332
reading from file /var/log/snort/tcpdump.log.1276969332, link-type EN10MB (Ethernet)
2010-06-19 19:43:56.282848 IP 192.168.1.1.domain > 192.168.1.128.60766: 54920 1/0/0 A 66.35.45.157 (46)
    0x0000: 4500 004a 0000 4000 fa11 fcd0 c0a8 0101  E..J..@.....
    0x0010: c0a8 0180 0035 ed5e 0036 dfdb d688 8180  ....5.^.6.....
    0x0020: 0001 0001 0000 0000 0369 7363 0473 616e  ....isc.san
    0x0030: 7303 6f72 6700 0001 0001 c00c 0001 0001  s.org.....
    0x0040: 0000 003c 0004 4223 2d9d                ...<..B#-.

```

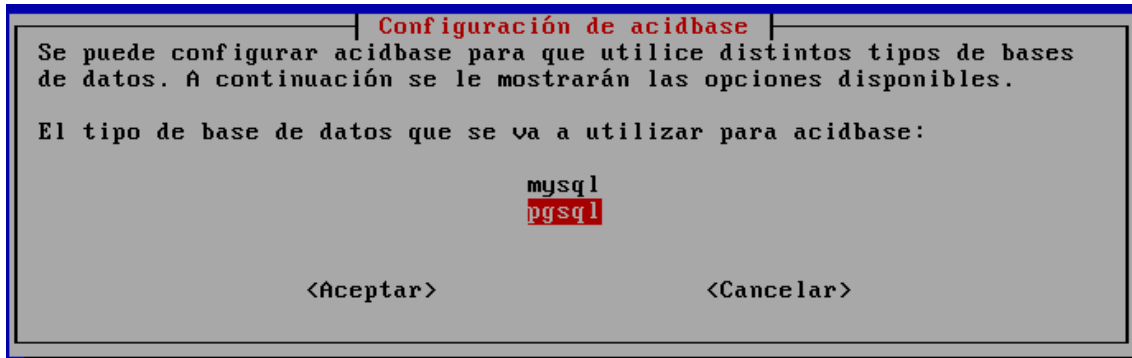
Tabla	Dueño	Tablespace	Estimación de filas
<input type="checkbox"/> data	postgres		61
<input type="checkbox"/> detail	postgres		0
<input type="checkbox"/> encoding	postgres		0
<input type="checkbox"/> event	postgres		61
<input type="checkbox"/> icmp_hdr	postgres		0
<input type="checkbox"/> ip_hdr	postgres		61
<input type="checkbox"/> opt	postgres		0
<input type="checkbox"/> reference	postgres		0
<input type="checkbox"/> reference_system	postgres		0
<input type="checkbox"/> schema	postgres		0
<input type="checkbox"/> sensor	postgres		0
<input type="checkbox"/> sig_class	postgres		0
<input type="checkbox"/> sig_reference	postgres		0
<input type="checkbox"/> signature	postgres		0
<input type="checkbox"/> tcp_hdr	postgres		0
<input type="checkbox"/> udp_hdr	postgres		60

Desde phppgadmin comprobamos que hay registros en la base de datos. Ahora para ver esos registros necesitaremos una interfaz web.

Como interfaz gráfica he elegido otra distinta a snortreport, esta es acidbase, la podemos instalar con:

```
#apt-get install acidbase
```

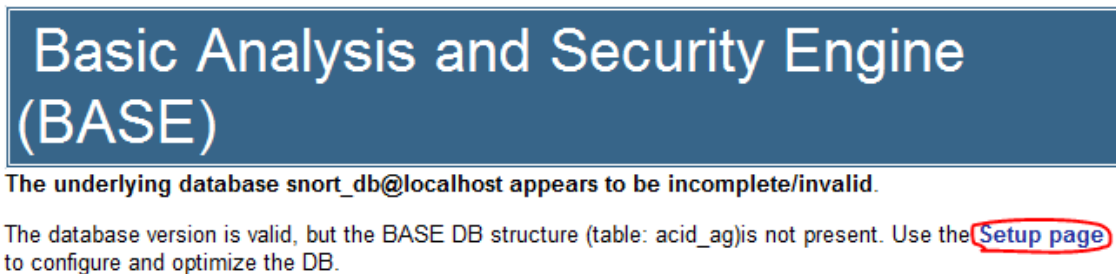
Esta instalación nos instalará apache, php, y todo lo necesario para que funcione, durante la instalación nos preguntará sobre el tipo de la base de datos, ahí seleccionamos pgsql (postgresql) y terminamos la instalación.



Editaremos el archivo `/etc/acidbase/database.php` con nuestros datos de postgres:

```
$alert_user='snort';  
$alert_password='admin';  
$basepath='';  
$alert_dbname='snort_db';  
$alert_host='localhost';  
$alert_port='5432';  
$DBtype='pgsql';
```

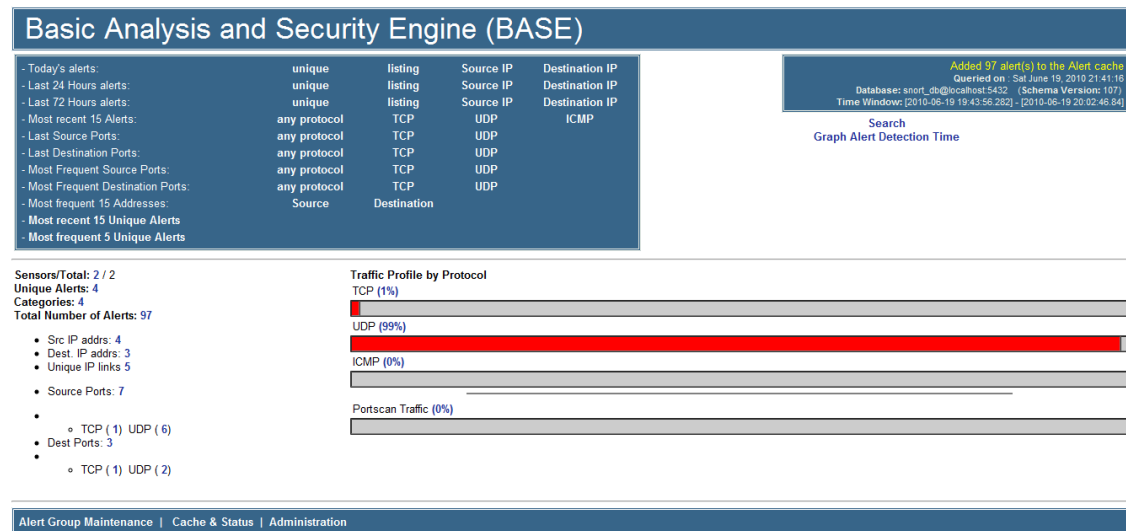
Y iremos al navegador y abrimos nuestro servidor web para comenzar la instalación:



En la página de instalación veremos un botón para crear las tablas de acid, pulsamos y se crearán con nuestras tablas de snort.

```
Successfully created 'acid_ag'  
Successfully created 'acid_ag_alert'  
Successfully created 'acid_ip_cache'  
Successfully created 'acid_event'  
Successfully created 'base_roles'  
Successfully INSERTED Admin role  
Successfully INSERTED Authenticated User role  
Successfully INSERTED Anonymous User role  
Successfully INSERTED Alert Group Editor role  
Successfully created 'base_users'
```

Pulsamos en el enlace de ir a la aplicación (situado al final de la página de instalación) y veremos una pantalla como esta:



Donde encontraremos el análisis de alertas que hayamos capturado con snort.

ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
#0-(1-1)	[local] [snort] DNS SPOOF query response with TTL of 1 min. and no authority	2010-06-19 19:43:56.282	192.168.1.1:53	192.168.1.128:80766	UDP
#1-(2-1)	[cve] [local] [bugtraq] [local] [snort] COMMUNITY WEB-MISC mod_jrun overflow attempt	2010-06-19 19:46:46.211	192.168.1.128:33963	65.54.166.219:80	TCP
#2-(2-2)	[local] [snort] SCAN UPnP service discover attempt	2010-06-19 19:47:39.303	192.168.1.128:59210	239.255.255.250:1900	UDP
#3-(2-3)	[local] [cve] [local] [cve] [local] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	2010-06-19 19:47:40.564	192.168.1.128:1900	239.255.255.250:1900	UDP
#4-(2-4)	[local] [snort] SCAN UPnP service discover attempt	2010-06-19 19:47:40.725	192.168.1.128:50864	239.255.255.250:1900	UDP
#5-(2-5)	[local] [snort] SCAN UPnP service discover attempt	2010-06-19 19:47:40.75	192.168.1.128:50864	239.255.255.250:1900	UDP
#6-(2-6)	[local] [snort] SCAN UPnP service discover attempt	2010-06-19 19:47:40.775	192.168.1.128:50864	239.255.255.250:1900	UDP
#7-(2-7)	[local] [cve] [local] [cve] [local] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	2010-06-19 19:47:40.859	192.168.1.128:1900	239.255.255.250:1900	UDP
#8-(2-8)	[local] [cve] [local] [cve] [local] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	2010-06-19 19:47:40.992	192.168.1.128:1900	239.255.255.250:1900	UDP
#9-(2-9)	[local] [cve] [local] [cve] [local] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	2010-06-19 19:47:41.25	192.168.1.128:1900	239.255.255.250:1900	UDP
#10-(2-10)	[local] [cve] [local] [cve] [local] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	2010-06-19 19:47:41.698	192.168.1.128:1900	239.255.255.250:1900	UDP
#11-(2-11)	[local] [snort] SCAN UPnP service discover attempt	2010-06-19 19:47:42.325	192.168.1.128:50864	239.255.255.250:1900	UDP
#12-(2-12)	[local] [cve] [local] [cve] [local] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	2010-06-19 19:47:42.927	192.168.1.128:1900	239.255.255.250:1900	UDP
#13-(2-13)	[local] [cve] [local] [cve] [local] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	2010-06-19 19:47:42.996	169.254.80.11:1900	239.255.255.250:1900	UDP
#14-(2-14)	[local] [cve] [local] [cve] [local] [bugtraq] [local] [snort] MISC UPnP malformed advertisement	2010-06-19 19:47:43.609	192.168.1.128:1900	239.255.255.250:1900	UDP

Se puede filtrar por ip origen, por dia, hora, fecha, etc.. Si entramos en una configuración más precisa de acidbase podríamos hacer que nos enviara alertas por email, archivarlas, y más cosas que no vamos a tratar aquí.

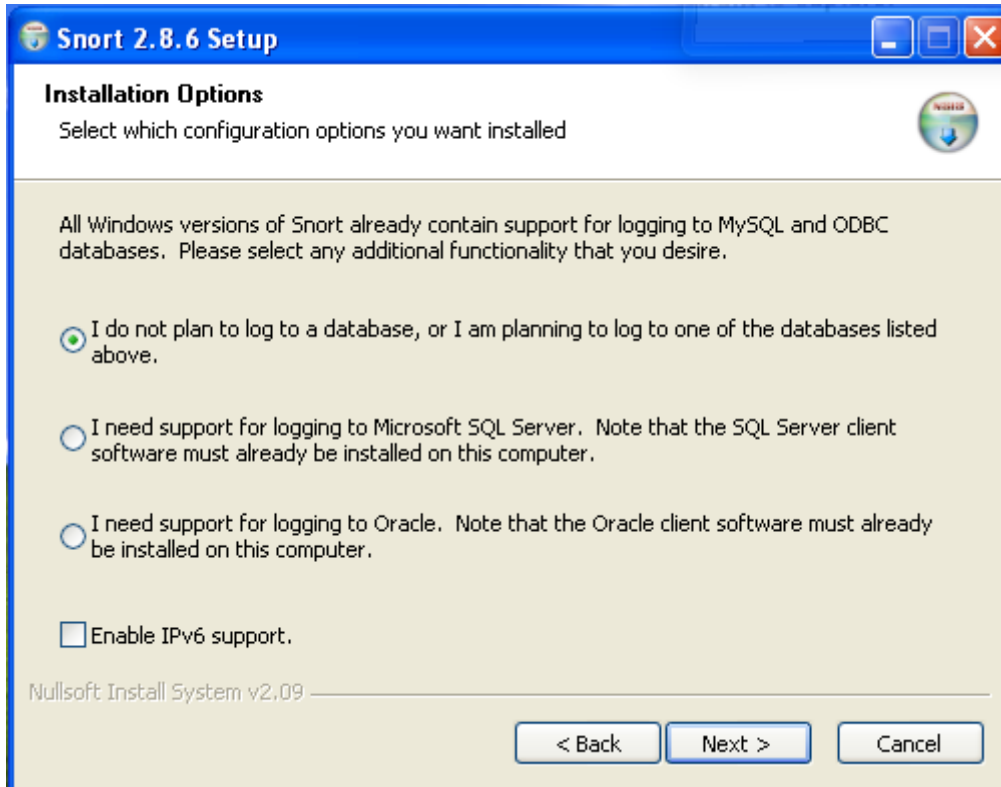
Instalación en Windows

Tan sólo tenemos que ir a la página de descargas de snort (<http://www.snort.org/downloads>) y descargar el ejecutable para windows (.exe)

Binaries

File	SIG	MD5	Release Notes	Last Modified
snort-2.8.6-1.F11.i386.rpm	SIG	MD5	README	26 Apr, 2010
Snort 2.8.6 Installer.exe	SIG	MD5	README	26 Apr, 2010
snort-postgresql-2.8.6-1.F11.i386.rpm	SIG	MD5	README	26 Apr, 2010
snort-postgresql-2.8.6-1.F11.i386.rpm	SIG	MD5	README	26 Apr, 2010

El siguiente paso será aceptar el acuerdo de licencia del programa, luego el mismo asistente nos preguntará si tenemos intención de loguear los datos en una base de datos, ya que Snort para windows trae soporte para MySQL y PostGres por defecto, pero no para MSSQL ni para Oracle, aquí habría que definir que queremos nosotros (Cómo vemos trae soporte para Ipv6):



Tras esto debemos instalar los drivers winpcap (<http://www.winpcap.org>) y obviamente tener nuestro SGBD instalado previamente si fuéramos a loguear los datos en alguno.

Ahora vamos a descargarnos las reglas de snort y las copiaremos en *C:/Snort/rules* (O donde hayamos instalado snort) . Ya vimos como de donde descargar las reglas arriba.



De las reglas que acabamos de descargar, copiaremos el archivo */etc/snort.conf* y sustituiremos el que nos dejó la instalación por defecto (éste último viene más actualizado).

Ahora abrimos la consola de comandos de windows (Menú ejecutar - escribimos "cmd") y vamos al directorio de snort\bin\ y ejecutamos snort con el parámetro -W, el cual nos dará como salida las interfaces de red de la máquina y su número con el cual la identificaremos.

```
C:\Snort\bin>snort.exe -W

o"~>~
'''~>~

eam

-*> Snort! <*-
Version 2.8.6-ODBC-MySQL-FlexRESP-WIN32 GRE <Build 38>
By Martin Roesch & The Snort Team: http://www.snort.org/snort-snort-t
eam
Copyright (C) 1998-2010 Sourcefire, Inc., et al.
Using PCRE version: 7.4 2007-09-21
Using ZLIB version: 1.2.3

Interface  Device                                     Description
-----
1  \Device\NPF_{68A5FC4B-6DAF-4572-8AC5-1CF89F9D293B}  AMD PCNET Family
Ethernet Adapter (Microsoft's Packet Scheduler)
```

Podríamos probarla porejemplo de la siguiente forma:

```
>snort.exe -dev -i 1
```

Para una completa instalación en windows tenemos que modificar el archivo snort.conf en la carpeta etc de la instalación del mismo. Ahora voy a explicar la estructura del archivo snort.conf e indicaré los cambios a realizar para una instalación en windows.

Finalmente para ejecutar snort en modo IDS en windows lo haremos con un comando como este:

```
>snort -c C:\Snort\etc\snort.conf -l c:\snort\log -i 1
```

*siempre que tengas snort instalado en esa ruta y tu interfaz de detección sea la número 1.

Archivo snort.conf

Partes del archivo de configuración de snort:

1) Definición de variables de red.

En esta parte se definirán las variables de red para una configuración lo más exacta y correcta de snort. Se definen mediante variables del tipo "var HOME_NET any" donde any es el valor de la red que quieres securizar (p.e. var HOME_NET 192.168.1.0/24). Como estas hay algunas más, que sobre todo conviene a definir si tienes servicios en esa red, para no crear falsas alertas.

Te permite definir lista de servidores DNS, HTTP, SMT, SQL, Telnet, SSH.... red externa, etc..

2) Configurar el decodizador.

En esta parte configuramos el decodizador, éste se encarga de definir a que protocolo pertenece el paquete que esta analizando y guarda el paquete de forma que pueda pasar un

mejor escaneo el preprocesador y el sistema de detección de snort. También se encarga de detectar posibles errores en el paquete (ya que solo lee el encabezado de los mismos).

Con la configuración del decodizador logramos afinar las alertas de snort si queremos, ya que no nos avisará de paquetes rotos, paquetes mal formados, obsoletos, etc..

Siempre podemos leer la documentación donde están todas las posibles configuraciones del decodizador en README.decoder.

3) Configurar el sistema de detección.

En esta parte se puede configurar el sistema de detección de snort, no requiere ningún cambio para la instalación en windows. Son variables que definen la exactitud con la que snort juzgará los paquetes, las más recomendables son las que vienen por defecto.

4) Configuración de librerías dinámicas.

En esta parte configuramos las librerías que trabajaran con snort, como podemos ver esas librerías vienen incluidas con el mismo snort, en formato .dll para windows y formato .so para linux, por defecto en el fichero de configuración de snort están las rutas para linux (en la versión de windows también) por eso en el caso en que la instalación sea en windows debermos cambiar las rutas de estilo:

```
dynamicpreprocessor directory /usr/local/lib/snort_dynamicpreprocessor/  
dynamicengine /usr/local/lib/snort_dynamicengine/libsf_engine.so
```

por otras así:

```
dynamicpreprocessor directory C:\Snort\lib\snort_dynamicpreprocessor\  
dynamicengine C:\Snort\lib\snort_dynamicengine\libsf_engine.dll
```

5) Configurar preprocesadores

Los preprocesadores son unas herramientas añadidas que usa snort para desarmar, analizar y rearmar los paquetes que le llegan del decodificador. No tiene cambios con la instalación en windows y se recomienda su modificación para personalizaciones concretas.

6) Configurar plugins de salida

En esta parte encontraremos todos los plugins que soporta snort para la salida de registros en sus capturas, como ya hemos visto arriba esos plugins pueden ser archivos en formato tcpdump, el log en formato texto simple, en una base de datos, etc..

Un tipo de salida que no viene por defecto en la instalación de windows es de log en formato de texto simple, para crearla basta con añadir en cualquier línea de esa parte del archivo esto:

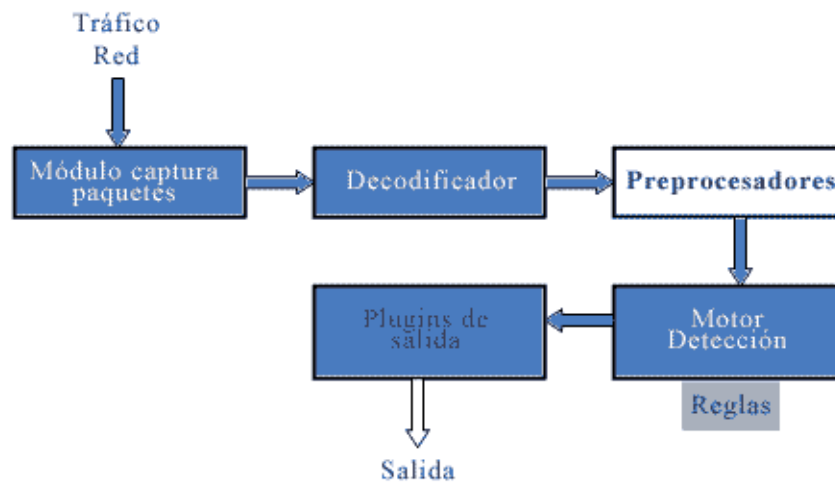
Output fast_alert: snortlog.log

Y crear el archivo snortlog.log en la carpeta “log” de nuestra instalación de snort.

7) Personalizar tu conjunto de reglas

En esta parte parte del archivo se incluyen todas las reglas que antes hemos descargador de snort, podemos personalizar quitando las que no queramos (y así aligerar trabajo a la máquina) in cluso podemos añadir más.

Básicamente el trabajo que hace snort podemos resumirlo en este esquema, que entenderemos mejor despues de la instalación y la explicación de la estructura del archivo de configuración:



Conclusiones

Cuando hice la primera parte del proyecto, con ossec, lo encontré muchísimo más sencillo de lo que creía, nada que ver al final con la de snort (Muchísimos más problemas). Pero al fin y al cabo me ha parecido un buen tema del que hacer el proyecto, que puede ser muy útil en algunos sitios y al menos no me quedo con las ganas de saber que son los IDS, que hacen y algunas cosas más sobre ellos que he aprendido, aunque soy consciente de las muchas cosas que no he aprendido y están ahí sobre el tema.

Por otro lado me parece increíble la cantidad de información que hay de estos temas pero siendo la mayoría actualmente desactualizada, muchas “posibles” soluciones que no valían, etc. Pero si veo recomendable instalar sistemas de detección de intrusos y mantenerlos si es necesario un cierto nivel de seguridad.

Documentación

<https://forums.snort.org/forums>

<http://www.symmetrixtech.com/articles/004-snortinstallguide286.pdf>

<http://www.ossec.net/main/documentation/>

<http://www.linuca.org/body.phtml?nIdNoticia=13>

<http://www.linux.ebre.cat/2009/11/detectar-intrusos-instalar-snort-debian.html>

<http://www.internetsecurityguru.com/documents/>

[http://www.debian-administration.org/article/Using the 'snort' Intrusion Detection System](http://www.debian-administration.org/article/Using_the_'snort'_Intrusion_Detection_System)

<http://seguridadyredes.nireblog.com/post/2009/03/03/snort-preprocesadores-i-parte>
