Instalar y configurar Snort 2.9.9.0 en Ubuntu 17.04 (Zesty Zapus) con Barnyard2, PulledPork y Snorby.

Este manual se ha realizado siguiendo la guía disponible en:

http://www.ubuntu-howtodoit.com/?p=138

Vamos a instalar y configurar Snort en una máquina virtual con VMWare:

- Seleccionar el adaptador de red de tipo VMXNET 3 al crear el sistema invitado.
- Despues de instalar Ubuntu, instalar VMware Tools en el sistema invitado.

Los dos puntos anteriores son opcionales, pero se deben realizar en caso de implementar el punto 8.

Para la instalación y configuración de Snort y Snorby vamos a instalar y configurar los siguientes componentes.

- **Snort v2.9.9.0** – Es un sensor de red que se encarga de supervisar el tráfico de red en crudo y generar las alertas comparando el tráfico de red con unas reglas de detección predefinidas.
- **PulledPork v0.7.3** – Componente encargado de actualizar de forma periódica las reglas de detección de Snort.
- **Barnyard2 v2.1.14 (Build 337)** – Procesa las alertas generadas por Snort y las adapta a un formato de base de datos, para poder interpretarlas posteriormente.
- **Snorby v2.6.3** – Aplicación web que interpreta la información generada por Barnyard2 en la base de datos y la presenta de forma visual, permitiendo acceder a los eventos de una forma sencilla. Muestra gráficos y estadísticas.

1. – Actualizar y configurar Ubuntu 17.04.

1.1 – Instalar las actualizaciónes del sistema:

La instalación del sistema operativo no se detalla en esta guía, ya que se sale del objetivo de esta.

Una vez instalado el sistema operativo, instalar las actualizaciónes e instalar openssh-server:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install openssh-server
```

Reiniciar el sistema si es necesario.

1.2 – Configurar la red.

A partir de Ubuntu 15.10, los nombres de las interfaces de red siguen la especificación Predictable Network Interfaces. Para saber el nombre de la interfaz de red, ejecutar en el Terminal el comando *ifconfig*.

Es recomendable deshabilitar LRO (Large Receive Offload) y GRO (Generic Receive Offload) en todas las interfaces de red donde Snort va a estar a la escucha. Para mas información sobre esto, leer la sección 1.5 Packet Acquisition del manual oficial de Snort.

Editar el fichero de configuración de interfaces:

```
sudo vim /etc/network/interfaces
```

Añadir las siguientes líneas a cada interfaz de red que se vaya a monitorizar, en este caso la interfaz de red es *ens33*:

```
post-up ethtool -K ens33 gro off
post-up ethtool -K ens33 lro off
```

Reiniciar el sistema y comprobar que LRO y GRO están desabilitados:

```
ethtool -k ens33 | grep receive-offload
```

La salida tiene que ser algo similar a lo siguiente:

```
generic-receive-offload: off
large-receive-offload: off
```

2. – Instalar y configurar Snort.

2.1 – Instalar los prerrequisitos de Snort.

Instalar librerías y prerrequisitos necesarios:

```
sudo apt-get install build-essential -y
sudo apt-get install libpcap-dev libpcre3-dev libdumbnet-dev -y
sudo apt-get install zlib1g-dev liblzma-dev openssl libssl-dev -y
sudo apt-get install bison flex -y
```

Descargar, compilar e instalar DAQ (Librerias de Aquisición de datos).

```
mkdir ~/snort_src
cd ~/snort_src/
wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
tar -zxvf daq-2.0.6.tar.gz
cd daq-2.0.6/
./configure
make
sudo make install
```

2.2 – Instalar Snort (Descargar, compilar e instalar).

Los siguientes comandos descargan, compilan e instalan Snort 2.9.9.0:

```
cd ~/snort_src/
wget https://www.snort.org/downloads/snort/snort-2.9.9.0.tar.gz
tar -zxvf snort-2.9.9.0.tar.gz
cd snort-2.9.9.0
./configure --enable-sourcefire
make
sudo make install
```

Actualizar las librerías compartidas. Si no se hace Snort da errores al ejecutarlo:

```
sudo ldconfig
```

Crear un enlace simbolico al binario de Snort, para poder ejecutarlo sin indicar la ruta:

```
sudo ln -s /usr/local/bin/snort /usr/sbin/snort
```

Para verificar que Snort se ha instalado correctamente se lanza con la opción -V:

```
snort -V
```

La salida del comando anterior es similar a lo siguiente:

```
   , ,_     -*> Snort! <*-
  o"  )~   Version 2.9.9.0 GRE (Build 56)
   ''  ''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
             Copyright (C) 2014-2016 Cisco and/or its affiliates. All rights reserved.
             Copyright (C) 1998-2013 Sourcefire, Inc., et al.
             Using libpcap version 1.8.1
             Using PCRE version: 8.39 2016-06-14
             Using ZLIB version: 1.2.11
```

2.3 – Configurar Snort.

Por motivos de seguridad, Snort no debe ejecutarse como root, así que creamos un usuario normal y un grupo para ejecutar el demonio snort:

```
sudo groupadd snort
sudo useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort
```

2.3.1 – Crear archivos y carpetas necesarios.

Crear directorios requeridos por Snort:

```
sudo mkdir /etc/snort
sudo mkdir /etc/snort/rules
sudo mkdir /etc/snort/rules/iplists
sudo mkdir /etc/snort/preproc_rules
sudo mkdir /etc/snort/so_rules
sudo mkdir /usr/local/lib/snort_dynamicrules
sudo mkdir /var/log/snort
sudo mkdir /var/log/snort/archived_logs
```

Crear ficheros requeridos por Snort:

```
sudo touch /etc/snort/rules/iplists/black_list.rules
sudo touch /etc/snort/rules/iplists/white_list.rules
sudo touch /etc/snort/rules/local.rules
sudo touch /etc/snort/sid-msg.map
```

2.3.2 – Establecer permisos necesarios en ficheros y carpetas.

Ajustar permisos en carpetas y ficheros:

```
sudo chmod -R 5775 /etc/snort
sudo chmod -R 5775 /var/log/snort
sudo chmod -R 5775 /usr/local/lib/snort_dynamicrules
```

Establecer propietario en carpetas y ficheros:

```
sudo chown -R snort:snort /etc/snort
sudo chown -R snort:snort /var/log/snort
sudo chown -R snort:snort /usr/local/lib/snort_dynamicrules
```

2.3.3 – Copiar archivos de configuración básica.

Copiar los archivos de configuración y prepocesadores dinámicos predefinidos:

```
cd ~/snort_src/snort-2.9.9.0/etc/
sudo cp *.conf* /etc/snort
sudo cp *.map /etc/snort
sudo cp *.dtd /etc/snort
cd ~/snort_src/snort-2.9.9.0/src/dynamic-preprocessors/build/usr/local/lib/snort_dynamicpreprocessor/
sudo cp * /usr/local/lib/snort_dynamicpreprocessor/
```

La estructura de directorios en este punto es la siguiente:

RUTA	DESCRIPCIÓN
/etc/Snort/	Directorio principal de Snort.
/etc/snort/rules/ /etc/snort/so_rules/ /etc/snort/preproc_rules/ /usr/local/lib/snort_dynamicrules/	Directorios de reglas.
/etc/snort/rules/iplists/	Directorio de listas de IP's.
/usr/local/lib/snort_dynamicpreprocessor/	Preprocesadores dinámicos de Snort.
/var/log/Snort/	Directorio para almacenar los logs y alertas.

El árbol de archivos y directorios es similar al siguiente:

```
~# tree /etc/snort
/etc/snort
├── attribute_table.dtd
├── classification.config
├── file_magic.conf
├── gen-msg.map
├── preproc_rules
├── reference.config
├── rules
│   ├── local.rules
│   ├── iplists
│   │   ├── black_list.rules
│   │   └── white_list.rules
├── sid-msg.map
```

```
├── snort.conf
├── so_rules
├── threshold.conf
└── unicode.map
```

2.3.4 – Editar el fichero de configuración de Snort (snort.conf).

El fichero de configuración de Snort se encuentra en la ruta */etc/snort/snort.conf*. Este fichero contiene la configuración y ajustes necesarios para utilizar Snort en modo NIDS.

snort.conf references several configuration files and since we are going to use **PulledPork** to manage the rulesets (**PulledPork** combines all rules into a single file) we need to comment out those references.

Primero vamos a desactivar todos los conjuntos de predefinidos, más adelante se configurarán según nuestras necesidades.

Desactivar todos los conjuntos de reglas predefinidos:

```
sudo sed -i "s/include \$RULE\_PATH/#include \$RULE\_PATH/" /etc/snort/snort.conf
```

Editar el fichero snort.conf manualmente:

```
sudo vim /etc/snort/snort.conf
```

Para configurar la red que se quiere monitorizar, ir a la línea **45** y modificar:

```
ipvar HOME_NET any
```

Por (especifar el rango de red necesario según el caso):

```
ipvar HOME_NET 192.168.0.0/24
```

Algunos manuales recomiendan establecer en la línea **48** el parámetro **EXTERNAL_NET** con el valor **!$HOME NET**. Esto no es recomendable, ya que en muchas ocasiones hace que Snort no detecte las alertas.

Parar evitar posibles errores, se modifican las rutas relativas de archivos y carpetas referenciados y se establecen rutas absolutas. Ir hasta la línea **104** y modificar los siguientes valores:

```
var RULE_PATH ../rules
var SO_RULE_PATH ../so_rules
var PREPROC_RULE_PATH ../preproc_rules

var WHITE_LIST_PATH ../rules
var BLACK_LIST_PATH ../rules
```

Por estos:

```
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules

var WHITE_LIST_PATH /etc/snort/rules/iplists
```

```
var BLACK_LIST_PATH /etc/snort/rules/iplists
```

Ir hasta la línea **545** y descomentarla para activar las reglas locales (local.rules). Estas se utilizarán mas adelante para testear Snort. En este punto aprovechamos para activar los archivos de reglas que sean necesarios (scan.rules, etc.).

```
include $RULE_PATH/local.rules
include $RULE_PATH/scan.rules
```

Con el siguiente comando, comprobamos si la configuración del fichero snort.conf es correcta:

```
sudo snort -T -i ens33 -c /etc/snort/snort.conf
```

Si todo funciona correctamente, las dos ultimas líneas de la salida del comando anterior, seran similares a lo siguente:

```
...
Snort successfully validated the configuration!
Snort exiting
```

2.3.5 – Verificar y testear el funcionamiento de Snort.

Como se han desactivado todas las referencias a ficheros de reglas en el fichero */etc/snort/snort.conf*, y los que se han añadido están vacios, en este momento no existe ninguna regla activa.

Esto se puede comprobar en la salida del comando anterior, esta es la zona del log donde indica las reglas activas:

```
+-------------------[Rule Port Counts]---------------------------------------
|         tcp   udp   icmp   ip
| src     0     0     0      0
| dst     0     0     0      0
| any     0     0     0      0
| nc      0     0     0      0
| s+d     0     0     0      0
+------------------------------------------------------------------------------
```

A continuación, vamos a crear manualmente una regla para testear Snort.

Editar el fichero **local.rules**:

```
sudo vim /etc/snort/rules/local.rules
```

Y añadir una regla, insertando la siguiente linea:

```
alert icmp any any -> $HOME_NET any (msg:"ICMP test detected"; GID:1; sid:10000001; rev:001; classtype:icmp-event;)
```

Editar el fichero que contiene los identificadores (sid) de las reglas **sid-msg.map**:

```
sudo vim /etc/snort/sid-msg.map
```

Y añadir un sid para la regla anterior, para ello insertar la línea siguiente:

```
1 || 10000001 || 001 || icmp-event || 0 || ICMP Test detected || url,tools.ietf.org/html/rfc792
```

Esta regla hace que Snort genere una alerta al detectar un paquete ICMP, como puede ser “Echo request” o “Echo reply” al realizar un ping sobre un host de la red interna.

Una vez modificada la configuración, volvemos a verificar que ésta es correcta:

```
sudo snort -T -c /etc/snort/snort.conf -i ens33
```

Observando el log del comando anterior debemos obtener una salida similar a la siguiente, donde se puede observar que esta activada la regla creada:

```
++++++++++++++++++++++++++++++++++++++++++++++++++
Initializing rule chains...
1 Snort rules read
   1 detection rules
   0 decoder rules
   0 preprocessor rules
1 Option Chains linked into 1 Chain Headers
0 Dynamic rules
++++++++++++++++++++++++++++++++++++++++++++++++++

+-------------------[Rule Port Counts]-----------------------------------
|        tcp   udp   icmp   ip
| src     0     0      0     0
| dst     0     0      0     0
| any     0     0      1     0
| nc      0     0      1     0
| s+d     0     0      0     0
+----------------------------------------------------------------------------
```

Ahora vamos a iniciar Snort en modo NIDS desde la línea de comandos y ver las alertas detectadas en la consola.

Para ello, utilizaremos las siguientes opciones:

OPCIÓN	DESCRIPCIÓN
-A console	Imprime las alertas en consola en el momento que son detectadas.
-q	Modo silencioso. No mostrar el banner y el informe de estado.
-u snort	Iniciar Snort con el usuario especificado.
-g snort	Iniciar Snort con el grupo especificado.
-c /etc/snort/snort.conf	Ruta del fichero de configuración a cargar.
-i ens33	Nombre de la interfaz en la que va a escuchar y monitorizar.

Iniciar Snort con la siguiente línea de comandos:

```
sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
```

En este punto, Snort esta ejecutándose y escuchando en la interfaz **ens33**.

Al realizar un ping desde cualquier ordenador de la red, Snort debe detectarlo y obtendremos una salida similar a la siguiente:

```
04/04-10:15:04.875655 [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.44 -> 192.168.0.10
04/04-10:15:04.875774 [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.10 -> 192.168.0.44
04/04-10:15:05.876799 [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.44 -> 192.168.0.10
04/04-10:15:05.876849 [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.10 -> 192.168.0.44
04/04-10:15:06.875872 [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.44 -> 192.168.0.10
```

Si obtenemos una salida similar a la anterior, significa que Snort esta ejecutándose en modo NIDS y generando las alertas correctamente.

Para detener Snort pulsar **CTRL+C**.

La información con las alertas detectadas se guarda en **/var/log/snort/snort.log.xxxxxxxxxx**, siendo **xxxxxxxxxx** diferente en cada ejecución y cambiando cuando el archivo alcanza cierto tamaño.

3. – Instalar Barnyard2.

Barnyard2 es un programa de código abierto que interpreta archivos binarios generados por Snort.

La salida de los eventos detectados por Snort se pueden guardar en formato binario en una carpeta, Barnyard2 lee estos ficheros, los interpreta y escribe el resultado en una base de datos MySQL. De esta forma se pueden visualizar, buscar y crear perfiles de eventos, realizar estadísticas, etc.

3.1 – Instalar los prerrequisitos de Barnyard2.

Intalación de prerrequisitos de Barnyard2:

```
sudo apt-get install mysql-server libmysqlclient-dev mysql-client autoconf libtool -y
```

Configurar Snort para que guarde los eventos en formato binario. Editar *snort.conf*:

```
sudo vim /etc/snort/snort.conf
```

Ir a la linea **521**:

```
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
```

Y añadir una nueva línea con lo siguiente:

```
output unified2: filename snort.u2, limit 128
```

3.2 – Descargar e instalar Barnyard2.

Los siguientes comandos descargan, compilan e instalan Barnyard2:

```
cd ~/snort_src/
git clone git://github.com/firnsy/barnyard2.git
```

```
cd barnyard2/
autoreconf -fvi -I ./m4
sudo ln -s /usr/include/dumbnet.h /usr/include/dnet.h
sudo ldconfig
./configure --with-mysql --with-mysql-libraries=/usr/lib/x86_64-linux-gnu
make
sudo make install
```

3.3 – Crear los archivos y carpetas necesarios.

```
sudo cp etc/barnyard2.conf /etc/snort
sudo mkdir /var/log/barnyard2
sudo chown snort.snort /var/log/barnyard2
sudo touch /var/log/snort/barnyard2.waldo
sudo chown snort.snort /var/log/snort/barnyard2.waldo
```

3.4 – Crear la base de datos MySQL para almacenar los eventos.

```
mysql -u root -p
create database snort;
use snort
source ~/snort_src/barnyard2/schemas/create_mysql
CREATE USER 'snort'@'localhost'IDENTIFIED BY '***PASSWORD***';
grant create, insert, select, delete, update on snort.* to 'snort'@'localhost';
exit
```

3.5 – Configurar Barnyard2 para usar la base de datos MySQL snort.

Editar el archivo barnyard2.conf

```
sudo vim /etc/snort/barnyard2.conf
```

Añadir la línea siguiente al final del fichero:

```
output database: log, mysql, user=snort password=PASSWORD dbname=snort host=localhost
```

Modificar los permisos para que otros usuarios no puedan leer el fichero *barniard2.conf*, evitando que puedan leer la contraseña de la base de datos:

```
sudo chmod o-r /etc/snort/barnyard2.conf
```

3.6 – Verificar la configuración de Barnyard2.

Para verificar la configuración hay que comprobar tres puntos:

- Que Snort escribe los eventos en el fichero binario de alertas correctamente.
- Que Barnyard2 puede leer los ficheros binarios de Snort.
- Que Barnyard escribe los eventos en la base de datos MySQL.

Ejecutar Snort en modo NIDS de alerta, quitar el parámetro **-A console** de la línea de comandos para que no muestre la salida por consola.

```
sudo /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
```

Lanzar un ping durante unos segundos sobre un equipo de la red a monitorizar, después, detener Snort pulsando **CTRL+C**.

Esto debe generar un fichero **snort.u2.xxxxxxxxxx** en la carpeta **/var/log/snort/**.

Al lanzar Barnyard2, este debe leer los eventos del archivo **snort.u2.xxxxxxxxxx** y cargarlos en la base de datos.

Para lanzar Barniard2 se van a utilizar las siguientes opciones:

OPTION	DESCRIPTION
-c /etc/snort/barnyard2.conf	Ruta del fichero de configuracion de Barnyard, barnyard2.conf.
-d /var/log/snort	Carpeta donde buscar los archivos de salida de Snort.
-f snort.u2	Prefijo del nombre de los ficheros a buscar en la carpeta anterior (snort.u2.nnnnnnnnnn).
-w /var/log/snort/barnyard2.waldo	Ruta del fichero waldo (fichero de marcadores).
-u snort	Ejecutar Barnyard2 con el usuario especificado.
-g snort	Ejecutar Barnyard2 con el grupo especificado.

Ejecutar el comando:

```
sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w
/var/log/snort/barnyard2.waldo -g snort -u snort
```

La salida debe ser similar a lo siguiente:

```
_______    -*> Barnyard2 <*-
/ ,,_  \   Version 2.1.14 (Build 336)
|o"  )~|   By Ian Firns (SecurixLive): http://www.securixlive.com/
+ '''' +  (C) Copyright 2008-2013 Ian Firns <firnsy@securixlive.com>

WARNING: Ignoring corrupt/truncated waldofile '/var/log/snort/barnyard2.waldo'
Opened spool file '/var/log/snort/snort.u2.1459758639'
04/04-10:30:59.918938 [**] [1:10000001:1] Snort Alert [1:10000001:1] [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.44 -> 192.168.0.10
INFO [dbProcessSignatureInformation()]: [Event: 1] with [gid: 1] [sid: 10000001] [rev: 1] [classification: 31] [priority: 3]
      was not found in barnyard2 signature cache, this could lead to display inconsistency.
      To prevent this warning, make sure that your sid-msg.map and gen-msg.map file are up to date with the snort process logging to the spool file.
      The new inserted signature will not have its information present in the sig_reference table.
      Note that the message inserted in the signature table will be snort default message "Snort Alert [gid:sid:revision]"
      You can always update the message via a SQL query if you want it to be displayed correctly by your favorite interface
```

```
04/04-10:30:59.918994 [**] [1:10000001:1] Snort Alert [1:10000001:1] [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.10 -> 192.168.0.44
04/04-10:31:00.920192 [**] [1:10000001:1] Snort Alert [1:10000001:1] [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.44 -> 192.168.0.10
04/04-10:31:00.920244 [**] [1:10000001:1] Snort Alert [1:10000001:1] [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.10 -> 192.168.0.44
04/04-10:31:01.919231 [**] [1:10000001:1] Snort Alert [1:10000001:1] [**] [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.44 -> 192.168.0.10
Waiting for new data
```

Al pulsar **CTRL+C** para detener Barnyard2, nos mostrará información de los registros procesados.

Comprobamos si Barnyard2 ha escrito los eventos en la base de datos:

```
mysql -u snort -p -D snort -e "select count(*) from event"
```

La salida tiene que ser similar a esto:

```
+-------------+
| count(*)  |
+-------------+
|      12    |
+-------------+
```

Si el numero es mayor de 0 significa que Snort y Barnyard2 están bien instalados y configurados.

4. – Instalar PulledPork (Descarga automática de reglas para Snort).

PulledPork es un script escrito en Perl, que permite de forma automática, descargar, combinar e instalar/actualizar las reglas más recientes de Snort.

4.1 – Instalar los prerrequisitos de PulledPork.

```
sudo apt-get install -y libcrypt-ssleay-perl liblwp-useragent-determined-perl
```

4.2 – Descargar e instalar PulledPork.

Los siguientes comandos descargan e instalan PulledPork:

```
cd ~/snort_src/
wget https://github.com/shirkdog/pulledpork/archive/master.tar.gz -O pulledpork-master.tar.gz
tar xzvf pulledpork-master.tar.gz
cd pulledpork-master/
sudo cp pulledpork.pl /usr/local/bin/
sudo chmod +x /usr/local/bin/pulledpork.pl
sudo cp etc/*.conf /etc/snort/
```

Verificar que PulledPork funciona:

```
/usr/local/bin/pulledpork.pl -V
```

Si todo va bien nos mostrará la salida con la versión instalada:

```
PulledPork v0.7.3 - Making signature updates great again!
```

4.3 – Configurar PulledPork.

Primero hay que crear una cuenta en https://www.snort.org para obtener el Oinkcode. Este código nos permite descargar las reglas actualizadas y documentación. El Oinkcode debe guardarse en un sitio seguro.

Una vez que se dispone del Oinkcode ya se puede configurar PulledPork:

```
sudo vim /etc/snort/pulledpork.conf
```

Reemplazar las palabras **<oinkcode>** de las líneas **19** y **26**, por el Oinkcode obtenido en el paso anterior.

Descomentar la linea **29**:

```
rule_url=https://rules.emergingthreats.net/|emerging.rules.tar.gz|open-nogpl
```

Modificar la línea **74**:

```
rule_path=/usr/local/etc/snort/rules/snort.rules
```

Por:

```
rule_path=/etc/snort/rules/snort.rules
```

Cambiar la linea **89**:

```
local_rules=/usr/local/etc/snort/rules/local.rules
```

Por:

```
local_rules=/etc/snort/rules/local.rules
```

Cambiar la linea **92**:

```
sid_msg=/usr/local/etc/snort/sid-msg.map
```

Por:

```
sid_msg=/etc/snort/sid-msg.map
```

Cambiar la linea **96**:

```
sid_msg_version=1
```

Por:

```
sid_msg_version=2
```

Cambiar la línea **119**:

```
config_path=/usr/local/etc/snort/snort.conf
```

Por:

```
config_path=/etc/snort/snort.conf
```

Cambiar la línea **133**:

```
distro=FreeBSD-8.1
```

Por:

```
distro=Ubuntu-12-04
```

Para cualquier versión de Ubuntu hay que especificar *Ubuntu-12-04*, aunque la versión sea distinta.

Cambiar la línea **141**:

```
black_list=/usr/local/etc/snort/rules/iplists/default.blacklist
```

Por:

```
black_list=/etc/snort/rules/iplists/black_list.rules
```

Cambiar la línea **150**:

```
IPRVersion=/usr/local/etc/snort/rules/iplists
```

Por:

```
IPRVersion=/etc/snort/rules/iplists
```

4.4 – Verificar la configuración de PulledPork y actualizar las reglas de Snort.

Para comprobar si PulledPork funciona correctamente, se utilizan las siguientes opciones:

OPCIÓN	DESCRIPCIÓN
-l	Escribe un log detallado en /var/log
-c /etc/snort/pulledpork.conf	Ruta del fichero de configuracion pulledpork.conf

Hay que crear el archivo de reglas snort.rules, si no existe, si no PulledPork da errores:

```
sudo touch /etc/snort/rules/snort.rules
```

Comprobar que PulledPork funciona correctamente:

```
sudo /usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf -l
```

El final de la salida del comando anterior es similar a lo siguiente:

```
Rule Stats...
        New:-------66
        Deleted:---19
        Enabled Rules:----30299
        Dropped Rules:----0
        Disabled Rules:---25833
        Total Rules:------56132
IP Blacklist Stats...
        Total IPs:-----22207

Done
Please review /var/log/sid_changes.log for additional details
Fly Piggy Fly!
```

Vemos que **PulledPork** tiene en este momento **56.132** rules y **22.207** black IP's.

Despues de esto, el fichero **/etc/snort/rules/snort.rules** contiene todas las reglas de **PulledPork**, combinadas en dicho fichero.

Editamos el archivo de configuración de Snort, para activar el fichero de reglas snort.rules:

```
sudo vim /etc/snort/snort.conf
```

A partir de la línea **538** se encuentra la sección de reglas **# Step #7: Customize your rule set**, seguido de todos los conjuntos de reglas desactivadas previamente.

Añadir en esta zona la siguiente línea:

```
include $RULE_PATH/snort.rules
```

Como hemos modificado la configuración de Snort, volvemos a verficar que sea correcta:

```
sudo snort -T -c /etc/snort/snort.conf -i ens33
```

Si todo esta bien, las dos ultimas líneas tienen que ser similares a lo siguiente:

```
…
Snort successfully validated the configuration!
Snort exiting
```

Ignorar los posibles warnings sobre flowbits not being checked y GID duplicate.

Configurar crontab para actualizar de forma automática las reglas de Snort con PulledPork:

```
sudo crontab -e
```

Para actualizar las reglas diariamente con PulledPork, añadir la siguiente tarea programada:

```
# PulledPork
30 05 * * * /usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf -l
```

5. – Scripts de inicio automático para Snort y Barnyard2.

Vamos a crear dos scripts, para ejecutar Snort y Barnyard2 automaticamente al iniciar el sistema.

5.1 – Script de autoinicio para Snort.

Crear o editar un fichero para almacenar el códido del script para Snort:

```
sudo vim /lib/systemd/system/snort.service
```

Añadir las siguientes líneas:

```
[Unit]
Description=Snort NIDS Daemon
After=syslog.target network.target
[Service]
Type=simple
ExecStart=/usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
```

```
[Install]
WantedBy=multi-user.target
```

Activar el script para que se cargue al inciar el sistema:

```
sudo systemctl enable snort
```

Iniciar el servicio manualmente:

```
sudo systemctl start snort
```

Y verificar que el servicio se ejecuta correctamente:

```
systemctl status snort
```

La salida del comando anterior es similar a lo siguiente:

```
● snort.service - Snort NIDS Daemon
   Loaded: loaded (/lib/systemd/system/snort.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2017-04-28 01:42:12 CEST; 3s ago
 Main PID: 49005 (snort)
    Tasks: 1 (limit: 19660)
   CGroup: /system.slice/snort.service
           └─49005 /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i ens33

abr 28 01:42:12 UbuntuV systemd[1]: Started Snort NIDS Daemon.
```

5.2 – Barnyard2 script de autoinicio.

Crear o editar un fichero para almacenar el códido del script para Barnyard2:

```
sudo vim /lib/systemd/system/barnyard2.service
```

Añadir las siguientes líneas:

```
[Unit]
Description=Barnyard2 Daemon
After=syslog.target network.target
[Service]
Type=simple
ExecStart=/usr/local/bin/barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -q -w
/var/log/snort/barnyard2.waldo -g snort -u snort -D -a /var/log/snort/archived_logs
[Install]
WantedBy=multi-user.target
```

Activar el script para que se cargue al inciar el sistema:

```
sudo systemctl enable barnyard2
```

Iniciar el servicio manualmente:

```
sudo systemctl start barnyard2
```

Y verificar que el servicio se ejecuta correctamente:

```
systemctl status barnyard2
```

La salida del comando anterior es similar a lo siguiente:

```
● barnyard2.service - Barnyard2 Daemon
   Loaded: loaded (/lib/systemd/system/barnyard2.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2017-04-28 01:43:27 CEST; 13s ago
 Main PID: 49100 (barnyard2)
    Tasks: 1 (limit: 19660)
   CGroup: /system.slice/barnyard2.service
           └─49100 /usr/local/bin/barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -q -w

abr 28 01:43:27 UbuntuV systemd[1]: Started Barnyard2 Daemon.
abr 28 01:43:27 UbuntuV barnyard2[49100]: Running in Continuous mode
abr 28 01:43:27 UbuntuV barnyard2[49100]:
abr 28 01:43:27 UbuntuV barnyard2[49100]:        --== Initializing Barnyard2 ==--
abr 28 01:43:27 UbuntuV barnyard2[49100]: Initializing Input Plugins!
abr 28 01:43:27 UbuntuV barnyard2[49100]: Initializing Output Plugins!
abr 28 01:43:27 UbuntuV barnyard2[49100]: Parsing config file "/etc/snort/barnyard2.conf"
abr 28 01:43:27 UbuntuV barnyard2[49100]:

                                +[ Signature Suppress list ]+
                                ----------------------------
abr 28 01:43:27 UbuntuV barnyard2[49100]: +[No entry in Signature Suppress List]+
abr 28 01:43:27 UbuntuV barnyard2[49100]: ----------------------------
                                +[ Signature Suppress list ]+
```

Reiniciar el sistema y verificar que los dos scripts se inician correctamente después de reiniciar.

6. – Snorby Web GUI para Snort.

Vamos a instalar Snorby. Snorby es una aplicación web que utiliza Ruby con Rails y permite gestionar las alertas detectadas por Snort.

6.1 – Instalar prerrequisitos para Snorby y Ruby 2.3.

```
sudo apt-get install libgdbm-dev libncurses5-dev git-core curl zlib1g-dev build-essential libssl-dev libreadline-dev libyaml-dev libsqlite3-dev sqlite3 libxml2-dev libxslt1-dev libcurl4-openssl-dev python-software-properties libffi-dev postgresql-server-dev-9.6 -y
```

6.2 – Instalar prerrequisitos para las Gemas de Ruby.

```
sudo apt-get install imagemagick apache2 libyaml-dev libxml2-dev libxslt-dev git libssl-dev -y
```

Snorby utiliza Ruby y requiere varias gemas para instalarlo. Para acelerar el proceso, se puede desactivar la instalación de la documentación de las gemas, para ello, ejecutar los siguientes comandos:

```
echo "gem: --no-rdoc --no-ri" > ~/.gemrc
sudo sh -c "echo gem: --no-rdoc --no-ri > /etc/gemrc"
```

6.3 – Descargar e instalar Ruby 2.3.0.

Los siguientes comandos descargan, compilan e instalan Ruby 2.3.0:

```
cd ~/snort_src/
wget http://cache.ruby-lang.org/pub/ruby/2.3/ruby-2.3.0.tar.gz
tar -zxvf ruby-2.3.0.tar.gz
cd ruby-2.3.0/
./configure
make
sudo make install
```

6.4 – Instalar las gemas requeridas.

```
sudo gem install wkhtmltopdf
sudo gem install bundler
sudo gem install rails
sudo gem install rake --version=11.1.2
```

6.5 – Descargar Snorby y crear el directorio web.

```
cd ~/snort_src/
git clone git://github.com/Snorby/snorby.git
sudo cp -r snorby/ /var/www/html/
```

6.6 – Instalar prerrequisitos para Snorby.

```
cd /var/www/html/snorby/
sudo bundle install
```

Ignorar los avisos sobre la ejecución de *bundle* como root.

6.7 – Configurar la conexión MySQL de Snorby.

Copiar el fichero de conexión que viene de ejemplo:

```
sudo cp /var/www/html/snorby/config/database.yml.example /var/www/html/snorby/config/database.yml
```

Editar el archivo de configuración:

```
sudo vim /var/www/html/snorby/config/database.yml
```

En este punto es necesario especificar el username y password de usuario root de MySQL par que Snorby pueda crear y configurar su base de datos al instalarlo. Para evitar problemas de seguridad, una vez instalado y configurado Snorby, hay que crear un usuario en MySQL especifico y con menos privilegios y no utilizar el usuario root.

A continuación, vemos parte del fichero *database.yml*:

```
# Snorby Database Configuration
#
# Please set your database password/user below
# NOTE: Indentation is important.
#
snorby: &snorby
  adapter: mysql
  username: root
  password: "PasswordForRoot" # Example: password: "s3cr3tsauce"
  host: localhost
…
```

6.8 – Instalar Snorby.

Primero hay que crear el archivo de configuración de Snobry, copiar la configuración de ejemplo:

```
sudo cp /var/www/html/snorby/config/snorby_config.yml.example var/www/html/snorby/config/snorby_config.yml
```

Editar snorby_config.yml y corregir la ruta del archivo *wkhtmlpdf*:

```
sudo sed -i s/"\/usr\/local\/bin\/wkhtmltopdf"/"\/usr\/bin\/wkhtmltopdf"/g /var/www/html/snorby/config/snorby_config.yml
```

El siguiente comando realiza la instalación de Snorby:

```
sudo bundle exec rake snorby:setup
```

Si se produce algún error, revisar el log de salida para ver cual es el problema.

6.9 – Configurar la base de datos MySQL de Snorby.

Para no utilizar el usuario root de MySQL, creamos un usuario específico para Snorby:

```
mysql -u root -p                                                    # Iniciar sesión en MySQL
create user 'snorby'@'localhost' IDENTIFIED BY 'PASSWORDNEW';       # Crear el usuario snorby
grant all privileges on snorby.* to 'snorby'@'localhost' with grant option;
                                                                    # Dar privilegios al usuario Snorby sobre la base
                                                                    # de datos de Snorby.
flush privileges;                                                   # Hacer efectivos los nuevos privilegios asignados
exit                                                                # Salir de MySQL
```

6.10 – Reconfigurar la conexión MySQL de Snorby.

Editar el fichero de configuración de conexión a la base de datos:

```
sudo vim /var/www/html/snorby/config/database.yml
```

Modificar el username y password por los del punto anterior. El fichero debe ser similiar a lo siguiente:

```
# Snorby Database Configuration
#
# Please set your database password/user below
# NOTE: Indentation is important.
#
snorby: &snorby
  adapter: mysql
  username: snorby
  password: "PASSWORDNEW" # Example: password: "s3cr3tsauce"
  host: localhost

development:
  database: snorby
  <<: *snorby

test:
  database: snorby
  <<: *snorby

production:
  database: snorby
  <<: *snorby
```

6.11 – Integrar Barnyard2 con Snorby.

Hay que reconfigurar Barnyard2 para que envíe las alertas a la base de datos de Snorby.

Editar el archivo de configuración de Barnyard2, barnyard2.cfg:

```
sudo vim /etc/snort/barnyard2.conf
```

Añadir la siguiente línea al final del fichero:

```
output database: log, mysql, user=snorby password=PASSWORD dbname=snorby host=localhost sensor_name=sensor1
```

Comentar la línea configurada pareviamente para la salida de la base de datos:

```
#output database: log, mysql, user=snort password=PASSWORD dbname=snort host=localhost
```

Reiniciar el servicio Barnyard2 para que cargue los cambios de configuración:

```
sudo service barnyard2 restart
```

6.12 – Verificar que Barnyard2 escribe las alertas en la base de datos de Snorby.

Esperar unos minutos para que se produzcan algunas alertas o ejecutar un ping en la red monitorizada.

Comprobar que los eventos se escriben en la base de datos *snorby*:

```
mysql -u snorby -p -D snorby -e "select count(*) from event"
```

Comprobar en la salida que el numero de registros es mayor de 0:

```
+-------------+
| count(*) |
```

```
+-------------+
|          36 |
+-------------+
```

6.13 – Crear un demonio Snorby worker.

Para el mantenimiento de la base de datos de Snorby hay que crear un demonio. Este demonio se encarga de lanzar el proceso que interpreta las alertas grabadas por Barnyard2 en la base de datos, para que puedan ser visualizadas en la interfaz web de Snorby. El Snorby worker también se puede lanzar desde la interfaz web.

Crear y editar el demonio Snorby worker:

```
sudo vim /lib/systemd/system/snorby_worker.service
```

Añadir las siguientes líneas:

```
[Unit]
Description=Snorby Worker Daemon
Requires=apache2.service
After=syslog.target network.target apache2.service
[Service]
Type=forking
WorkingDirectory=/var/www/html/snorby
ExecStart=/usr/local/bin/ruby script/delayed_job start
[Install]
WantedBy=multi-user.target
```

Activar el script para que se ejecute al iniciar el sistema:

```
sudo systemctl enable snorby_worker
```

Verificar que el script se esta ejecutando:

```
systemctl status snorby_worker.service
```

6.14 – Testear Snorby.

Para testear Snorby, ejecutamos el siguiente comando desde el directorio web de Snorby:

```
cd /var/www/html/snorby/
sudo bundle exec rails server -e production
```

El comando anterior inicia **Snorby** en el puerto **3000** del host donde esta instalado.

Para comprobar que Snorby esta funcionando correctamente, acceder con el navegador a http://192.168.0.23:3000, cambiar la IP por la correspondiente al equipo donde esta configurando Snorby. Si apararece la pantalla de Login, la instalación se ha realizado correctamente.

En este momento no es necesario inciar sesión ya que vamos a realizar una configuarcion adicional con Phusion Passenger para lanzar snorby automáticamente.

Para detener **Snorby** pulsar **CRTL+C**.

7. – Instalar Phusion Passenger y configurar Apache.

Phusion Passenger es un módulo de servidor de aplicaciones para Apache que permite lanzar Snorby de forma automática.

7.1 – Instalar los prerrequisitos para Phusion Passenger.

```
sudo apt-get install libcurl4-openssl-dev libaprutil1-dev libapr1-dev apache2-dev
```

7.2 – Instalar la gema Passenger y el módulo de Apache.

Instalar la gema de Passenger:

```
sudo gem install passenger
```

Instalar el módulo passenger para Apache:

```
sudo passenger-install-apache2-module
```

El comando anterior inicia el asistente de instalación de Phusion Passenger.

- Pulsar **Enter.**
- Con las flechas de dirección, bajar hasta la opción **Python**.
- Presionar la Barra Spaciadora para deseleccionar **Python**, ya que no es necesario.
- Presionar **Enter**.

Ahora se iniciará la compilación de Phusion Passenger y tarda un rato.
Cuando termine de compilar, saldrá una pantalla con la configuración predeterminada del módulo Apache:

```
LoadModule passenger_module /usr/local/lib/ruby/gems/2.3.0/gems/passenger-5.0.26/buildout/apache2/mod_passenger.so
<IfModule mod_passenger.c>
        PassengerRoot /usr/local/lib/ruby/gems/2.3.0/gems/passenger-5.0.26
        PassengerDefaultRuby /usr/local/bin/ruby
</IfModule>
```

En este caso, esta configuración no sirve ya que Apache2 en la actualidad utiliza dos archivos, uno para la ruta del módulo y otro para la configuración de dicho módulo. Copiar el contenido, ya que, es necesario para configurar los siguientes pasos.

Pulsar **Enter** para salir del asistente.

7.3 – Configurar Apache para usar Phusion Passenger.

Crear y editar el archivo *passenger.load*:

```
sudo vim /etc/apache2/mods-available/passenger.load
```

E insertar la siguiente línea:

```
LoadModule passenger_module /usr/local/rvm/gems/ruby-2.3.0/gems/passenger-5.0.26/buildout/apache2/mod_passenger.so
```

Crear y editar el fichero *passenger.conf*:

```
sudo vim /etc/apache2/mods-available/passenger.conf
```

E insertar las siguientes lineas:

```
PassengerRoot /usr/local/lib/ruby/gems/2.3.0/gems/passenger-5.0.26
PassengerDefaultRuby /usr/local/bin/ruby
```

Activar el módulo de passenger y reiniciar apache:

```
sudo a2enmod passenger
sudo service apache2 restart
```

Comprobar que el módulo está cargado:

```
sudo apache2ctl -t -D DUMP_MODULES
```

Si está cargado, en la salida aparecerá el módulo passenger_module:

```
…
negotiation_module (shared)
 passenger_module (shared)
 reqtimeout_module (shared)
 setenvif_module (shared)
```

7.4 – Crear virtualhost Apache para Snorby.

Crear y editar el fichero de configuración del sitio web Apache para Snorby:

```
sudo vim /etc/apache2/sites-available/001-snorby.conf
```

Añadir las siguientes lineas:

```
<VirtualHost *:80>
        ServerAdmin admin@example.com
        ServerName snort-ids.example.com
        DocumentRoot /var/www/html/snorby/public

        <Directory "/var/www/html/snorby/public">
                AllowOverride all
                Order deny,allow
                Allow from all
                Options -MultiViews
        </Directory>
</VirtualHost>
```

Habilitar el virtualhost para Snorby y reiniciar el servicio apache2 para activar el sitio web:

```
cd /etc/apache2/sites-available/
sudo a2ensite 001-snorby.conf
sudo service apache2 restart
```

Deshabilitar el virtualhost que hay por defecto:

```
cd /etc/apache2/sites-enabled/
sudo a2dissite 000-default.conf
sudo service apache2 restart
```

En un servidor que disponga de dominio, no es necesario desactivar el virtual host por defecto. Para acceder a Snort se puede hacer a través de un subdominio o creando un alias, por ejemplo.

7.8 – Testear interfaz web de Snorby.

Reiniciar el sistema y acceder con el navegador web a la url o IP del servidor de Snorby. Iniciar sesión con los siguientes datos:

EMAIL	PASSWORD
snorby@example.com	Snorby

Si todo esta funcionando correctamente, visualizaremos la página principal de Snorby con un resumen de las alertas detectadas por Snort.

Desde la interfaz web podemos acceder a toda la información relativa a las alertas detectadas por Snort.

8. – Snort en modo promiscuo (ESXi).

Es posible configurar Snort para capturar el tráfico de una red diferente a la del host donde se encuentra instalado, para ello hay que instalar una interfaz de red en modo promiscuo.

Vamos a configurar un servidor ESXi para que refleje todo el tráfico en una interfaz específica del servidor NIDS.

A partir de este punto queda pendiente traducir y probar.

8.1 – Create the virtual switch

We need create a vSwitch in ESXi and configure it:

- Log into the vCenter server.
- Select the ESXi host.
- Click the **Configuration** tab.
- On the left, under **Hardware**, select **Networkin**g.
- Click **Propertie**s on the vSwitch where Snort will have the listening interface on.
- Under the **Ports** tab, double click on the virtual switch or portgroup you want to modify.
- Under the **Security** tab click on the **Promiscuous Mode** checkbox and select **Accept** from the dropdown menu.
- Click the **OK**, then the **Close** button.

8.2 – Create a new interface

We need to ad a new interface to the IDS server:

- Log into the vCenter server.
- Right click the IDS server and select **Edit Settings…**
- Under the **Hardware** tab click on the **Add** button.
- Select **Ethernet Adapter** and click Next >
- Select **VMXNET 3** as **Adapter Type**.

- Under **Named network with specified label:** select the vSwitch with promiscuous mode enabled.
- Click **Next >** and then **Finish**.
- Click on the new network adapter and check the MAC address, we are going to need it later.
- Click on the **OK** button to close the virtual machine properties window.

8.3 – Configure the interface

Remember that the interfaces are named as Predictable Network Interfaces, so we need to find out the name of our new interface.

```
ip link
```

Check the output, look for the MAC address of the new interface and get the name.
In my case it was **ens66**

```
sudo vim /etc/network/interfaces
```

Configure the new interface in promiscuous mode.
Add the following lines:

```
auto ens66
iface ens66 inet manual
        up ifconfig $IFACE 0.0.0.0 up
        up ip link set $IFACE promisc on
        down ip link set $IFACE promisc off
        down infconfig $IFACE down

post-up ethtool -K ens66 gro off
post-up ethtool -K ens66 lro off
```

This is how my **interfaces** file looks like now:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# Management interface
auto ens33
iface ens33 inet static
    address 192.168.1.10
    netmask 255.255.252.0
    gateway 192.168.1.1
dns-nameservers 192.168.1.1
```

```
# Listening interface in Promiscous Mode
auto ens66
iface ens66 inet manual
      up ifconfig $IFACE 0.0.0.0 up
      up ip link set $IFACE promisc on
      down ip link set $IFACE promisc off
      down infconfig $IFACE down

post-up ethtool -K ens66 gro off
post-up ethtool -K ens66 lro off
```

8.4 – Reconfigure Snort

Now we have to tell Snort to listen on the interface with promiscuous mode enabled:

```
sudo vim /lib/systemd/system/snort.service
```

Replace **ens33** with **ens66**

And that's all. Now you have **Snort**, **Barnyard2**, **PulledPork** and **Snorby** all working together.

¡Espero que hayáis disfrutado!!!!!