

# Detección de intrusos IDS - SNORT + EasyIDS

Ismael Macareno Chouikh

2025-01-19

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Materiales necesarios</b>	<b>2</b>
<b>3. Instalación de snort en GNU/Linux</b>	<b>2</b>
3.1. Configuración de <b>snort</b> . . . . .	3
3.2. Ejemplo 1 . . . . .	3
3.3. Ejecutando <b>snort</b> en modo <i>sniffer</i> . . . . .	4
3.4. Ejecutando <b>snort</b> en modo <i>log</i> de paquetes . . . . .	4
<b>4. EasyIDS</b>	<b>5</b>
4.1. Configuración de <b>EasyIDS</b> . . . . .	7
4.2. Comprobar los servicios . . . . .	8
4.2.1. Solución de problemas para los servicios <b>NTOP</b> y <b>snort</b> . . . . .	9
4.3. Observación sobre lo capturado . . . . .	11

## 1. Introducción

**snort** es un **sistema de detección de intrusos en red**. Funciona de manera muy similar a un *sniffer* ya que se configura para monitorizar todo el tráfico de red en búsqueda de cualquier tipo de intrusión.

**snort** está disponible bajo licencia GPL, es gratuito y funciona tanto en *Microsoft Windows* como en GNU/Linux.

## 2. Materiales necesarios

- Máquina Virtual Ubuntu 20.04
  - RAM ->4069
  - Disco duro ->40 GiBi
  - Cores ->1
- Máquina virtual tipo CentOS
  - arquitectura de 32bits
  - S.O ->EasyIDS
  - RAM ->4096
  - Disco ->30 GiBi

## 3. Instalación de snort en GNU/Linux

En distribuciones **Ubuntu** se instala fácilmente con **apt**. Se puede instalar únicamente **snort** o también se puede instalar con soporte a mysql.

- `sudo apt-get update`
- `sudo apt-get install snort`

A la hora de instalar **snort** se nos pedirán dos cosas:

- interfaz de red donde va a escuchar **snort** (en mi caso **enp0s3**)
- rango de la red (en mi caso **172.26.0.0/16**)

En caso de que no se instale mediante el comando se puede ejecutar el script **BASH** que adjunto incrustado en el PDF [Adjunto fichero BASH](#)

Luego de la instalación lo que haremos será modificar la dirección IP de nuestra MV.

En este ejemplo se supone una red interna **172.26.0.0/16** por lo que tendremos que configurar nuestra MV de tal manera que su dirección IP pertenezca a está red. En mi caso lo hago mediante la edición del fichero **/etc/netplan/01-netcfg.yaml** de la siguiente manera:

```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
```

```

enp0s3:
  dhcp4: true
enp0s8:
  addresses:
    - 172.26.1.1/24
  dhcp4: false

```

### 3.1. Configuración de snort

A la hora de configurar **snort** lo que tendremos que hacer será acceder al directorio `/etc/snort/rules` y modificar el fichero `local.rules` para añadir nuevas reglas

```

# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures.  Put your local
# additions here.
alert tcp 10.0.0.0/24 any -> any any (content: "www.facebook.com"; msg:"alguien se encuentra
visitado facebook";sid:1000001; rev:1 )

alert tcp any any -> any any (content: "www.facebook.com";msg:"alguien se encuentra
visitando facebook!";sid:1000002; rev:1 )

alert icmp any any -> 10.0.0.2 any (msg:"Se ha recibido PING al equipo servidor!"; sid:1000003;
rev:1 )

alert tcp any any -> 10.0.0.2 22 (msg:"conexion via SSH al servidor!"; sid:1000004; rev:1 )

```

Las reglas **snort** las podemos dividir en dos secciones lógicas, a saber: **cabecera de la regla y opciones**:

- **Cabecera:** contiene la acción de la regla.
- **Opciones:** contiene los mensajes y la información necesaria para la decisión a tomar por parte de la alerta.

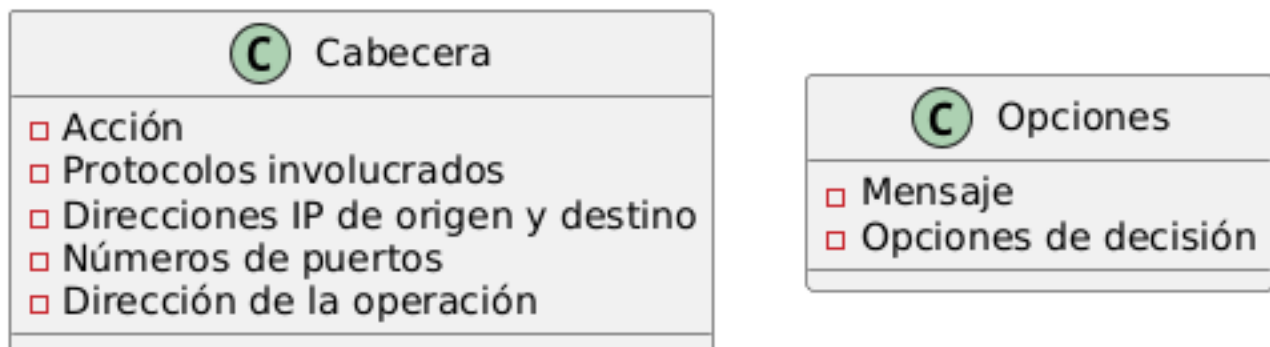


Figura 1: Macareno, Ismael. (2025). División de reglas **snort** [PNG]. Propio

### 3.2. Ejemplo 1

Veamos ahora un ejemplo de regla **snort** para alertar de un escaneo **nmap** del tipo TPC/ping

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any / (msg:"Escaneo ping con
nmap";flags:A;ack:0; / reference:arachnids,28;classtype:attempted-recon;
sid:628;/ rev:1;)
```

### 3.3. Ejecutando snort en modo *sniffer*

Después de salir guardando los cambios, ejecutamos **snort** con el siguiente comando:

- `sudo snort -c /etc/snort/snort.conf -A console -i enp0s3`

Una vez ejecutemos **snort** lo que tendremos que hacer será desde otra máquina hacer un **ping** al ubuntu server 20.04. En mi caso he usado mi máquina real.

El resultado será algo parecido a lo siguiente:

```
01/16-00:41:31.087793  ** [1:366:7] ICMP PING *NIX ** [Classification: Misc activity]
[Priority: 3] {ICMP} 192.168.1.49 -> 192.168.1.120
01/16-00:41:31.087793  ** [1:384:5] ICMP PING ** [Classification: Misc activity]
[Priority: 3] {ICMP} 192.168.1.49 -> 192.168.1.120
01/16-00:41:31.087843  ** [1:408:5] ICMP Echo Reply ** [Classification: Misc activity]
[Priority: 3] {ICMP} 192.168.1.120 -> 192.168.1.49
01/16-00:41:32.096712  ** [1:366:7] ICMP PING *NIX ** [Classification: Misc activity]
[Priority: 3] {ICMP} 192.168.1.49 -> 192.168.1.120
01/16-00:41:32.096712  ** [1:384:5] ICMP PING ** [Classification: Misc activity]
[Priority: 3] {ICMP} 192.168.1.49 -> 192.168.1.120
01/16-00:41:32.096766  ** [1:408:5] ICMP Echo Reply ** [Classification: Misc activity]
[Priority: 3] {ICMP} 192.168.1.120 -> 192.168.1.49
01/16-00:41:33.109201  ** [1:366:7] ICMP PING *NIX ** [Classification: Misc activity]
[Priority: 3] {ICMP} 192.168.1.49 -> 192.168.1.120
```

Como se puede apreciar se nos muestra:

- desde que dirección IP se está realizando el **ping**
- el tipo de protocolo (ICMP)

### 3.4. Ejecutando snort en modo *log* de paquetes

La ejecución de **snort** produce una gran cantidad de datos de salida. Por ello, para un análisis más pausado, se recomienda almacenarla en algún archivo de *log*.

La forma de ejecutar **snort** para que la salida se guarde en un fichero de *log* es la siguiente:

- `sudo snort -dev -c /etc/snort/snort.conf -l /var/snort/snort.log -h 172.26.0.0/16`

Como se puede apreciar en el comando de arriba el fichero en el cuál vamos a almacenar los *logs* se localiza en `/var/snort/snort.log` debido a esto tendremos que hacer lo siguiente:

- `sudo mkdir /var/snort`
- `sudo chmod 755 /var/snort/`

También habrá que modificar el fichero `/etc/snort/snort.conf` en la siguiente línea:

```
#####
# Step #6: Configure output plugins
# For more information, see Snort Manual, Configuring Snort - Output Modules
#####

# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
output unified2: filename /var/snort/snort.log, limit 128, nostamp, mpls_event_types, vlan_event_types
```

Una vez realizados los pasos anteriores lo que tendremos que hacer será ejecutar el comando del principio, dejarlo actuar un poco y revisar el fichero `/var/snort/snort.log`. (Está en binario ).

## 4. EasyIDS

**snort** es un **sistema de detección de intrusos en red**. Funciona de manera muy similar a un *sniffer* ya que se configura para monitorizar todo el tráfico de red en búsqueda de cualquier tipo de intrusión.

**EasyIDS está preparado para una conexión de red DHCP**

Antes de empezar es conveniente tener activado un servidor DHCP en la red interna.

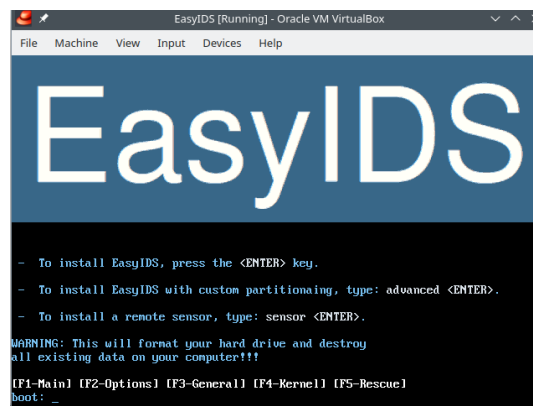
Crearemos una máquina virtual de tipo **CentOS 32 bits**. Cuando pida la imagen ISO se le dará la **EasyIDS-0.4.iso**.

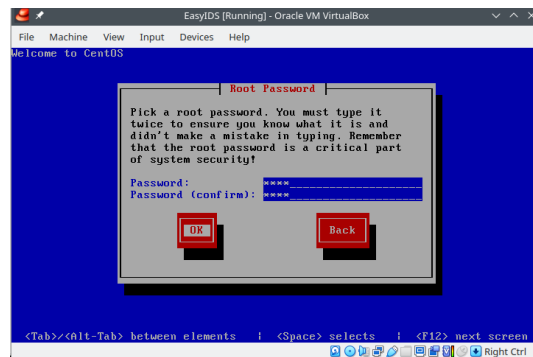
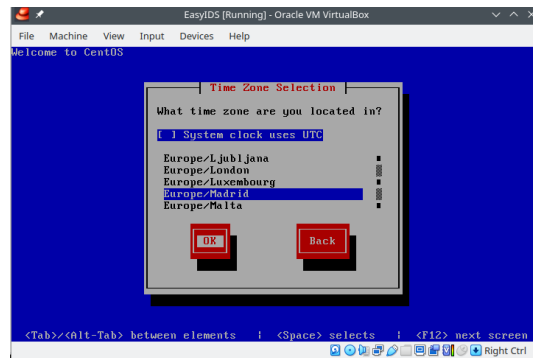
Se puede descargar la ISO en el siguiente link

- <https://sourceforge.net/projects/easyids/files/latest/download>

A la hora de ir a Oracle Virtualbox para crear nuestra máquina virtual para **EasyIDS**, lo que tendremos que hacer será elegir que será un Linux, específicamente un CentOS de arquitectura de 32bits. Luego seguiremos los siguientes pasos:

- Añadir la imagen ISO
- Ir a Configuración ->sistema ->decir que arranque de la ISO
- arrancar la MV e instalar EasyIDS





Luego lo que haremos será apagar la MV y volver a configuración ->sistema ->decir que arranque desde el disco duro.

Una vez finalizada la instalación del S.O EasyIDS lo que tendremos que hacer será configurar la tarjeta de red como **red interna** y con un dirección IP en este caso de clase C.

Para modificar la configuración de red en CentOS lo que tendremos que hacer será modificar el fichero `/etc/sysconfig/network-scripts/ifcfg-eth0` de la siguiente manera:

```
DEVICE=eth0
BOOTPROTO=static
DHCPCLASS=
HWADDR=08:00:27:05:29:FD
ONBOOT=yes
IPADDR="192.168.1.20"
PREFIX="24"
```

Luego para poder aplicar la nueva configuración de red tendremos que ejecutar los siguientes comandos:

- `sudo ifdown eth0`
- `sudo ifup eth0`

Se instalarán y configurarán mediante *scripts* automatizados, los paquetes correspondientes a las herramientas necesarias tales como:

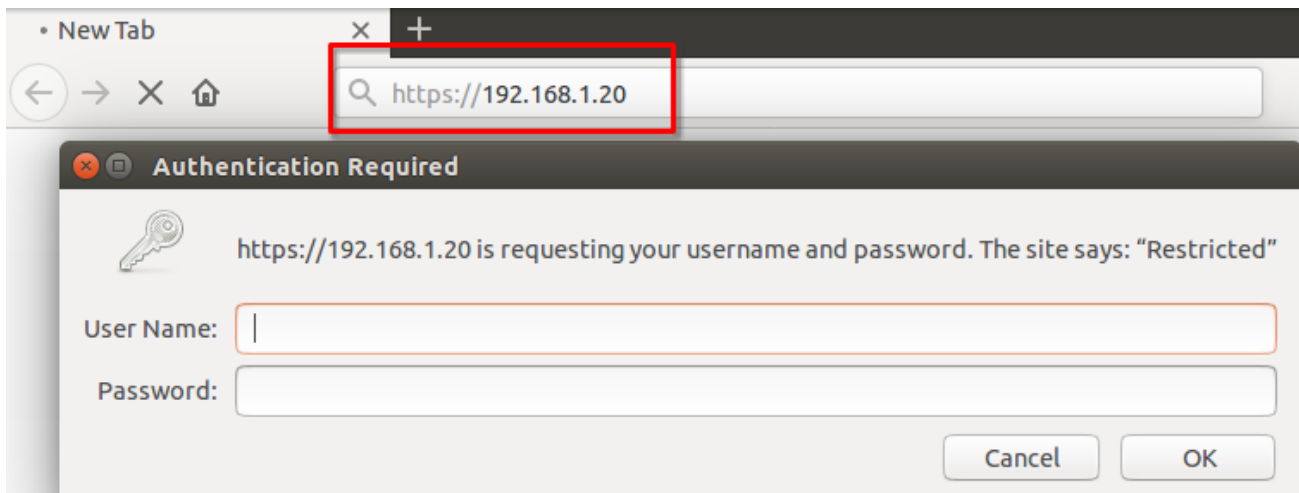
- `mysql`
- `snort`

- librerías
- apache
- oinkmaster
- etc.

#### 4.1. Configuración de EasyIDS

Bien, ya podemos acceder desde el navegador de otra máquina virtual siempre y cuando estén en la misma red (contando con direcciones IP y switch).

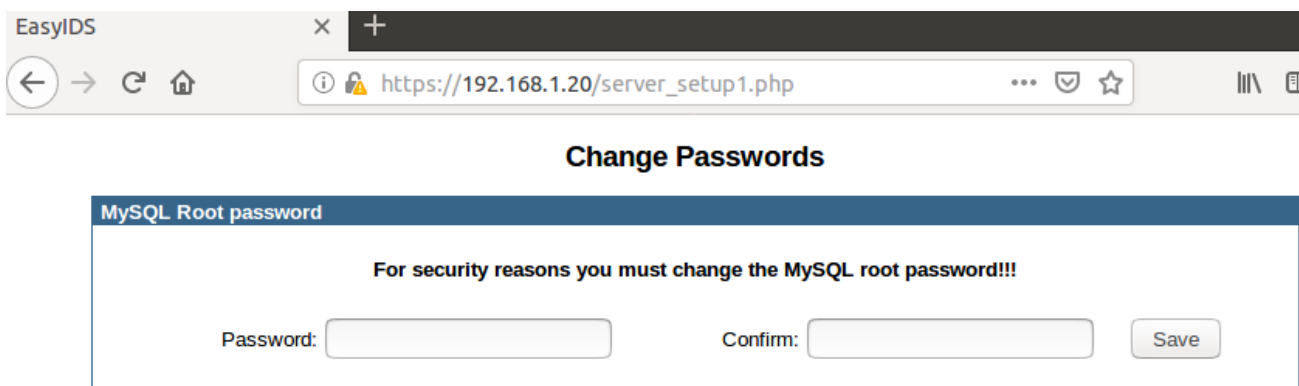
Yo en mi caso he elegido usar una MV de Ubuntu 14.04 LTS ya que el navegador de esta máquina al estar muy desactualizado no me dará muchos problemas a la hora de conectarme.



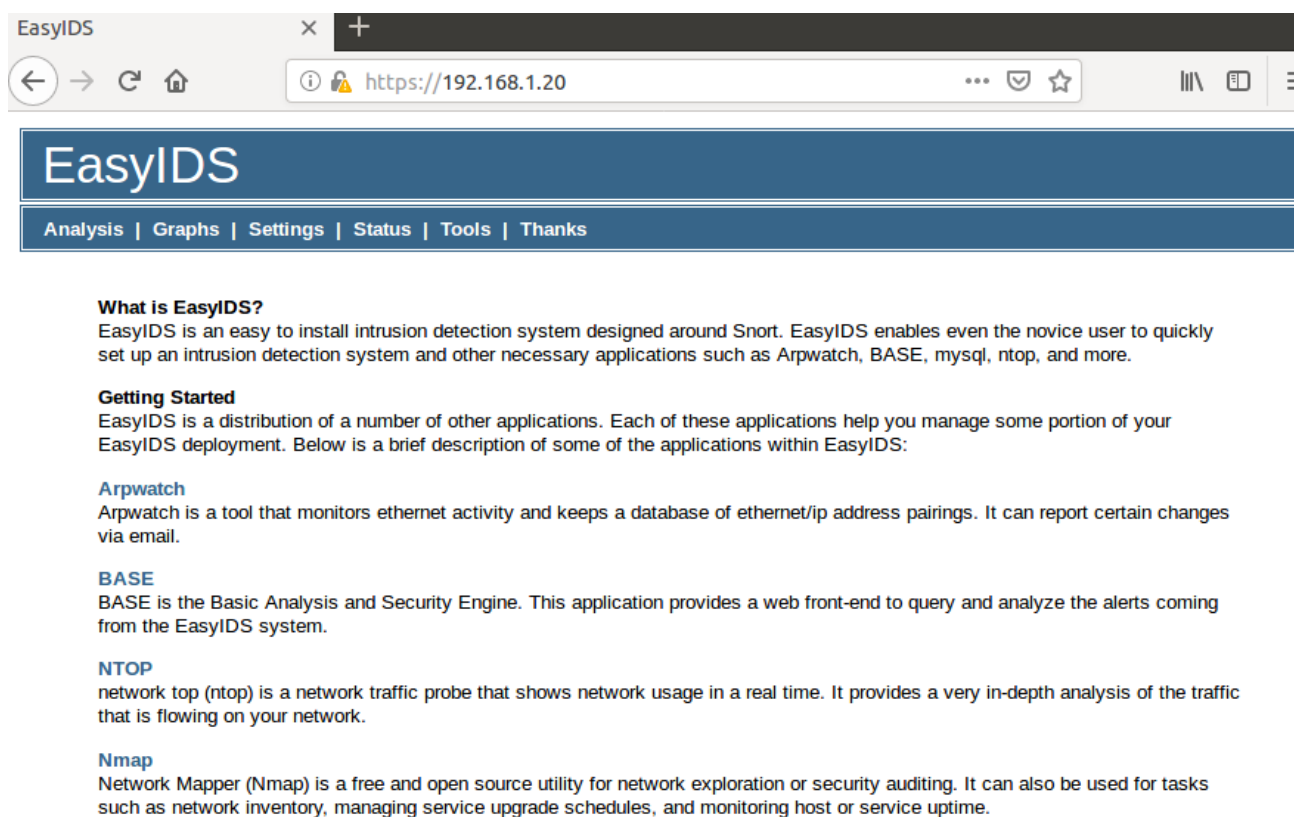
El usuario y contraseña son por defecto los siguientes:

- **usuario:** admin
- **contraseña:** *password*

Una vez entremos con esas credenciales se nos pedirá cambiar la contraseña del usuario administrador (**root**) de **mysql**



Una vez hayamos modificado las credenciales del usuario **root** de **mysql** nos aparecerá lo siguiente:



**EasyIDS**

Analysis | Graphs | Settings | Status | Tools | Thanks

**What is EasyIDS?**  
EasyIDS is an easy to install intrusion detection system designed around Snort. EasyIDS enables even the novice user to quickly set up an intrusion detection system and other necessary applications such as Arpwatch, BASE, mysql, ntop, and more.

**Getting Started**  
EasyIDS is a distribution of a number of other applications. Each of these applications help you manage some portion of your EasyIDS deployment. Below is a brief description of some of the applications within EasyIDS:

**Arpwatch**  
Arpwatch is a tool that monitors ethernet activity and keeps a database of ethernet/ip address pairings. It can report certain changes via email.

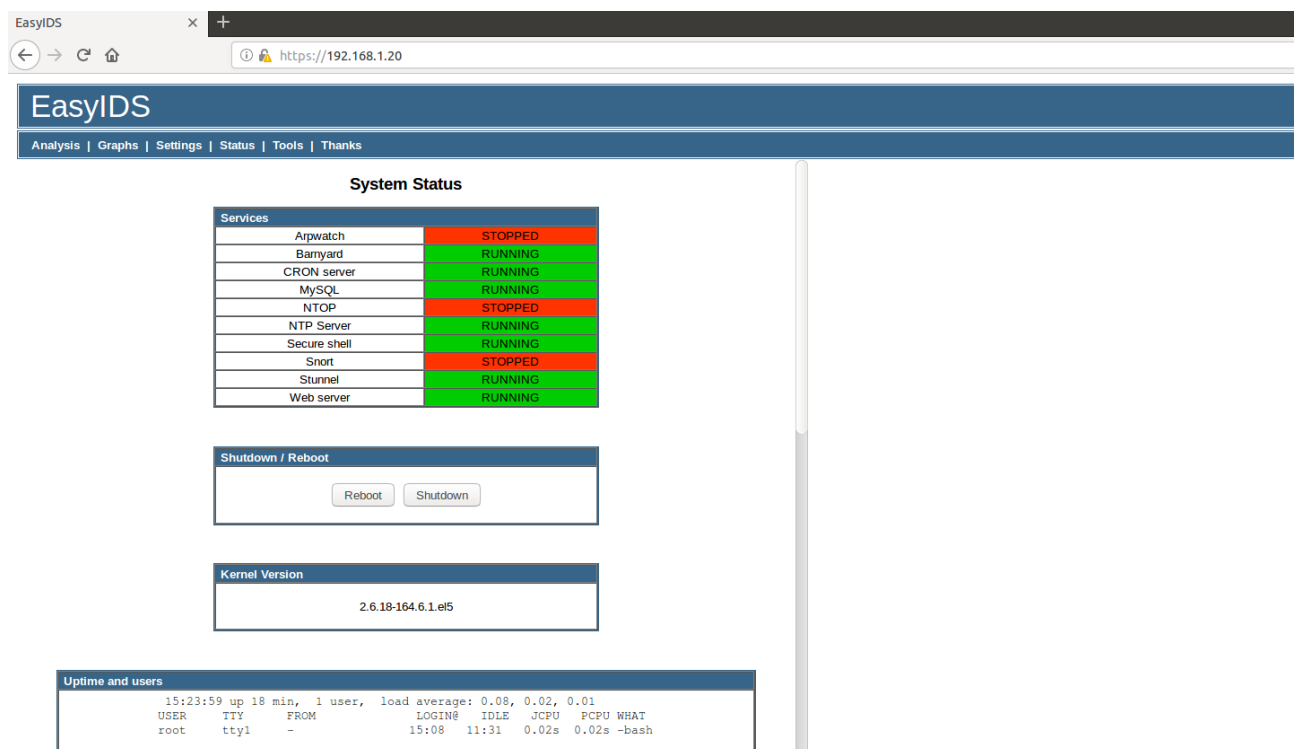
**BASE**  
BASE is the Basic Analysis and Security Engine. This application provides a web front-end to query and analyze the alerts coming from the EasyIDS system.

**NTOP**  
network top (ntop) is a network traffic probe that shows network usage in a real time. It provides a very in-depth analysis of the traffic that is flowing on your network.

**Nmap**  
Network Mapper (Nmap) is a free and open source utility for network exploration or security auditing. It can also be used for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

## 4.2. Comprobar los servicios

Lo primero que haremos será comprobar el estado los servicios. Para ello tendremos que acceder a **status** -> **system**



**EasyIDS**

Analysis | Graphs | Settings | Status | Tools | Thanks

**System Status**

Services	Status
Arpwatch	STOPPED
Barnyard	RUNNING
CRON server	RUNNING
MySQL	RUNNING
NTOP	STOPPED
NTP Server	RUNNING
Secure shell	RUNNING
Snort	STOPPED
Stunnel	RUNNING
Web server	RUNNING

**Shutdown / Reboot**

Reboot Shutdown

**Kernel Version**

2.6.18-164.6.1.el5

**Uptime and users**

```
15:23:59 up 18 min, 1 user, load average: 0.08, 0.02, 0.01
USER  TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
root  tty1    -                15:08   11:31   0.02s  0.02s -bash
```

Como se puede apreciar en la imagen de arriba no todo va bien ya que los siguiente servicios no se arrancaron:



- NTOP
- snort

#### 4.2.1. Solución de problemas para los servicios NTOP y snort

##### 1. *settings/NTOP/Network Settings*

- Configuramos *local subnet*
- Una red de laboratorio (Ej. 192.168.0.0/24)

### NTOP Network Settings

Interface:	eth0 ▼
Track Local Hosts:	NO ▼
Local Subnets: *	192.168.0.0/24
<b>You must restart NTOP for changes to take effect!</b>	
<div>Save Cancel</div>	

\* Required if tracking local hosts.

##### 1. Ahora vamos a por el motor de snort. *settings/snort/network settings*

- Tenemos que configurar las variables. Lo podemos dejar todo por defecto

## Snort Network Settings

<b>Home Network:</b>	<input type="text" value="\$eth0_ADDRESS"/>
<b>External Network:</b>	<input type="text" value="!\$HOME_NET"/>
<b>DNS Servers:</b>	<input type="text" value="\$HOME_NET"/>
<b>Mail Servers:</b>	<input type="text" value="\$HOME_NET"/>
<b>Web Servers:</b>	<input type="text" value="\$HOME_NET"/>
<b>FTP Servers:</b>	<input type="text" value="\$HOME_NET"/>
<b>SQL Servers:</b>	<input type="text" value="\$HOME_NET"/>
<b>Telnet Servers:</b>	<input type="text" value="\$HOME_NET"/>
<b>SNMP Servers:</b>	<input type="text" value="\$HOME_NET"/>
<b>You must restart Snort for changes to take effect!</b>	
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

En este caso, seguramente **snort** siga fallando, si es así lo que tendremos que hacer es lo siguiente:

- Pulsar sobre *edit config file* para modificar el fichero **snort.conf** directamente si es necesario
- Si sigue fallando:
  - Desde el CLI modificaremos el fichero **/etc/easyids/easyids.conf**

EasyIDS parte de la base de **snort** trabaja sobre una máquina con dos interfaces de red. Una para gestionar el sistema y la otra para monitorizar. En este caso hemos creado la máquina con una sola interfaz de red. Por tanto, pondremos **eth0** en ambas líneas del fichero.

Una vez modificadas, reiniciamos **snort** y ya debe hacerlo correctamente.

```
#EasyIDS configuration file
#Manual editing not suggested

#Specify the version number
EASYIDS_VERSION="0.4"

#Specify the brand name
EASYIDS_BRANDNAME="EasyIDS"

#Network interface for monitoring
EASYIDS_MONETH="eth0"

#Network interface for management
EASYIDS_MANETH="eth0"

#Network interface for bridge
#Requires monitoring interface be br0
EASYIDS_BRIDGE1=""

#Network interface for bridge
#Requires monitoring interface be br0
EASYIDS_BRIDGE2=""

#Allow inline mode
```

## System Status

Services	
Arpwatch	STOPPED
Barnyard	RUNNING
CRON server	RUNNING
MySQL	RUNNING
NTOP	RUNNING
NTP Server	RUNNING
Secure shell	RUNNING
Snort	RUNNING
Stunnel	RUNNING
Web server	RUNNING

### 4.3. Observación sobre lo capturado

Vamos a *analysis* -> *BASE* y podemos observar lo capturado sobre diversas pantallas

EasyIDS

Analysis | Graphs | Settings | Status | Tools | Thanks

- Today's alerts:

- Last 24 Hours alerts:

- Last 72 Hours alerts:

- Most recent 15 Alerts:

- Last Source Ports:

- Last Destination Ports:

- Most Frequent Source Ports:

- Most Frequent Destination Ports:

- Most frequent 15 Addresses:

- Most recent 15 Unique Alerts

- Most frequent 5 Unique Alerts

unique

unique

unique

any protocol

any protocol

any protocol

any protocol

any protocol

Source

listing

listing

listing

TCP

TCP

TCP

TCP

Destination

Source IP

Source IP

Source IP

UDP

UDP

UDP

UDP

Destination IP

Destination IP

Destination IP

ICMP

Queried on : Sun January 19, 2025 15:59:17

Database: snort@oscapad - (Schema Version: 10)

Time Window: [2025-01-19 14:58:50] - [2025-01-19 14:58:55]

Search

Graph Alert Data

Graph Alert Detection Time

Use Archive Database

Sensors/Total: 1 / 1

Unique Alerts: 1

Categories: 1

Total Number of Alerts: 10

• Src IP addr: 1

• Dest. IP addr: 1

• Unique IP links 1

• Source Ports: 1

◦ TCP ( 1 ) UDP ( 0 )

• Dest Ports: 1

◦ TCP ( 1 ) UDP ( 0 )

Traffic Profile by Protocol

TCP (100%)

UDP (0%)

ICMP (0%)

Portscan Traffic (0%)