



PROYECTO ASIR CONTROL WEB

Eduardo de Lamo Téllez

ÍNDICE

| | |
|--|-----------|
| 1. Introducción. | 1 |
| 2. Objetivos del Proyecto. | 1 |
| 3. Infraestructura Informática. Software utilizado. | 1 |
| 3.1. Escenario informático de red. | 2 |
| 3.2. Sistema Operativo Ubuntu. | 3 |
| 3.3. Snort. | 3 |
| 3.4. Apache, MySQL, PHP, phpMyAdmin. | 4 |
| 4. Instalación y Configuración. | 5 |
| 4.1. Ubuntu. | 5 |
| 4.2. Snort. | 8 |
| 4.2.1 Configuración de Snort. | 10 |
| 4.3. Apache. | 13 |
| 4.4. MySQL. | 14 |
| 4.5. PHP. | 15 |
| 4.6. phpMyAdmin. | 16 |
| 5. Base de datos implementada. | 18 |
| 6. Desarrollo de la aplicación web. | 19 |
| 6.1. Interfaz de usuario. | 21 |
| 6.2. Lógica de programación utilizada en la aplicación web. | 24 |
| 6.3. Ejemplos de uso de la aplicación Web. | 27 |
| 7. Coste económico del Proyecto. | 31 |
| 8. Conclusiones. | 33 |
| 9. Mejoras futuras del proyecto. | 34 |

1. Introducción.

En la actualidad las empresas, colegios, instituciones y demás organizaciones están conectadas a Internet y sus empleados, alumnos, usuarios y personal tienen acceso a la Web a través de ellas. El imparable crecimiento de la WWW y de su uso la hacen fundamental a día de hoy tanto para el aprendizaje como para el trabajo de las personas. Esto ha supuesto una gran revolución y ha cambiado la manera de hacer las cosas, hoy en día ya no entenderíamos un mundo sin acceso a la Web. Como sabemos, en ella podemos encontrar contenido infinito, lo que nos lleva a pensar que aunque la gran mayoría del cual sea legal y recomendable, podemos encontrarnos con otro tanto poco recomendable e incluso dañino para los ordenadores, por ejemplo, hay páginas Web que contienen virus, troyanos, malware, adware, contenido ilegal, etc.

Los usuarios de la WWW deben de ser libres de navegar por los sitios que quieran, pero en organizaciones como en las mencionadas anteriormente puede ser bueno tener esto un poco controlado, ya que la cantidad de accesos puede ser muy grande y los usuarios pueden navegar por sitios peligrosos o poco recomendables aunque ellos mismos no sean conscientes de ello. Por otro lado también podemos encontrarnos con usuarios que abusen del acceso a Internet y mientras están en horario de trabajo o de aprendizaje se dediquen a visitar páginas Web de ocio u otros contenidos que no tengan nada que ver con su dedicación, con lo cual están haciendo un mal uso de la conexión que les está brindando la organización.

2. Objetivos del Proyecto.

Para abordar esto, la idea es poder tener controlados los accesos de los usuarios de la organización a ciertas páginas Web que los administradores de la misma consideren que puedan ser dañinas o poco apropiadas para la organización. Hay que hacerlo de manera poco invasiva y transparente al usuario, es decir, dejamos a los usuarios navegar por las páginas que quieran, pero si acceden a alguna que los administradores consideren no adecuada, este acceso debe quedar registrado. Si hay que tomar medidas o no una vez analizados los accesos a los sitios marcados, lo decidirán después los administradores de la red. A través de esta idea, voy a desarrollar mi proyecto; lo ideal es crear una herramienta para controlar los accesos a ciertas páginas Web de una pequeña-mediana organización.

3. Infraestructura Informática. Software utilizado.

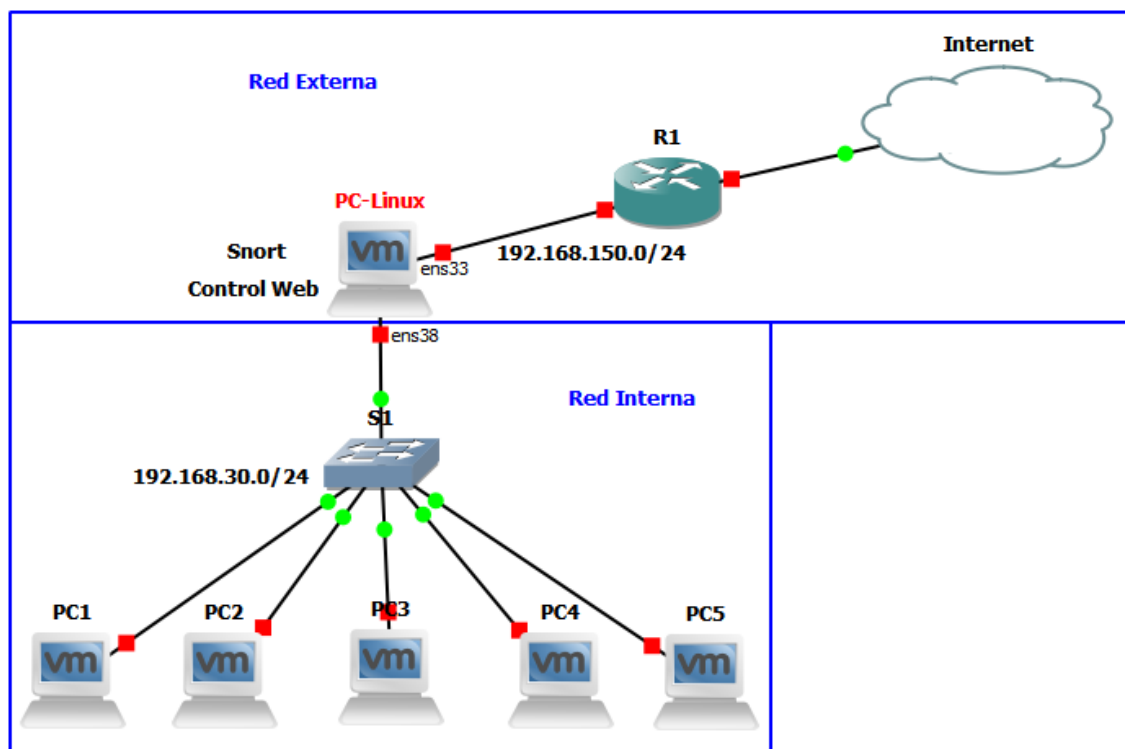
Para ello voy a utilizar una máquina con 2 tarjetas de red y la distribución de Linux Ubuntu 16.04 LTS instalada en ella, la cual actuará como puerta de enlace de la organización y dará salida hacia Internet a la misma, por lo tanto todo el tráfico de red tiene que pasar obligatoriamente por dicha máquina. Para poder controlar el tráfico que pase por la máquina instalaré el IDS/IPS de código abierto Snort, con el cual a través de la generación de reglas y su sistema de alertas podremos tener controlados los accesos a las páginas Web que queramos.

Para facilitar la gestión de las reglas de Snort y los usuarios que tendrán acceso para crearlas y eliminarlas será necesario diseñar una aplicación Web que nos facilite el trabajo, para ello utilizaré Apache como servidor Web, MySQL-Server como sistema gestor de bases de datos y utilizaré los lenguajes de programación HTML, CSS, JavaScript y PHP, de los cuales me serviré para el desarrollo de dicha aplicación.

Ubuntu, Apache, MySQL y PHP forman lo que se conoce comúnmente como la pila LAMP, que es una infraestructura de servidor Web, utilizando un paradigma de programación para el desarrollo a través del código libre.

3.1. Escenario informático de red.

Como se puede observar en la imagen, he dividido el esquema de la red en 2 partes bien diferenciadas; red interna y red externa, esto es debido a que Snort trabaja de esta manera, diferenciando los paquetes que entran y salen de la red interna hacia la externa y viceversa. En la red interna se encuentran todos los equipos de la organización y el Switch que comunica a todos ellos con la tarjeta de red del PC Linux que actúa como puerta de enlace. Entre las 2 tarjetas de red del PC Linux se encuentra el perímetro de la red interna, a partir de la tarjeta de red que conecta con el Router frontera de la organización se considerará red externa para Snort.



3.2. Sistema Operativo Ubuntu.

Ubuntu es un sistema operativo de código abierto para ordenadores. Es una distribución de Linux basada en la arquitectura de Debian. Actualmente corre en ordenadores de escritorio y servidores, en arquitecturas Intel, AMD y ARM. Está orientado al usuario promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia del usuario. Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto. Las estadísticas sugieren que la cuota de mercado de Ubuntu dentro de las distribuciones Linux es, aproximadamente, del 52%, y con una tendencia a aumentar como servidor web. Su patrocinador, Canonical, ofrece el sistema de manera gratuita, y se financia por medio de servicios vinculados al sistema operativo y vendiendo soporte técnico. Además, al mantenerlo libre y gratuito, la empresa es capaz de aprovechar a los desarrolladores de la comunidad para mejorar los componentes de su sistema operativo. Cada seis meses se publica una nueva versión de Ubuntu, esta recibe soporte por parte de Canonical durante nueve meses por medio de actualizaciones de seguridad, parches para bugs críticos y actualizaciones menores de programas. Las versiones LTS (Long Term Support), que se liberan cada dos años, reciben soporte durante cinco años en los sistemas de escritorio y de servidor.

3.3. Snort.

Una de las herramientas fundamentales del proyecto es Snort, ya que gran parte del mismo gira en torno a dicho software. Snort es un Sistema de Detección/Prevención de Intrusos (IDS/IPS) basado en red de código abierto. Cuenta con un lenguaje de creación de reglas en el que se pueden definir los patrones que se utilizarán a la hora de monitorizar el sistema. Además, ofrece una serie de reglas y filtros ya predefinidos que se pueden ajustar durante su instalación y configuración para que se adapte lo máximo posible a lo que deseamos. Una de las ventajas de este sistema es que puede funcionar como Sniffer, ya que podemos ver en consola y en tiempo real qué ocurre en nuestra red, todo nuestro tráfico, registro de paquetes, etc. y permite guardar en una serie de archivos los logs generados a partir de estos para su posterior análisis. Snort es una herramienta muy completa que permite controlar de muchas formas aquellos paquetes que son de interés. Esto puede ser de gran importancia si se desea controlar el tráfico de red dentro de un entorno corporativo, lo cual es ideal para llevar a cabo la idea del proyecto.

Las reglas de Snort se crean para buscar patrones dentro de los paquetes de datos que van atravesando la red, estas reglas son utilizadas por el motor de detección para comparar los paquetes recibidos y generar las alertas en caso de existir coincidencia entre el contenido de los paquetes y las reglas. Snort permite mucha versatilidad para crear reglas nuevas, la idea es crearlas en función de los servicios que se desean monitorizar.

El fichero de texto donde se escriben y guardan las reglas de Snort generadas por los usuarios es `/etc/snort/rules/local.rules`.

Las reglas se pueden dividir en dos secciones lógicas; cabecera de la regla y opciones:

- La cabecera contiene la acción de la regla en sí; protocolo, IP's, máscaras de red, puertos de origen, puertos de destino y destino del paquete o dirección de la operación.
- La sección de opciones contiene los mensajes y la información necesaria para la decisión a tomar por parte de la alerta en forma de opciones.

A continuación expongo un ejemplo de regla para alertar de un acceso a la URL www.youtube.com, la parte en azul corresponde a la cabecera y la parte en verde corresponde a las opciones.

```
alert tcp $HOME_NET any -> any any (msg: "Acceso a Youtube!"; content: "www.youtube.com"; sid:3; rev:3;)
```

Cada regla tiene sus propios números identificación lógicamente para diferenciar unas de otras, estos se indican mediante el sid y rev como se puede observar, cada regla tiene que llevar sus propios identificadores y todos tienen que ser diferentes.

3.4. Apache, MySQL, PHP, phpMyAdmin.

- Apache es un servidor web HTTP de código abierto, para plataformas Unix, Windows y Macintosh, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Tiene una amplia aceptación en la red: desde 1996, Apache es el servidor HTTP más usado, jugó un papel fundamental en el desarrollo de la World Wide Web y alcanzó su máxima cuota de mercado en 2005, siendo el servidor empleado en el 70% de los sitios web en el mundo.

- MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

- PHP es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo para que procese los datos. El código es interpretado por un servidor Web con un módulo de procesador de PHP que genera el HTML resultante.

- phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de un navegador web. Actualmente puede crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 72 idiomas. Se encuentra disponible bajo la licencia GPL Versión 2.

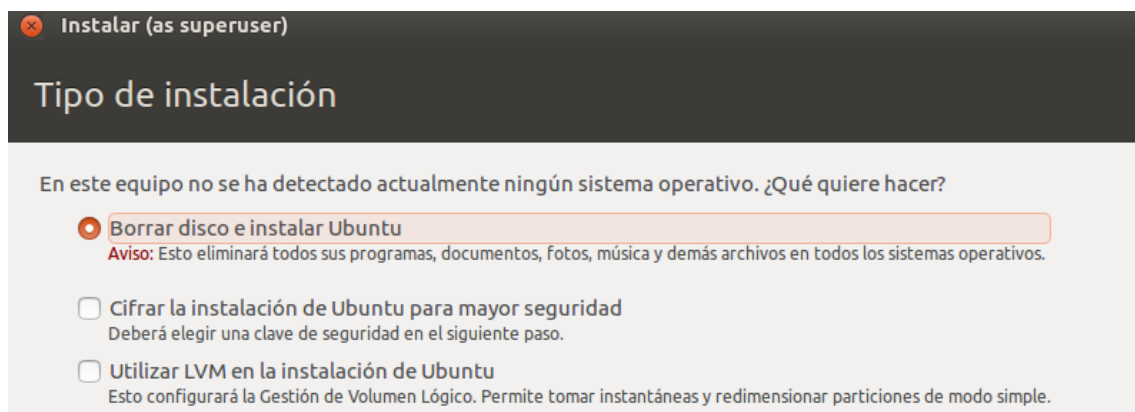
4. Instalación y Configuración.

4.1. Ubuntu.

El primer paso en el desarrollo del proyecto es la instalación del sistema operativo Ubuntu 16.04.3 LTS en la máquina que conectará a los equipos de la red interna con el exterior.



No considero necesario crear particiones en este caso, por lo que ocupo todo el disco con la instalación del sistema.



Configuración del superusuario y nombre de la máquina.

Instalar (as superuser)

¿Quién es usted?

Su nombre: ✓

El nombre de su equipo: ✓
El nombre que usa cuando habla con otros equipos.

Introduzca un nombre de usuario: ✓

Introduzca una contraseña: **Contraseña corta**

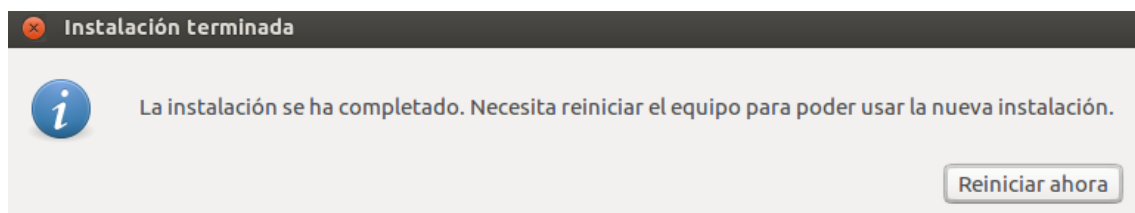
Confirme su contraseña: ✓

☒ Iniciar sesión automáticamente

☐ Solicitar mi contraseña para iniciar sesión

☐ Cifrar mi carpeta personal

Finaliza la instalación.



Configuración de las 2 tarjetas de red en el fichero /etc/network/interfaces: ens33 salida hacia Internet, ens38 red interna.

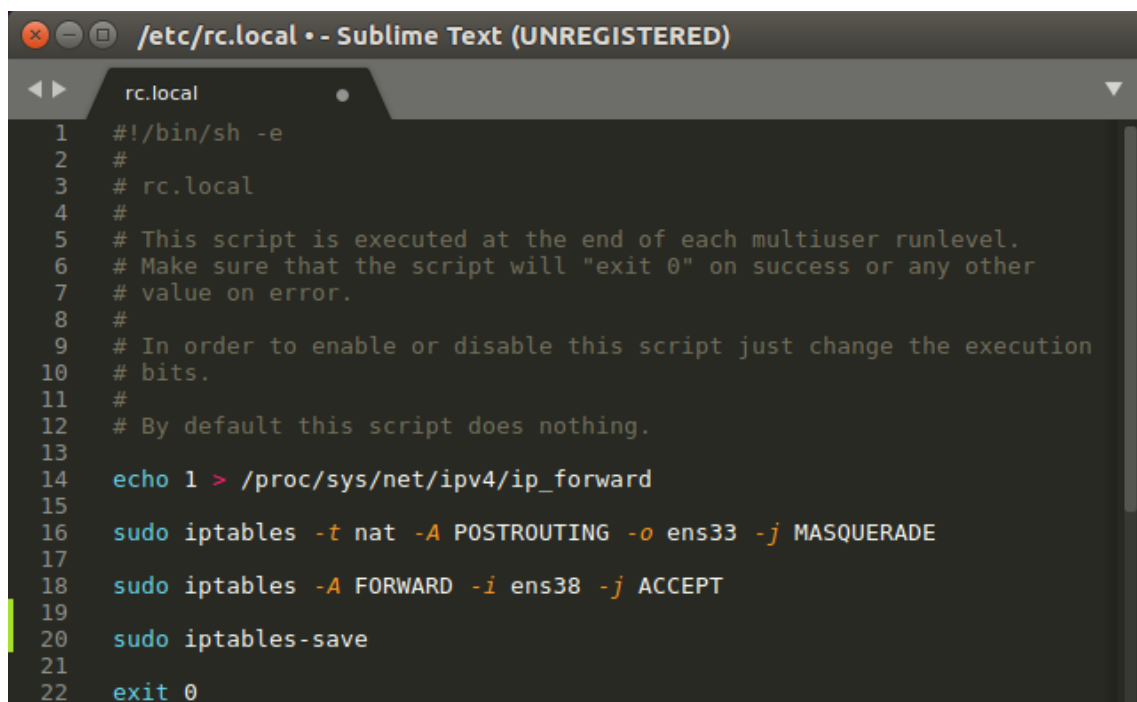
```
interfaces
1 # interfaces(5) file used by ifup(8) and ifdown(8)
2 auto lo
3 iface lo inet loopback
4
5 # Internet
6 auto ens33
7 iface ens33 inet dhcp
8
9 # Red Interna
10 auto ens38
11 iface ens38 inet static
12 address 192.168.30.1
13 netmask 255.255.255.0
```


Configuración de los DNS en el fichero /etc/resolv.conf.

```
GNU nano 2.5.3      Archivo: /etc/resolv.conf

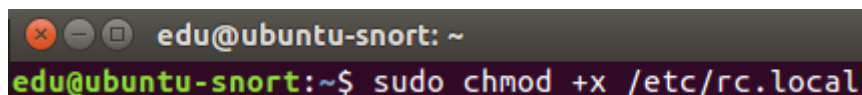
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolv$
#      DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.1.1
search localdomain
nameserver 1.1.1.1
nameserver 1.0.0.1
```

Una vez configurado todo lo referente a la red, creo un script en el fichero /etc/rc.local para la activación del enrutamiento y el NAT al iniciar el PC, de esta manera daremos salida a Internet a los equipos de la red interna.



```
/etc/rc.local - Sublime Text (UNREGISTERED)
rc.local
1  #!/bin/sh -e
2  #
3  # rc.local
4  #
5  # This script is executed at the end of each multiuser runlevel.
6  # Make sure that the script will "exit 0" on success or any other
7  # value on error.
8  #
9  # In order to enable or disable this script just change the execution
10 # bits.
11 #
12 # By default this script does nothing.
13
14 echo 1 > /proc/sys/net/ipv4/ip_forward
15
16 sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
17
18 sudo iptables -A FORWARD -i ens38 -j ACCEPT
19
20 sudo iptables-save
21
22 exit 0
```

Asigno los pertinentes permisos de ejecución sobre el mismo.



```
edu@ubuntu-snort: ~
edu@ubuntu-snort:~$ sudo chmod +x /etc/rc.local
```

4.2. Snort.

Primero se instalan las librerías necesarias para que Snort funcione correctamente.

```
edu@ubuntu-snort: ~  
edu@ubuntu-snort:~$ sudo apt install -y gcc libpcap-dev zlib1g-dev lib  
lua5.1-dev libpcap-dev openssl libssl-dev libnghttp2-dev libdumbnet  
-dev bison flex libdnet|
```

A continuación descargo el código fuente, lo compilo y lo instalo en el equipo.

```
edu@ubuntu-snort: ~/snort_src  
edu@ubuntu-snort:~$ mkdir ~/snort_src && cd ~/snort_src  
edu@ubuntu-snort:~/snort_src$ wget https://www.snort.org/downloads/snor  
t/daq-2.0.6.tar.gz|
```

```
edu@ubuntu-snort:~/snort_src$ tar -xvzf daq-2.0.6.tar.gz
```

```
edu@ubuntu-snort:~/snort_src$ cd daq-2.0.6  
edu@ubuntu-snort:~/snort_src/daq-2.0.6$ ./configure && make && sudo mak  
e install
```

```
edu@ubuntu-snort:~$ cd ~/snort_src  
edu@ubuntu-snort:~/snort_src$ wget https://www.snort.org/downloads/snor  
t/snort-2.9.13.tar.gz
```

```
edu@ubuntu-snort:~/snort_src$ tar -xvzf snort-2.9.13.tar.gz
```

```
edu@ubuntu-snort:~/snort_src$ cd snort-2.9.13  
edu@ubuntu-snort:~/snort_src/snort-2.9.13$ ./configure --enable-sourcef  
ire && make && sudo make install
```

Una vez instalado, configuro Snort en modo IDS.

```
edu@ubuntu-snort:~$ sudo ldconfig  
edu@ubuntu-snort:~$ sudo ln -s /usr/local/bin/snort /usr/sbin/snort
```

Por seguridad, es mejor utilizar un usuario y grupo específicos para ejecutar Snort.

```
edu@ubuntu-snort:~$ sudo groupadd snort  
edu@ubuntu-snort:~$ sudo useradd snort -r -s /sbin/nologin -c SNORT_IDS  
-g snort
```

Creo la estructura necesaria de directorios para los ficheros de log y los ficheros de reglas.

```
edu@ubuntu-snort:~$ sudo mkdir -p /etc/snort/rules  
edu@ubuntu-snort:~$ sudo mkdir /var/log/snort  
edu@ubuntu-snort:~$ sudo mkdir /usr/local/lib/snort_dynamicrules
```

Les asigno el propietario y grupo creados anteriormente y los permisos necesarios.

```
edu@ubuntu-snort:~$ sudo chmod -R 5775 /etc/snort
edu@ubuntu-snort:~$ sudo chmod -R 5775 /var/log/snort
edu@ubuntu-snort:~$ sudo chmod -R 5775 /usr/local/lib/snort_dynamicrules
edu@ubuntu-snort:~$ sudo chown -R snort:snort /etc/snort
edu@ubuntu-snort:~$ sudo chown -R snort:snort /var/log/snort
edu@ubuntu-snort:~$ sudo chown -R snort:snort /usr/local/lib/snort_dynamicrules
```

Creo los ficheros para la lista blanca, la lista negra y las reglas locales. El fichero de reglas locales será el que utilizarán los usuarios para crear las reglas personalizadas.

```
edu@ubuntu-snort:~$ sudo touch /etc/snort/rules/white_list.rules
edu@ubuntu-snort:~$ sudo touch /etc/snort/rules/black_list.rules
edu@ubuntu-snort:~$ sudo touch /etc/snort/rules/local.rules
```

Ahora hay que copiar los ficheros de configuración del directorio de descargas a su ubicación definitiva en /etc/snort.

```
edu@ubuntu-snort:~$ sudo cp ~/snort_src/snort-2.9.13/etc/*.conf* /etc/snort
edu@ubuntu-snort:~$ sudo cp ~/snort_src/snort-2.9.13/etc/*.map /etc/snort
```

En este punto Snort ya está instalado y preparado para funcionar. A continuación voy a descargar las reglas que va actualizando la comunidad de Snort para detectar intrusiones y las voy a añadir a mi instalación para dar mayor seguridad a la red interna.



```
edu@ubuntu-snort: ~
edu@ubuntu-snort:~$ wget https://www.snort.org/rules/community -O ~/community.tar.gz
edu@ubuntu-snort:~$ sudo tar -xvf ~/community.tar.gz -C ~/
edu@ubuntu-snort:~$ sudo cp ~/community-rules/* /etc/snort/rules
```

4.2.1 Configuración de Snort.

Ahora que tengo Snort instalado y preparado el siguiente paso es configurarlo, para ello edito su fichero de configuración /etc/snort/snort.conf.

Primero defino la red interna del escenario, la dirección de red que le voy a asignar es la 192.168.30.0/24 y todo lo que no sea esta red se va a considerar red externa.

```
43
44 # Setup the network addresses you are protecting
45 ipvar HOME_NET 192.168.30.0/24
46
47 # Set up the external network addresses. Leave as "any" in most situations
48 ipvar EXTERNAL_NET !$HOME_NET
```

Indico la ruta hacia los ficheros de las reglas de la lista blanca y de la lista negra.

```
101 # Path to your rules files (this can be a relative path)
102 # Note for Windows users: You are advised to make this an absolute path,
103 # such as: c:\snort\rules
104 var RULE_PATH /etc/snort/rules
105 var SO_RULE_PATH /etc/snort/so_rules
106 var PREPROC_RULE_PATH /etc/snort/preproc_rules
107
108 # If you are using reputation preprocessor set these
109 # Currently there is a bug with relative paths, they are relative to where
110 # snort is
111 # not relative to snort.conf like the above variables
112 # This is completely inconsistent with how other vars work, BUG 89986
113 # Set the absolute path appropriately
114 var WHITE_LIST_PATH /etc/snort/rules
115 var BLACK_LIST_PATH /etc/snort/rules
```

Defino el nombre del log donde se guardarán las capturas de las alertas.

```
519 # unified2
520 # Recommended for most installs
521 # output unified2: filename merged.log, limit 128, nostamp, mpls_event_types,
522 # output unified2: filename snort.log, limit 128
```

Habilito los ficheros de reglas locales y reglas de la comunidad descomentando su línea correspondiente.

```
538 #####
539 # Step #7: Customize your rule set
540 # For more information, see Snort Manual, Writing Snort Rules
541 #
542 # NOTE: All categories are enabled in this conf file
543 #####
544
545 # site specific rules
546 include $RULE_PATH/local.rules
547 include $RULE_PATH/community.rules
```

En este punto Snort está configurado, para comprobar que todo está correcto hago un testeo al fichero de configuración con el siguiente comando.

```
edu@ubuntu-snort:~$ sudo snort -T -c /etc/snort/snort.conf
```

Como se puede observar Snort está correctamente configurado dado que recibimos el siguiente mensaje.

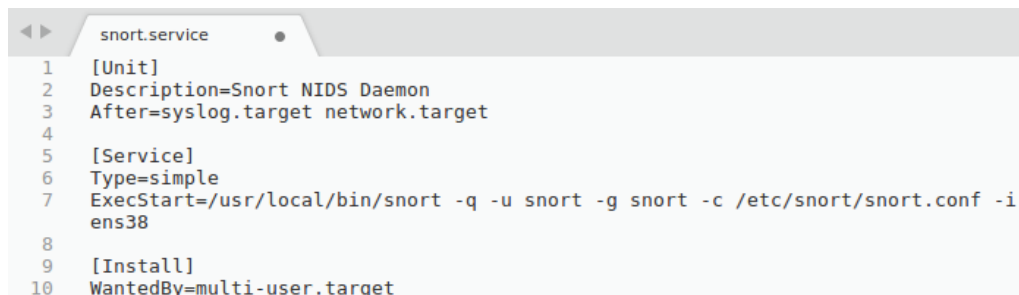
```
--== Initialization Complete ==--

o''~  -> Snort! <*-
''''~  Version 2.9.13 GRE (Build 15013)
By Martin Roesch & The Snort Team: http://www.snort.org/cont
act#team
Copyright (C) 2014-2019 Cisco and/or its affiliates. All rig
hts reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.7.4
Using PCRE version: 8.38 2015-11-23
Using ZLIB version: 1.2.8

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.1 <Build
1>
```

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

Para completar todo el proceso de configuración falta que Snort funcione como un servicio del sistema, por lo tanto lo creo a mano en /lib/systemd/system/snort.service.



```
snort.service
1 [Unit]
2 Description=Snort NIDS Daemon
3 After=syslog.target network.target
4
5 [Service]
6 Type=simple
7 ExecStart=/usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i
  ens38
8
9 [Install]
10 WantedBy=multi-user.target
```

Recargo el demonio systemd.

```
edu@ubuntu-snort:~$ sudo systemctl daemon-reload
```

Snort ya se puede arrancar, parar y reiniciar como cualquier otro servicio del sistema.

```
edu@ubuntu-snort:~$ sudo service snort restart
edu@ubuntu-snort:~$ sudo service snort status
● snort.service - Snort NIDS Daemon
   Loaded: loaded (/lib/systemd/system/snort.service; disabled; vendor
   Active: active (running) since dom 2019-04-14 11:58:54 CEST; 5s ago
   Main PID: 33225 (snort)
   CGroup: /system.slice/snort.service
           └─33225 /usr/local/bin/snort -q -u snort -g snort -c /etc/sn
```

```
abr 14 11:58:54 ubuntu-snort systemd[1]: Started Snort NIDS Daemon.
lines 1-8/8 (END)
```

Para facilitar la iniciación de captura de paquetes creo un script donde introduzco el comando correspondiente en la ruta /usr/local/bin y le asigno permisos ejecución.

```
snort.sh
1  #!/bin/bash
2
3  echo 'inves' | /usr/bin/sudo -S snort -A console -i ens38 -u edu -g edu -c /etc/snort/snort.conf -K ascii
4
```

```
edu@ubuntu-snort:~$ sudo chmod +x /usr/local/bin/snort.sh
[sudo] password for edu:
edu@ubuntu-snort:~$
```

Creo un acceso directo en /usr/share/applications apuntando a dicho script.

```
/usr/share/applications/snort-console.desktop - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

snort-console.desktop
1  [Desktop Entry]
2  Name=Snort
3  Comment=Consola de Snort
4  Exec=/usr/local/bin/snort.sh
5  Icon=/home/edu/Imágenes/logo_snort.png
6  Terminal=true
7  Type=Application
8
```

Ya puedo acceder a él desde el Dash de Ubuntu e incluso colocarlo en el lanzador de aplicaciones.



```
o''_~
o''_~
o''_~
act#team

-*> Snort! <*-
Version 2.9.13 GRE (Build 15013)
By Martin Roesch & The Snort Team: http://www.snort.org/cont
Copyright (C) 2014-2019 Cisco and/or its affiliates. All rig
hts reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.7.4
Using PCRE version: 8.42 2018-03-20
Using ZLIB version: 1.2.8
```

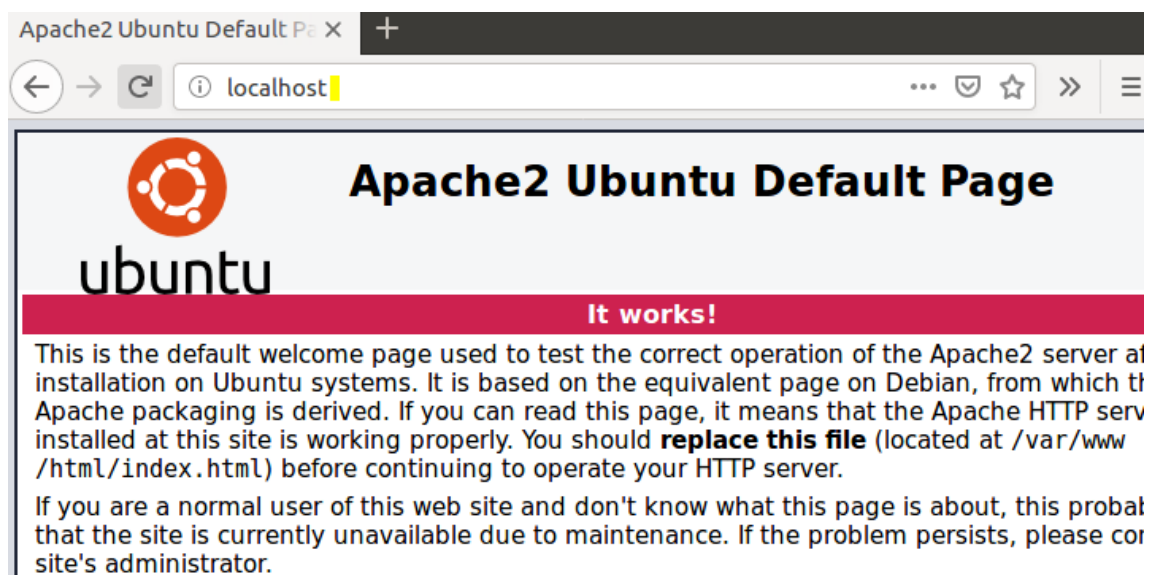

4.3. Apache.

Con Snort instalado y configurado, ahora procedo con la instalación del servidor Web Apache sobre el cual haré funcionar la aplicación Web para la gestión de las reglas Snort de manera más sencilla y eficiente.

```
edu@ubuntu-snort:~$ sudo apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática
y ya no son necesarios.
  liblvm4.0 snapd-login-service
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0
```

Una vez finalizado el proceso, compruebo que el sitio por defecto está activo y en funcionamiento.

```
edu@ubuntu-snort:~$ service apache2 status
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since dom 2019-04-14 18:59:38 CEST; 17s ago
     Docs: man:systemd-sysv-generator(8)
   CGroup: /system.slice/apache2.service
           └─3778 /usr/sbin/apache2 -k start
             3780 /usr/sbin/apache2 -k start
             3781 /usr/sbin/apache2 -k start
```

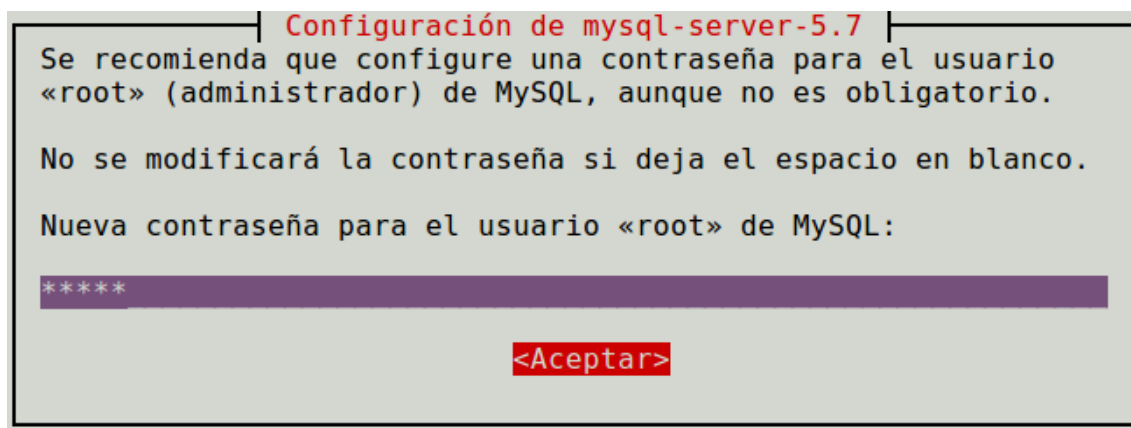


4.4. MySQL.

El siguiente paquete a instalar es MySQL-Server ya que utilizaré una base de datos para la aplicación Web. En dicha base de datos guardaré a los usuarios que van a tener acceso a la aplicación con sus correspondientes niveles de acceso y también las reglas de Snort que creen los mismos para después volcarlas en el fichero /etc/snort/rules/rules.local mediante PHP.

```
edu@ubuntu-snort:~$ sudo apt-get install mysql-server mysql-client
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática
y ya no son necesarios.
  liblvm4.0 snapd-login-service
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libaio1 libevent-core-2.0-5 libhtml-template-perl mysql-client-5.7
  mysql-client-core-5.7 mysql-common mysql-server-5.7
  mysql-server-core-5.7
```

Se me solicita la contraseña del usuario administrador para MySQL, la cual introduzco 2 veces durante el proceso.



Una vez terminada la instalación, compruebo si el servicio está activo y en funcionamiento.

```
edu@ubuntu-snort:~$ service mysql status
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor p
   Active: active (running) since dom 2019-04-14 19:05:40 CEST; 1min 42
   Main PID: 7985 (mysqld)
     CGroup: /system.slice/mysql.service
             └─7985 /usr/sbin/mysqld

abr 14 19:05:37 ubuntu-snort systemd[1]: Starting MySQL Community Serve
abr 14 19:05:40 ubuntu-snort systemd[1]: Started MySQL Community Server
lines 1-9/9 (END)
```


4.5. PHP.

Para la lógica de programación de la aplicación Web voy a utilizar la última versión estable de PHP a día de hoy, es decir la 7.2.

Primero añado el repositorio.

```
edu@ubuntu-snort:~$ sudo add-apt-repository ppa:ondrej/php
Co-installable PHP versions: PHP 5.6, PHP 7.x and most requested extensions are included. Only Supported Versions of PHP (http://php.net/supported-versions.php) for Supported Ubuntu Releases (https://wiki.ubuntu.com/Releases) are provided. Don't ask for end-of-life PHP versions or Ubuntu release, they won't be provided.
```

A continuación hago un apt-get update para que el sistema agregue dicho repositorio a su lista interna.

```
edu@ubuntu-snort:~$ sudo apt-get update
Des:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Des:2 http://ppa.launchpad.net/ondrej/php/ubuntu xenial InRelease [23,9 kB]
Obj:3 http://es.archive.ubuntu.com/ubuntu xenial InRelease
Des:4 http://es.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Des:6 http://ppa.launchpad.net/ondrej/php/ubuntu xenial/main amd64 Packages [52,8 kB]
Des:7 http://ppa.launchpad.net/ondrej/php/ubuntu xenial/main i386 Packages [52,6 kB]
Des:8 http://ppa.launchpad.net/ondrej/php/ubuntu xenial/main Translation-en [29,0 kB]
Descargados 483 kB en 2s (184 kB/s)
Leyendo lista de paquetes... Hecho
```

Procedo con la instalación de PHP 7.2.

```
edu@ubuntu-snort:~$ sudo apt-get install php7.2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libllvm4.0 snapd-login-service
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libapache2-mod-php7.2 libargon2-0 libpcre16-3 libpcre3 libpcre3-dev
  libpcre32-3 libpcrecpp0v5 libsodium23 libssl1.1 php-common
  php7.2-cli php7.2-common php7.2-json php7.2-opcache php7.2-readline
```

También instalo algunos módulos adicionales que podrían ser necesarios.

```
edu@ubuntu-snort:~$ sudo apt-get install php-pear php7.2-curl php7.2-dev php7.2-gd php7.2-mbstring php7.2-zip php7.2-mysql php7.2-xml
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  liblvm4.0 snapd-login-service
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  autoconf automake autotools-dev debhelper dh-strip-nondeterminism
  libfile-stripnondeterminism-perl libltdl-dev libmail-sendmail-perl
  libsys-hostname-long-perl libtool libzip4 pkg-php-tools po-debconf
  shtool
```

Para concluir compruebo que la versión instalada es la correcta.

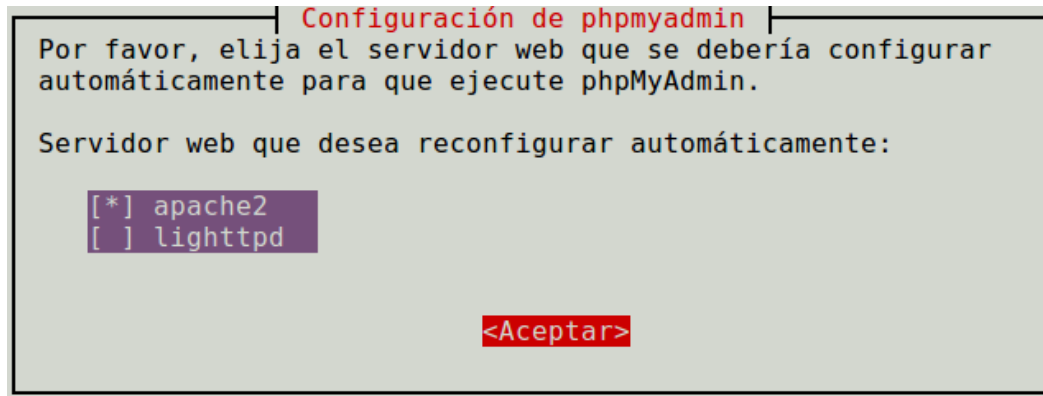
```
edu@ubuntu-snort:~$ php -v
PHP 7.2.17-1+ubuntu16.04.1+deb.sury.org+3 (cli) (built: Apr 10 2019 10:50:19) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.2.17-1+ubuntu16.04.1+deb.sury.org+3, Copyright (c) 1999-2018, by Zend Technologies
```

[4.6. phpMyAdmin.](#)

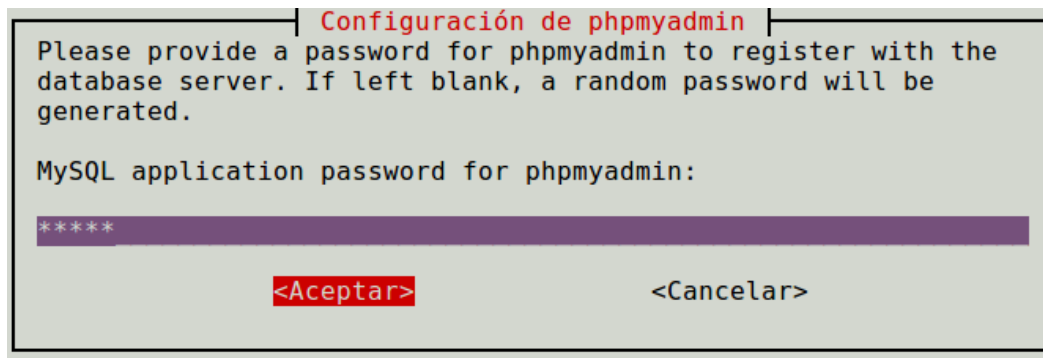
Para facilitarme un poco la gestión de la base de datos voy a instalar también phpMyAdmin, este paquete me ayudará a operar con ella mediante un entorno Web.

```
edu@ubuntu-snort:~$ sudo apt-get install phpmyadmin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  liblvm4.0 snapd-login-service
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  dbconfig-common dbconfig-mysql javascript-common libjs-jquery
  libjs-sphinxdoc libjs-underscore libmcrypt4 php-gettext php-mcrypt
  php-phpseclib php-tcpdf php7.0-common php7.0-mcrypt
Paquetes sugeridos:
  libmcrypt-dev mcrypt php-libsodium php-gmp php-imagick
Se instalarán los siguientes paquetes NUEVOS:
  dbconfig-common dbconfig-mysql javascript-common libjs-jquery
  libjs-sphinxdoc libjs-underscore libmcrypt4 php-gettext php-mcrypt
  php-phpseclib php-tcpdf php7.0-common php7.0-mcrypt phpmyadmin
0 actualizados, 14 nuevos se instalarán, 0 para eliminar y 7 no actualizados.
```

Elijo Apache como servidor Web para ejecutarlo ya que es el que he instalado anteriormente.

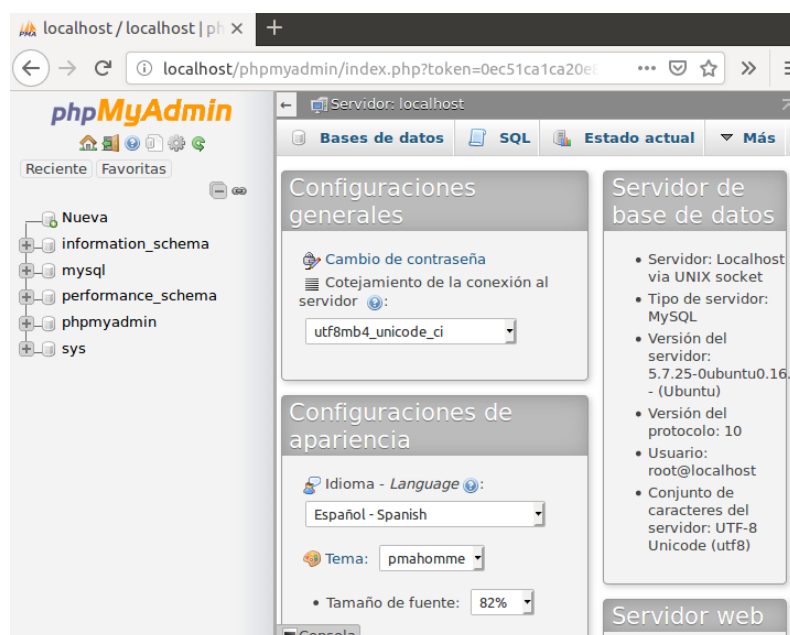


Introduzco la contraseña de administrador para después poder acceder a la aplicación.



Una vez terminada la instalación, reinicio Apache y compruebo que puedo entrar correctamente. Ya tengo el equipo preparado para desarrollar la aplicación Web.

```
edu@ubuntu-snort:~$ sudo service apache2 restart
```



5. Base de datos implementada.

Para la gestión de la aplicación Web creo la base de datos snort, se compone de únicamente 2 tablas, una que contiene a los usuarios de la aplicación y otra que contiene las opciones de las alertas que después se vuelcan en el fichero de reglas de Snort /etc/snort/rules/local.rules.

Diagrama E/R.

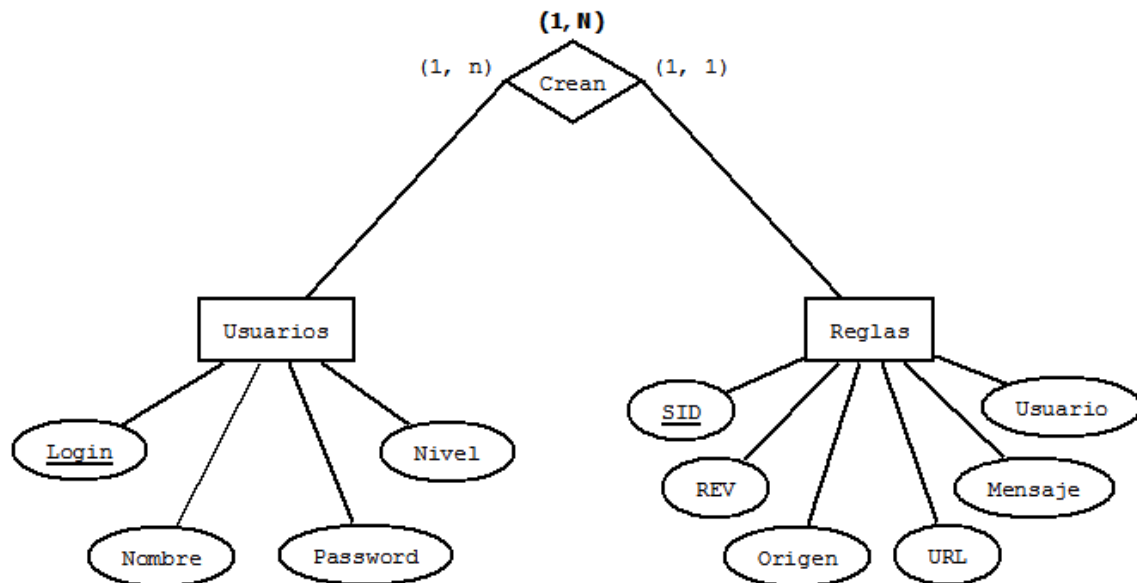




Tabla de usuarios.

| + Opciones | | | | | | | |
|--------------------------|----------|----------|-----------|---------|----------|----------------------------------|-------|
| ← T → | | | | | | | |
| | | | login | nombre | password | | nivel |
| <input type="checkbox"/> | ✎ Editar | 📋 Copiar | 🗑️ Borrar | eduardo | Eduardo | 3fbaab6bbeea838d46186119a7ed977c | 2 |
| <input type="checkbox"/> | ✎ Editar | 📋 Copiar | 🗑️ Borrar | javier | Javier | 247da55b11abc312ac64ab1f0c96d238 | 1 |
| <input type="checkbox"/> | ✎ Editar | 📋 Copiar | 🗑️ Borrar | laura | Laura | 247da55b11abc312ac64ab1f0c96d238 | 1 |

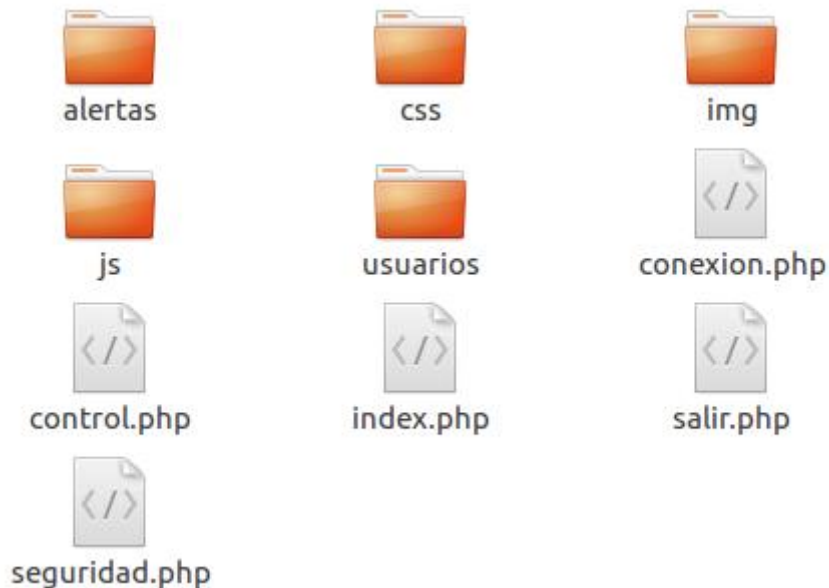
Tabla de reglas.

| + Opciones | | | | | | | | | |
|--------------------------|--|--|--|-----|-----|------------|--------------|---------------------|---------|
| ← T → | | | | sid | rev | origen | url | mensaje | usuario |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 7 | 7 | \$HOME_NET | www.as.com | Acceso al diario AS | javier |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 8 | 8 | \$HOME_NET | www.marca.es | Acceso a Marca | eduardo |
| <input type="checkbox"/> |  Editar |  Copiar |  Borrar | 9 | 9 | \$HOME_NET | www.hbo.com | Acceso a HBO | laura |

6. Desarrollo de la aplicación web.

He llamado a la aplicación “Control Web”, a continuación se detalla la estructura de ficheros de la misma con el código y la función que realiza cada uno.

Carpeta raíz situada en /var/www/html.



- index.php -> página de entrada a la aplicación Web, contiene un formulario para introducir las credenciales de usuario para el acceso a la misma.

- conexión.php -> fichero que contiene los datos de conexión con la base de datos.

- control.php -> controla que el usuario introducido en el formulario del index.php exista en la base de datos, si existe se inicia una sesión PHP y lo manda a la página de inicio de las alertas, si no existe lo vuelve a mandar al index.php marcando un error de acceso.

- seguridad.php -> fichero de seguridad de la aplicación, comprueba si el usuario que accede a la aplicación está logueado correctamente, si lo está le brinda acceso a la página. Cuando hay 5 minutos de inactividad lo devuelve al index.php por seguridad para que se vuelva a loguear. Si el usuario no está logueado entonces lo manda al index.php directamente.

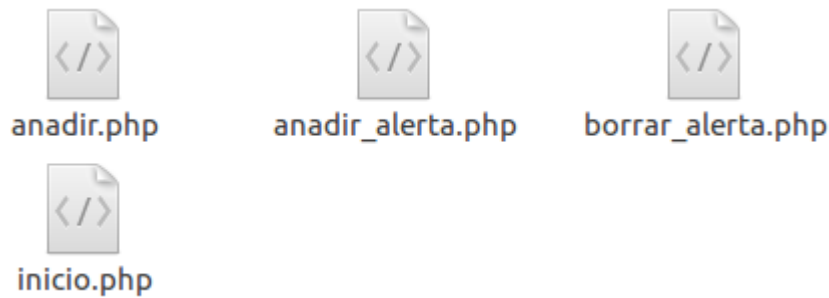
- salir.php -> destruye la sesión de usuario PHP y le manda al index.php.

- La carpeta js contiene los ficheros JavaScript necesarios para la comprobación de los formularios de la aplicación.

- La carpeta css contiene las hojas de estilo de la aplicación.

- La carpeta img contiene las imágenes que aparecen en la aplicación.

Carpeta de alertas.



- inicio.php -> página de inicio de la aplicación una vez que el usuario está logueado correctamente, hace una consulta a la base de datos para sacar las alertas que están activas y las muestra por pantalla.

- anadir_alerta.php -> contiene el formulario para la creación de manera sencilla de las alertas Snort para monitorizar URL's, una vez los datos de la alerta son recogidos estos se envían al script anadir.php.

- anadir.php -> script más importante de la aplicación, primero hace una consulta a la base de datos para calcular cual es el último número de sid y rev de las alertas existentes, es decir los números de identificación de las mismas, a los últimos que existan se les suma uno, entonces estos serán los sid y rev de la nueva alerta a crear. Si no existieran alertas ya creadas en la base de datos entonces los inicializa a 1. A continuación comprueba que la URL introducida no esté ya monitorizada en otra regla, si lo está nos devuelve a anadir_alerta.php marcando un error. Una vez que está todo correcto inserta los datos de la alerta en la base de datos, después vuelca en el archivo /etc/snort/rules/local.rules todas las alertas que hay en la base de datos, una vez hecho esto se reinicia el servicio Snort para hacerlas efectivas y nos devuelve a anadir_alerta.php marcando éxito en la creación de la alerta.

- borrar_alerta.php -> hace una consulta a la base de datos para sacar las alertas activas por pantalla junto a un checkbox para marcarlas si queremos borrarlas. Una vez pulsado el botón de borrar, las alertas marcadas se eliminan de la base de datos y se vuelcan las restantes en el fichero /etc/snort/rules/local.rules, por último se reinicia el servicio Snort para que se hagan efectivos los cambios.

Carpeta de usuarios.



Esta sección solo está permitida para usuarios con nivel 2, estos son superadministradores con privilegios para crear y borrar a usuarios que tienen acceso a la aplicación Web.

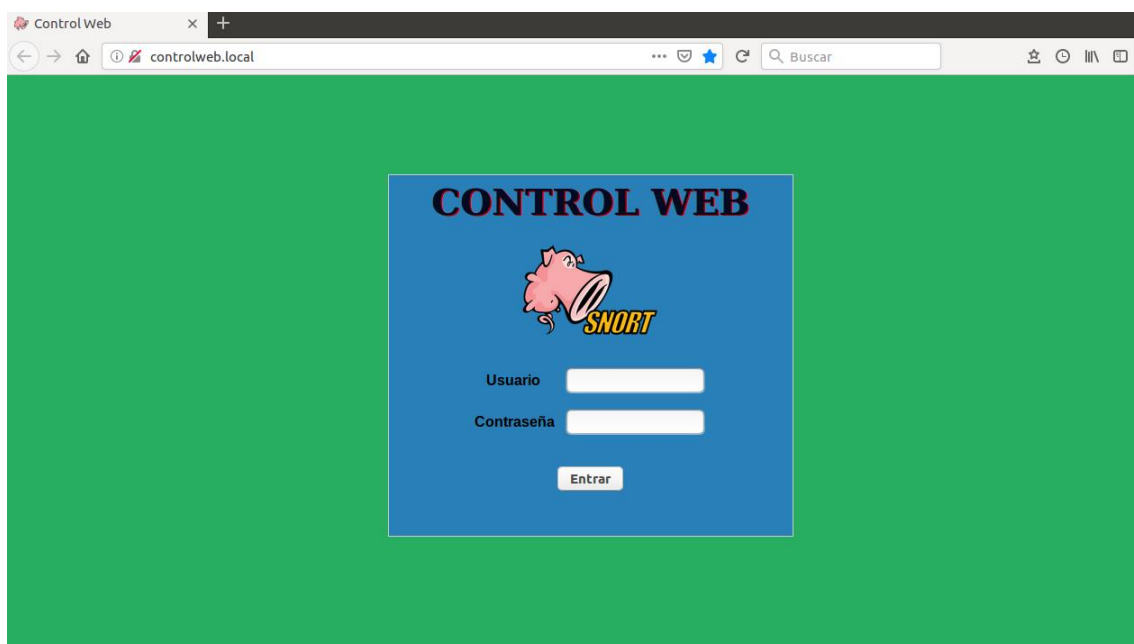
- crear_usuario.php -> contiene un formulario para rellenar los datos del usuario a crear, una vez estos son recogidos se envían al script crear.php.

- crear.php -> este script recibe los datos del usuario a crear del formulario de crear_usuario.php y los inserta en la base de datos cifrando la contraseña en MD5 por seguridad.

- borrar_usuario.php -> hace una consulta a la base de datos para presentar por pantalla a los usuarios de la aplicación existentes junto a un botón de radio. Una vez se selecciona uno para eliminar y se pulsa el botón este se borra de la base de datos. Como al borrar al usuario todas sus reglas se borran en cascada porque lo tengo así diseñado, se vuelven a escribir las restantes en el fichero /etc/snort/rules/local.rules y por último se reinicia el servicio Snort para que se hagan efectivos los cambios.

6.1. Interfaz de usuario.

Lo primero que nos encontramos al entrar a la aplicación Web es la página de login para introducir las credenciales de usuario que nos permitirán acceder a la aplicación.



Una vez entramos nos encontramos con la página de Inicio, la cual nos muestra las alertas que tenemos activas en este momento.

Eduardo

| CONTROL WEB | | |
|---|------------------|---------------------|
|  | | |
| Inicio | Añadir Alerta | Borrar Alerta |
| Añadir Usuario | | Borrar Usuario |
| Alertas Activas | | |
| Origen | URL | Alerta |
| 192.168.30.10 | www.20minutos.es | Acceso a 20 Minutos |
| 192.168.30.20 | www.netflix.com | Acceso a Netflix |

En Añadir Alerta tenemos la interfaz para crear las nuevas alertas, únicamente debemos indicar si queremos controlar todos los equipos de la red o uno en concreto, la URL de la Web a monitorizar y el mensaje a mostrar en la alerta.

Inicio

Añadir Alerta

Borrar Alerta

Salir

Añadir Usuario

Borrar Usuario

Configure la alerta

Control para

Toda la Red

Equipo específico

URL

Sitio Web a controlar

Alerta

Mensaje a mostrar

Crear Alerta

En Borrar Alerta podemos seleccionar una o varias alertas para eliminar su monitorización.

| Inicio | | Añadir Alerta | Borrar Alerta | Salir ☺ |
|-----------|---------------|------------------|----------------|--------------------------|
| | | Añadir Usuario | Borrar Usuario | |
| ID Alerta | Origen | URL | Creador | Borrar |
| 1 | 192.168.30.10 | www.20minutos.es | Eduardo | <input type="checkbox"/> |
| 2 | 192.168.30.20 | www.netflix.com | Eduardo | <input type="checkbox"/> |
| 3 | Cualquier IP | www.marca.com | Eduardo | <input type="checkbox"/> |

Los administradores de nivel 2 (Superadministradores) pueden agregar y eliminar usuarios de la aplicación Web. Para añadir un usuario nuevo se debe de indicar su login (el que se le pedirá al entrar), su contraseña (en la base de datos se cifra en MD5), el nombre completo del usuario (por ejemplo: Eduardo de Lamo Téllez) y el nivel de privilegios que tendrá dicho usuario en la aplicación.

| Inicio | Añadir Alerta | Borrar Alerta | Salir ☺ |
|--------|----------------|----------------|---------|
| | Añadir Usuario | Borrar Usuario | |

Crear Usuario

Login

Contraseña

Nombre del usuario

Privilegios

Usuario Normal ▾

Crear Usuario

En Borrar usuario podemos eliminar a un usuario de la aplicación, hay que tener en cuenta que cuando lo hagamos todas las alertas que tenga activas ese usuario también se eliminarán, por lo tanto las URL's que este usuario tenía monitorizadas dejan de estarlo.

| Inicio | | Añadir Alerta | Borrar Alerta | Salir ☺ |
|---------|----------------|--------------------------|----------------|---------|
| | | Añadir Usuario | Borrar Usuario | |
| Usuario | Privilegios | Borrar | | |
| Javier | Usuario Normal | <input type="checkbox"/> | | |
| Laura | Usuario Normal | <input type="checkbox"/> | | |

6.2. Lógica de programación utilizada en la aplicación web.

La base del proyecto es la comunicación entre la aplicación Web y Snort, esta se hace mediante el fichero de la aplicación Web /alertas/anadir.php y el fichero de reglas Snort /etc/snort/rules/local.rules.

Cuando un usuario genera una nueva alerta, las opciones de la misma se introducen en el formulario de Añadir Regla, este formulario manda los datos recogidos al script /alertas/anadir.php. Este script lo primero que hace es conectar con la base de datos y después de las comprobaciones pertinentes inserta los datos recibidos en la tabla reglas de la base de datos.

```
anadir.php
66 // Insertar las opciones que hemos recogido del formulario de la nueva regla en la bd
67 $insertar = "INSERT INTO reglas VALUES ($sid, $rev, '$origen', '$URL', '$mensaje', '$usuario')";
68
69 if (mysqli_query($conexion, $insertar)) {
70     $insercion = true;
71 }
72 else {
73     echo "Error al insertar en la base de datos: " . $insertar . "<br>" . mysqli_error($conexion);
74 }
75
```

Una vez que la regla está en la base de datos se abre el fichero /etc/snort/rules/local.rules en modo escritura, se hace una consulta a la base de datos para sacar todas las reglas existentes en la misma y mediante un bucle se va escribiendo cada regla en una línea del fichero.

```
anadir.php
76 // Abrir el fichero local.rules de Snort para escribir las reglas de la bd en el mismo
77 $fichero = fopen("/etc/snort/rules/local.rules", "w") or die ("Hubo algún problema al abrir el fichero!");
78
79 // Consulta a la bd para ir sacando las reglas
80 $consulta = "SELECT sid, rev, origen, url, mensaje, usuario FROM reglas";
81
82 $resultado = mysqli_query($conexion, $consulta);
83
84 // Variable para introducir un salto de línea entre regla y regla
85 $salto_linea = "\n";
86
87 // Bucle para ir consultando en la bd e ir insertando en el fichero las reglas
88 if (mysqli_num_rows($resultado) > 0) { // si hay filas en la consulta
89     while($fila = mysqli_fetch_assoc($resultado)) {
90         // Sacamos los resultados a variables
91         $origen=$fila["origen"];
92         $mensaje=$fila["mensaje"];
93         $url=$fila["url"];
94         $sid=$fila["sid"];
95         $rev=$fila["rev"];
96
97         // Escribimos la regla en el fichero
98         fwrite($fichero, 'alert tcp ' . $origen . ' any -> any any (msg: "' . $mensaje . '"; content: "' . $url . '"; sid: ' . $sid . '
99             ; rev: ' . $rev . ');' . $salto_linea);
100     }
101 }
102 else {
103     echo "No hay reglas en la base de datos!";
104 }
105
106
```

Cuando termina el bucle de escritura en el fichero, se cierra la conexión con el mismo y con la base de datos, por último se reinicia el servicio de Snort para que se hagan efectivas las nuevas reglas.

```
107 // Cerramos el fichero y la conexión con la bd
108 fclose($fichero);
109 mysqli_close($conexion);
110
111 // Reiniciamos el servicio de Snort para hacer efectivas las reglas
112 $comando = "sudo service snort restart";
113
114 exec($comando);
115
```

Para poder reiniciar el servicio de Snort desde PHP el usuario www-data (usuario con el cual corre PHP) debe poder ejecutar comandos como superusuario del sistema, para ello lo he introducido en el archivo del sistema /etc/sudoers con el comando visudo.

```
#includedir /etc/sudoers.d
```

```
www-data          ALL=(root)          NOPASSWD:          ALL
```

Cuando borramos una alerta, esta se elimina de la base de datos y se vuelve a reescribir el fichero de reglas Snort de la misma manera, con las reglas restantes en la base de datos. Si no quedan alertas en la base de datos dejamos en blanco el fichero de reglas.

```
borrar_alerta.php x
91  ///// Si hemos seleccionado alertas para borrar /////
92  if( isset($_POST['borrar'])) {
93
94      // array con todos los sid de las reglas seleccionadas para borrar en el formulario
95      $borrar=$_POST['borrar'];
96
97      $i=0; // contador de alertas borradas
98
99      if ($borrar != "Borrar") { // si el array $borrar no está vacío seguimos con la ejecución normal
100
101      // Borramos las reglas en la bd
102      foreach ($borrar as $valor) {
103
104          $delete = "DELETE FROM reglas WHERE sid='$valor'";
105
106          if (mysqli_query($conexion, $delete)) {
107              $i++;
108          }
109          else {
110              echo "<p style='color:red'>Error borrando: " . mysqli_error($conexion) . "</p>";
111          }
112      }
113
```

```

114 // Ahora hay que actualizar el fichero local.rules con las reglas que quedan en la bd después de borrar las seleccionadas
115
116 // Abrir el fichero local.rules de snort
117 $fichero = fopen("/etc/snort/rules/local.rules","w") or die ("Hubo algún problema al abrir el fichero!");
118
119 // Consulta a la bd para ir sacando las reglas
120 $consulta = "SELECT sid, rev, origen, url, mensaje, usuario FROM reglas";
121
122 $resultado = mysqli_query($conexion, $consulta);
123
124 // Variable para introducir un salto de línea entre regla y regla
125 $salto_linea = "\n";
126
127 // Bucle para ir consultando en la bd e ir insertando en el fichero las reglas
128 if (mysqli_num_rows($resultado) > 0) { // si hay filas en la consulta
129
130     while($fila = mysqli_fetch_assoc($resultado)) {
131
132         // Sacamos los resultados a variables
133         $origen=$fila["origen"];
134         $mensaje=$fila["mensaje"];
135         $url=$fila["url"];
136         $sid=$fila["sid"];
137         $rev=$fila["rev"];
138
139         // Escribimos la regla en el fichero
140         fwrite($fichero, 'alert tcp ' . $origen . ' any -> any any (msg: "' . $mensaje . '"; content: "' . $url . '"; sid: ' . $sid . '
            ; rev: ' . $rev . ');' . $salto_linea);
141     }
142 }
143 else {
144     // Si no quedan reglas en la bd, vaciamos el fichero
145     fwrite($fichero, '');
146 }
147
148 // Cerramos el fichero
149 fclose($fichero);
150
151 // Reiniciamos el servicio de Snort para hacer efectivas las reglas
152 $comando = "sudo service snort restart";
153
154 exec($comando);

```

Cuando eliminamos a un usuario, sus reglas se eliminan de la base de datos en cascada también, por lo que hay que volver a escribir el fichero de reglas con las restantes en la base de datos.

```

92 // Si hemos marcado un usuario para borrar y recibimos los datos del formulario
93
94 if( isset($_POST['borrar_usu']) ) { // Comienzo del isset
95
96     $usuario = $_POST['borrar_usu'];
97
98     // Borramos al usuario de la bd
99
100     $borrar = "DELETE FROM usuarios where login='$usuario'";
101
102     if (mysqli_query($conexion, $borrar)) {
103
104         echo "<p style='color: green'>Usuario borrado</p>";
105     }
106
107     else {
108
109         echo "<p style='color:red'>Error borrando: " . mysqli_error($conexion) . "</p>";
110     }
111 }
112

```

```

113 // Cuando borramos un usuario debemos volver a escribir las reglas en el fichero local.rules
114 // ya que estas se eliminan en cascada al eliminar al usuario en la base de datos pero no en el fichero
115
116 //// Abrir el fichero local.rules de snort para escribir las reglas de la bd en él
117 $fichero = `fopen("/etc/snort/rules/local.rules","w")` or die ("Hubo algún problema al abrir el fichero!");
118
119 // Consulta a la bd para ir sacando las reglas
120 $consulta = "SELECT sid, rev, origen, url, mensaje, usuario FROM reglas";
121
122 $resultado = mysqli_query($conexion, $consulta);
123
124 // Variable para introducir un salto de línea entre regla y regla
125 $salto_linea = "\n";
126
127 // Bucle para ir consultando en la bd e ir insertando en el fichero las reglas
128 if (mysqli_num_rows($resultado) > 0) { // si hay filas en la consulta
129
130     while($fila = mysqli_fetch_assoc($resultado)) {
131
132         // Sacamos los resultados a variables
133         $origen=$fila["origen"];
134         $mensaje=$fila["mensaje"];
135         $url=$fila["url"];
136         $sid=$fila["sid"];
137         $rev=$fila["rev"];
138
139         // Escribimos la regla en el fichero
140         fwrite($fichero, 'alert tcp ' . $origen . ' any -> any any (msg: "' . $mensaje . '"; content: "' . $url . '"; sid: ' . $sid
141             . ' ; rev: ' . $rev . ');' . $salto_linea);
142     }
143 } else {
144     // Si no quedan reglas en la bd, vaciamos el fichero
145     fwrite($fichero, '');
146     echo "<p style='color: red'>No quedan alertas activas</p>";
147 }
148
149 // Cerramos el fichero y la conexión con la bd
150 fclose($fichero);
151
152 // Reiniciamos el servicio de Snort para hacer efectivas las reglas
153 $comando = "sudo service snort restart";
154
155 exec($comando);

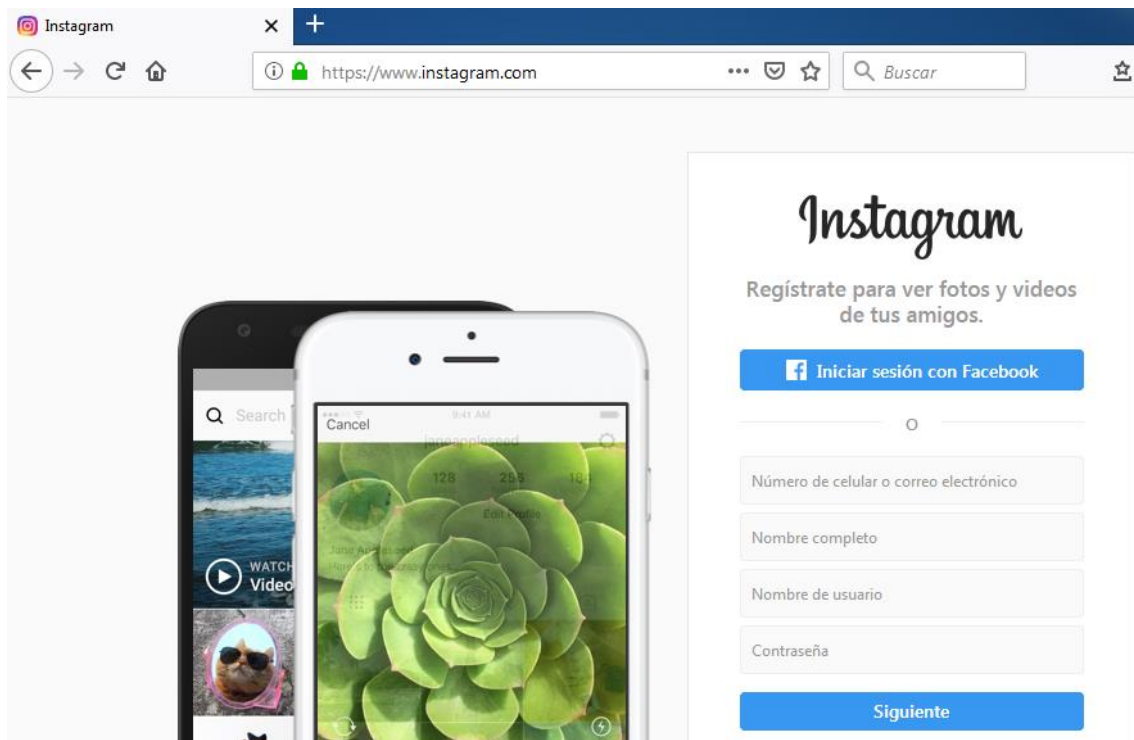
```

6.3. Ejemplos de uso de la aplicación Web.

Ejemplo 1 -> Creamos una alerta para monitorizar los accesos a la URL www.instagram.com desde el equipo con IP 192.168.30.10.

| Inicio | Añadir Alerta | Borrar Alerta | Salir ☺ |
|--|---------------|----------------|---------|
| Añadir Usuario | | Borrar Usuario | |
| Configure la alerta | | | |
| Control para | | | |
| <div>Toda la Red <input type="radio"/></div> <div>Equipo específico <input checked="" type="radio"/></div> <div>IP del equipo</div> <div>192.168.30.10</div> | | | |
| URL | | | |
| <div>Sitio Web a controlar</div> <div>www.instagram.com</div> | | | |
| Alerta | | | |
| <div>Mensaje a mostrar</div> <div>Acceso a Instagram</div> | | | |
| <div>Crear Alerta</div> | | | |

El usuario accede a dicha página desde el PC indicado en la regla.



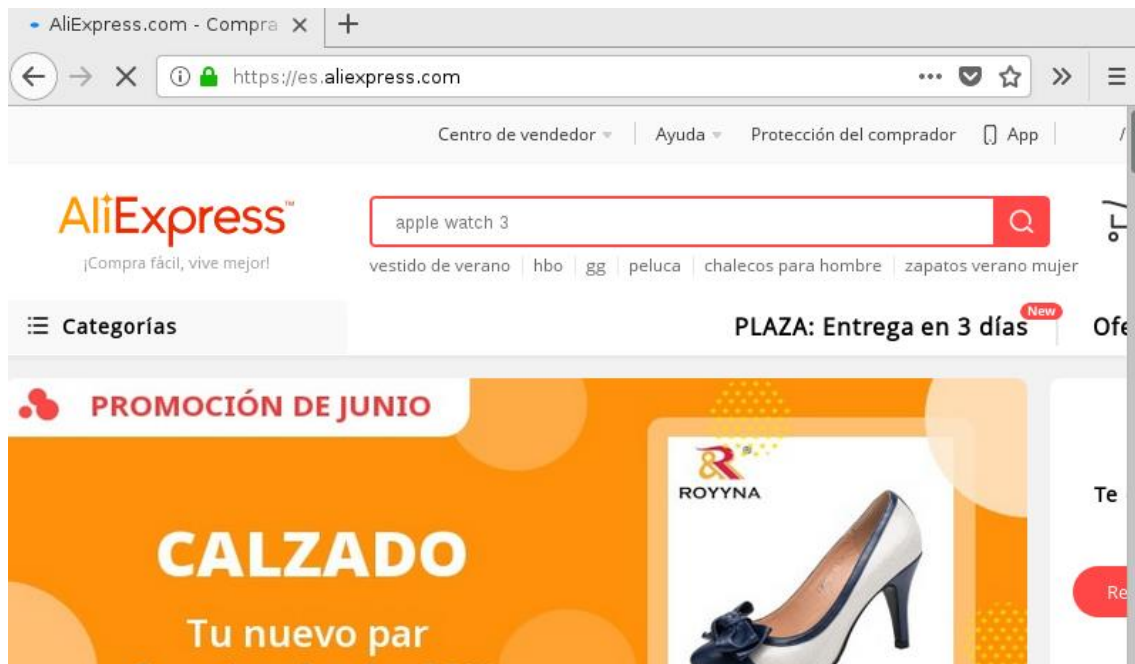
Vemos como el acceso es detectado por Snort y nos lo muestra mediante la alerta correspondiente en consola.

```
Commencing packet processing (pid=5193)
06/16-12:13:58.814190  [**] [1:4:4] Acceso a Instagram [**] [Priority:
0] {TCP} 192.168.30.10:49198 -> 31.13.83.174:80
06/16-12:13:58.956745  [**] [1:4:4] Acceso a Instagram [**] [Priority:
0] {TCP} 192.168.30.10:49199 -> 31.13.83.174:443
```

Ejemplo 2 -> Creamos una alerta para monitorizar los accesos a la URL www.aliexpress.com desde el equipo con IP 192.168.30.20.

| Inicio | Añadir Alerta | Borrar Alerta | Salir ☺ |
|---|--|----------------|---------|
| Añadir Usuario | | Borrar Usuario | |
| Configure la alerta | | | |
| Control para | <input type="radio"/> Toda la Red <input checked="" type="radio"/> Equipo específico | | |
| | IP del equipo 192.168.30.20 | | |
| URL | Sitio Web a controlar www.aliexpress.com | | |
| Alerta | Mensaje a mostrar Acceso a Aliexpress | | |
| <input type="button" value="Crear Alerta"/> | | | |

El usuario accede a dicha página desde el PC indicado en la regla.



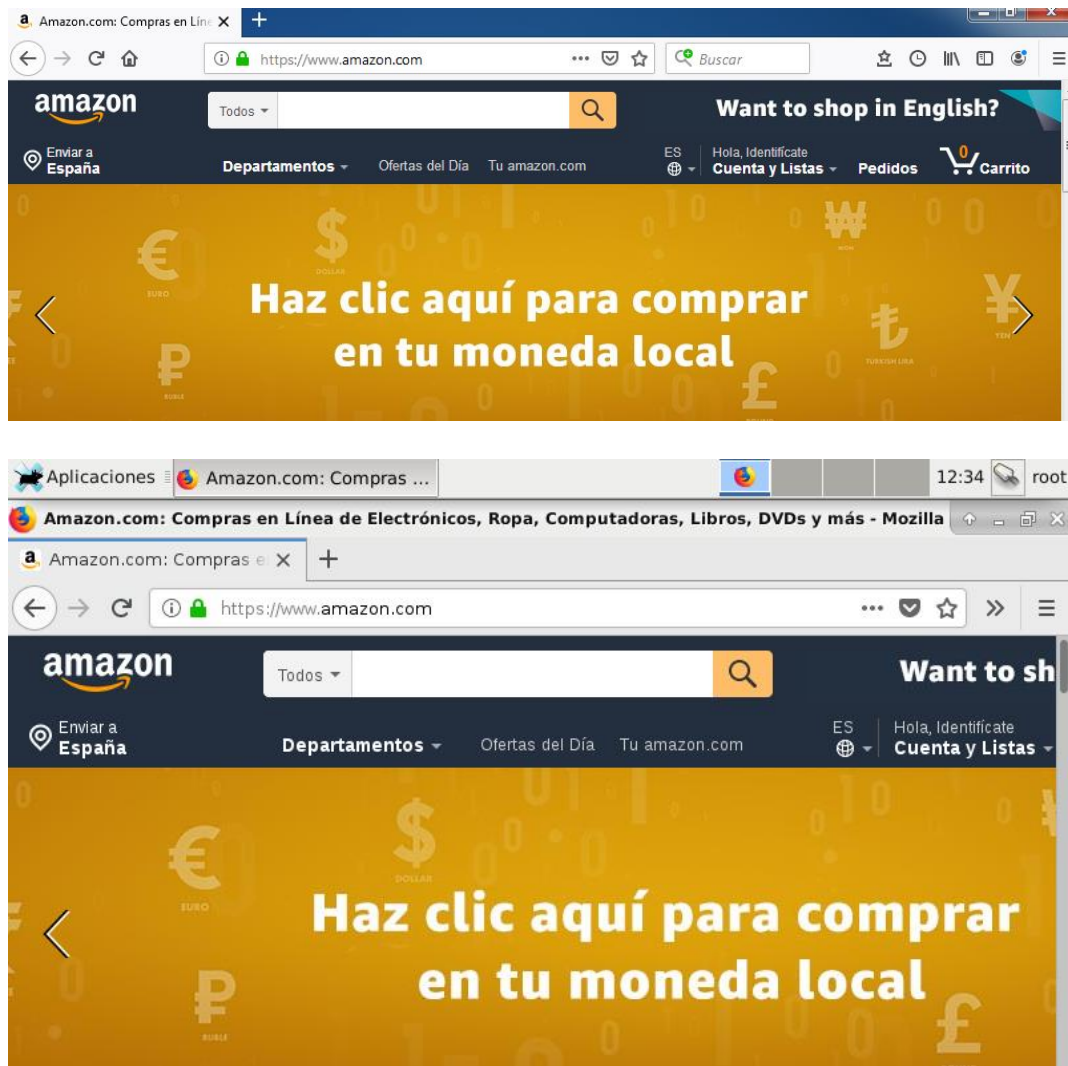
Vemos como el acceso es detectado por Snort y nos lo muestra mediante la alerta correspondiente en consola.

```
Commencing packet processing (pid=5273)
06/16-12:20:27.583347  [**] [1:5:5] Acceso a Aliexpress [**] [Priority:
0] {TCP} 192.168.30.20:42906 -> 23.39.74.22:80
```

Ejemplo 3 -> Creamos una alerta para monitorizar los accesos a la URL www.amazon.com desde cualquier equipo de la red interna.

| Inicio | Añadir Alerta | Borrar Alerta | Salir ☺ |
|--|---------------|---|---------|
| Añadir Usuario | | Borrar Usuario | |
| Configure la alerta | | | |
| Control para | | | |
| Toda la Red <input checked="" type="radio"/> | | Equipo específico <input type="radio"/> | |
| URL | | | |
| Sitio Web a controlar | | | |
| <input type="text" value="www.amazon.com"/> | | | |
| Alerta | | | |
| Mensaje a mostrar | | | |
| <input type="text" value="Acceso a Amazon"/> | | | |
| <input type="button" value="Crear Alerta"/> | | | |

Los usuarios acceden a dicha página desde sus respectivos PC's.



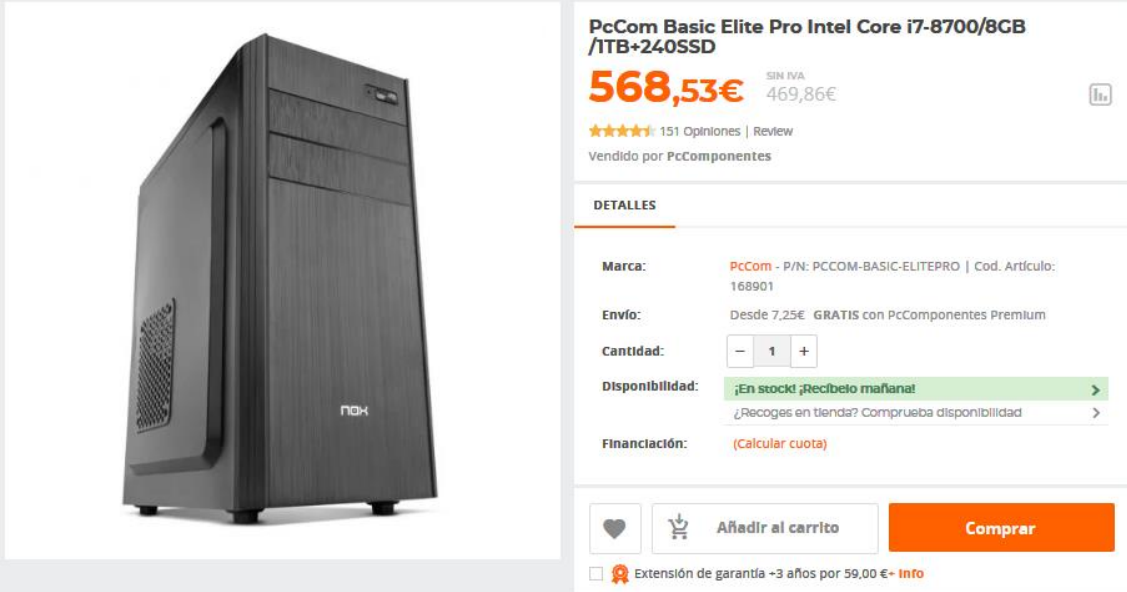
Vemos como los accesos son detectados por Snort y nos los muestra mediante las alertas correspondientes en consola.

```
Commencing packet processing (pid=5393)
06/16-12:29:33.849550  /**] [1:6:6] Acceso a Amazon /**] [Priority: 0]
{TCP} 192.168.30.10:49321 -> 92.122.233.46:443
06/16-12:29:33.873363  /**] [1:6:6] Acceso a Amazon /**] [Priority: 0]
{TCP} 192.168.30.10:49322 -> 92.122.233.46:443
06/16-12:29:43.720507  /**] [1:6:6] Acceso a Amazon /**] [Priority: 0]
{TCP} 192.168.30.10:49338 -> 92.122.233.46:443
06/16-12:29:52.839030  /**] [1:6:6] Acceso a Amazon /**] [Priority: 0]
{TCP} 192.168.30.20:59284 -> 104.83.80.212:443
06/16-12:29:52.848458  /**] [1:6:6] Acceso a Amazon /**] [Priority: 0]
{TCP} 192.168.30.20:59294 -> 104.83.80.212:443
06/16-12:29:52.854227  /**] [1:6:6] Acceso a Amazon /**] [Priority: 0]
{TCP} 192.168.30.20:59292 -> 104.83.80.212:443
06/16-12:29:52.856785  /**] [1:6:6] Acceso a Amazon /**] [Priority: 0]
{TCP} 192.168.30.20:59290 -> 104.83.80.212:443
06/16-12:29:52.871547  /**] [1:6:6] Acceso a Amazon /**] [Priority: 0]
{TCP} 192.168.30.20:59288 -> 104.83.80.212:443
06/16-12:29:52.877389  /**] [1:6:6] Acceso a Amazon /**] [Priority: 0]
{TCP} 192.168.30.20:59286 -> 104.83.80.212:443
```


7. Coste económico del Proyecto.

El proyecto se desarrolla en una máquina con 2 tarjetas de red sobre la que instalamos todo el software necesario, esta máquina debe tener suficiente potencia como para servir de puerta de enlace a los equipos de la red interna y a la vez capturar todo el tráfico que pasa por ella.

Para ello me decanto por el siguiente equipo de sobremesa:



PcCom Basic Elite Pro Intel Core i7-8700/8GB /1TB+240SSD

568,53€ SIN IVA 469,86€

★★★★★ 151 Opiniones | Review

Vendido por PcComponentes

DETALLES

Marca: PcCom - P/N: PCCOM-BASIC-ELITEPRO | Cod. Artículo: 168901

Envío: Desde 7,25€ GRATIS con PcComponentes Premium

Cantidad: - 1 +

Disponibilidad: ¡En stock! ¡Recíbelo mañana! >
¿Recoges en tienda? Comprueba disponibilidad >

Financiación: (Calcular cuota)

♥️ 🛒 Añadir al carrito **Comprar**

☐ 🛡️ Extensión de garantía ~3 años por 59,00 €+ [Info](#)

Estas son sus características principales:

Caja: Nox LITE010 USB 3.0.

Procesador: Intel Core i7-8700.

Placa base: MSI B360M.

Discos duros:

- SSD 240GB SATA3.
- 1TB HDD SATA3.

Memoria RAM: DDR4 2400 PC4-19200 8GB.

Tarjeta gráfica: Gráficos HD Intel® 630.

Precio: 568,53 €.

Utilizaré 2 tarjetas de red dedicadas idénticas sobre el equipo, estas tendrán soporte para grandes velocidades de conexión a Internet.



Características:

1 Puerto RJ-45 para Ethernet x 100Mbps/1Gbps/2.5Gbps/5Gbps/10Gbps Mbps.
Autonegociación: Speed.

Estándares:

- IEEE 802.3an 10GBASE-T.
- IEEE 802.3u 100Base-TX Fast Ethernet.
- IEEE 802.3ab 1000BASE-T Gigabit Ethernet.
- IEEE 802.3az Energy Efficient Ethernet.
- IEEE 802.1p Priority Queuing.

Rendimiento: Jumbo Frame hasta 9 KB.

Precio: 86,45 €.

* Todo el software que se instala sobre la máquina es software libre como se ha mencionado anteriormente, por lo que en esta parte el costo es de 0 €.

Coste total del proyecto:

1 PC Basic Elite Pro -> 568,53 €.

2 Tarjetas de red Asus XG-C100C -> 172,90 €.

Total = 741,43 €.

8. Conclusiones.

Aunque la herramienta desarrollada pueda parecer que tiene similitudes con un Proxy HTTP, en realidad no funciona de la misma manera y su propósito es diferente. Un Proxy actúa de intermediario entre un cliente que solicita una petición de una página HTTP y el servidor Web, mientras que en mi herramienta no hay intermediarios en las peticiones HTTP, Snort solamente se dedica analizar los paquetes que pasan por la red y si alguno de ellos coincide con alguna de las reglas configuradas genera una alerta. Debido a que actúa como intermediario, el Proxy también ralentiza un poco los accesos Web a los clientes, ya que la petición no es directa del cliente al servidor; primero esta petición va del cliente al Proxy, el Proxy hace la petición a servidor Web, el servidor Web responde al Proxy y este manda la respuesta al cliente, sin embargo con Snort la petición si es directa del cliente al servidor ya que este no interviene, con lo que no se genera el pequeño retraso que se produce a través de un Proxy HTTP. Por otra parte un Proxy HTTP registra todos y cada uno de los accesos Web que pasan por él, mientras que la herramienta desarrollada solo registra los accesos a los sitios Web que los administradores quieren tener monitorizados, esto es mucho más eficiente en el caso que nos ocupa ya que solo quedan registrados los accesos interesantes y es mucho más fácil para los administradores mirar los registros de log y sacar conclusiones.

Módulos del ciclo de ASIR en los que me he basado para crear la herramienta:

- Implantación de sistemas operativos: Instalación y configuración del sistema operativo Ubuntu 16.04 LTS.
- Planificación y administración de redes: Diseño del esquema de la red, configuración de las tarjetas de red de los equipos y del enrutamiento del escenario.
- Fundamentos del hardware: Montaje y puesta en funcionamiento de los equipos del proyecto (Máquinas Virtuales).
- Gestión de bases de datos: Diseño de la base de datos necesaria para la aplicación Web.
- Lenguajes de marcas y sistemas de gestión de información: Programación del interfaz de usuario de la aplicación Web mediante HTML, CSS, y JavaScript.
- Administración de sistemas operativos: Programación de los Scripts necesarios para el funcionamiento del proyecto.
- Servicios de red e Internet: Puesta en marcha y configuración de los servicios necesarios para el desarrollo del proyecto, como por ejemplo Apache.
- Implantación de aplicaciones web: Lógica de programación de la aplicación Web del lado del servidor mediante PHP.
- Administración de sistemas gestores de bases de datos: Implantación y sincronización de la base de datos con la aplicación Web, sacar la información necesaria de la misma mediante PHP.
- Seguridad y alta disponibilidad: Snort es una herramienta de seguridad, por lo tanto todo lo referido con su puesta en marcha, su configuración y la generación de reglas.

9. Mejoras futuras del proyecto.

- Para iniciar la captura del tráfico de red con Snort es necesario abrir un terminal y escribir el comando pertinente, he logrado poner el comando en un Script y crear un lanzador en el Dash de Ubuntu que enlaza a dicho Script para abrirlo de forma más sencilla. Para mejorar esto he intentado poner un botón en la aplicación Web desde el cual poder iniciar Snort desde la misma, pero tras muchas horas de investigación no me ha sido posible ya que no hay posibilidad de abrir un terminal de Ubuntu mediante PHP. La conclusión que he sacado es que esto se debe a motivos de seguridad del sistema. Aunque desde PHP se puedan ejecutar comandos del sistema, no se pueden abrir aplicaciones externas ejecutables. Probablemente con el framework o API adecuada sea posible realizarlo.

- Los logs que generan las alertas se guardan en una carpeta específica en ficheros de texto plano. Ir analizando los ficheros uno por uno puede llegar a ser tedioso y pesado para los administradores, por lo que intentado que los logs se guarden en la base de datos para luego sacarlos de manera más visual y ordenada en la aplicación Web pero sin éxito. Hasta hace unos años Snort permitía el volcado de los logs en la base de datos de forma nativa, pero por alguna razón incomprensible lo quitaron. Actualmente investigando he visto que se puede utilizar una API llamada Banyard2 para lograrlo, pero aunque la he instalado y configurado varias veces no he conseguido que las alertas de Snort se llegaran a volcar en la base de datos. Puede que dicha API esté desactualizada o que ya no funcione conjuntamente con Snort. Habría que seguir investigando la manera de conseguir volcar las alertas en la base de datos.

- En el desarrollo de la herramienta me he centrado solo en controlar los accesos a ciertas URL's por parte de los usuarios, pero las reglas de Snort son altamente configurables y se puede controlar cualquier paquete que pasa por la red. La aplicación Web se podría ampliar para dar más opciones al usuario para el control dichos paquetes, por ejemplo se pueden controlar más protocolos de red como el ICMP y el UDP, podemos controlar por puertos, podemos controlar por dirección de la operación, por payload del paquete, etc.

- En consecuencia con el punto anterior, la herramienta actualmente solamente controla el tráfico que sale desde los equipos de la red interna hacia el exterior, pero se podrían ampliar las opciones para controlar el tráfico que viene desde el exterior hacia la red interna, con lo cual mejoraríamos la seguridad de la organización.

- Actualmente la función de Snort en mi proyecto es la de capturar el tráfico de la red y generar las alertas pertinentes, pero también se puede configurar para actuar en modo IPS, esto significa que se puede sincronizar con IPtables para que este firewall bloquee los paquetes que queramos, con lo cual podríamos evitar que los usuarios navegaran por ciertos sitios Web aumentando así mucho más la seguridad de la organización.