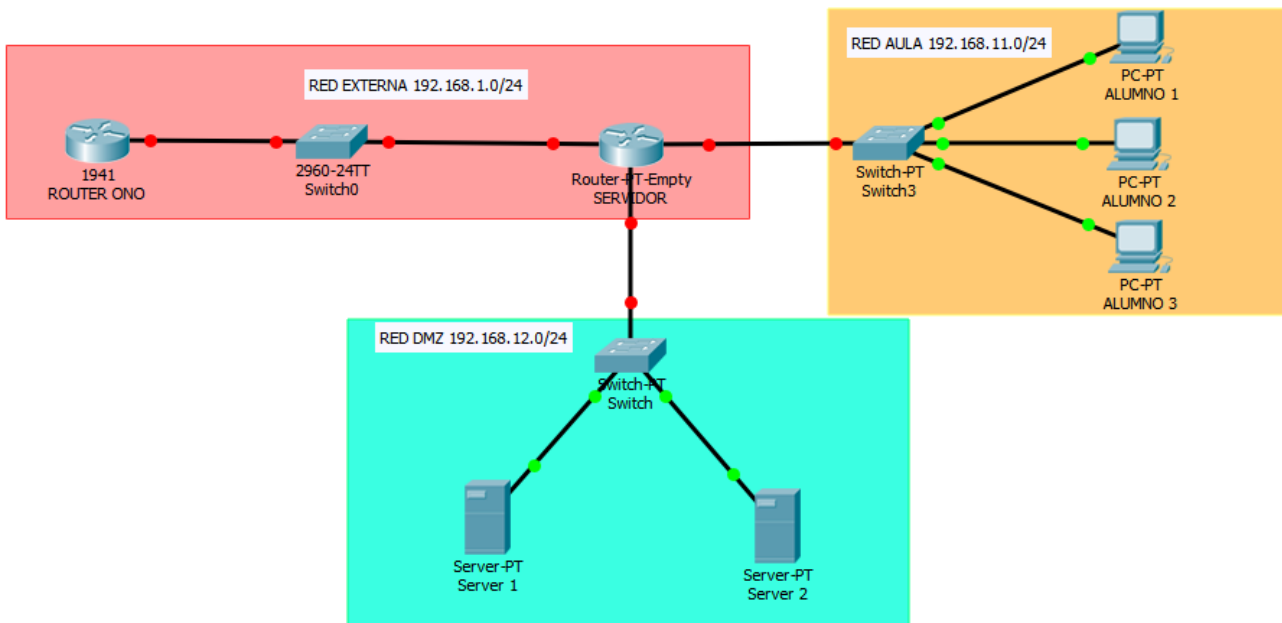


## 1. Servidor

### 1.1. Esquema de red

Este servidor actuará como enrutador para las diferentes redes mediante Iptables. Existirán 3 redes distintas; la red externa, la red del aula y la red del DMZ.



### 1.2. Sistema operativo

El sistema empleado es Debian versión 8.9. Los datos de los pasos de instalación del sistema operativo son:

- Usuario → usuario
- Contraseña → Tico\*65
- Contraseña root → Tico\*65
- IP servidor → 192.168.1.18

## 2. Respaldo mediante Ansible

### 2.1. Clave publica y SSH

Lo primero será instalar en el servidor el paquete `openssh-server` y modificar el fichero de configuración ubicado en `/etc/ssh`. Establecer *yes* el parámetro de configuración *PermitRootLogin* y reiniciar el servicio

Una vez configurado SSH en el servidor, es necesario autorizar al nodo *orquestador* por SSH enviando desde el nodo *orquestador* la clave pública al servidor.

Ahora, desde el nodo orquestador, es necesario clonar [el repositorio](#). Después entrar en *ansible > respaldo*. Dentro de esa carpeta se encuentra el *playbook* llamado *respaldo*, el cual se necesita ejecutar.

Es importante que si se va a desplegar Ansible en otra dirección IP que no sea la dirección 192.168.1.18 se modifique en el fichero `hosts`.

Para comenzar se necesita enviar la clave pública del nodo *orquestador* al servidor. Desde el nodo *orquestador* se crea la clave y se envía por SSH.

*ssh-keygen*

```
root@pc:/home/jesus/Proyecto_Jesus_Zamora_Jimenez/ansible/respaldo# ssh-key
ssh-keygen  ssh-keyscan
root@pc:/home/jesus/Proyecto_Jesus_Zamora_Jimenez/ansible/respaldo# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
51:0d:89:e2:01:fd:1a:03:d2:b4:b8:17:02:44:57:46 root@pc
The key's randomart image is:
+--[ RSA 2048 ]-----+
|+o +=E   .o+       |
|.O.+..+ ... .      |
| o.OO +.           |
| o .+ ..           |
|. . +S             |
|. .                |
|                   |
|                   |
+-----+

```

Tras haber creado la clave pública se envía por SSH a servidor. Requerirá introducir la clave el usuario root del servidor. Previamente se ha modificado el fichero de configuración de SSH del servidor para que en este paso permita la autenticación de root.

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.1.18
```

```
root@pc:/home/jesus/Proyecto_Jesus_Zamora_Jimenez/ansible/respaldo# ssh-copy-id
-i ~/.ssh/id_rsa.pub root@192.168.1.18
The authenticity of host '192.168.1.18 (192.168.1.18)' can't be established.
ECDSA key fingerprint is 9f:4d:90:aa:ac:64:eb:c5:26:68:d4:61:26:52:ad:63.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
root@192.168.1.18's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@192.168.1.18'"
and check to make sure that only the key(s) you wanted were added.
```

Una vez copiada se verifica que Ansible detecta la IP para poder desplegar la maqueta.

```
root@orquestador: /home/jesus
root@orquestador: /home/jesus x jesus@orquestador: ~ x
root@orquestador: /home/jesus# ansible all -m ping -u root
192.168.1.18 | success >> {
  "changed": false,
  "ping": "pong"
}
root@orquestador: /home/jesus#
```

Tras esto ya se puede desplegar Ansible sobre el servidor. Para ello se utilizará el comando *ansible-playbook*.

```
root@pc:/home/jesus/Proyecto_Jesus_Zamora_Jimenez/ansible/respaldo# ansible-play
book respaldo.yml

PLAY [respaldo] *****

GATHERING FACTS *****
ok: [192.168.1.18]

TASK: [Actualizar paquetes] *****
ok: [192.168.1.18]

TASK: [Instalar Squid] *****
changed: [192.168.1.18]

TASK: [Configurar squid.conf] *****
```

### 3. Proxy no transparente

El proxy no transparente es Squid en la versión 3.2.8, ubicado entre la red interna y el router.

La configuración de Squid se encuentra en `/etc/squid3/squid.conf`. Las reglas que se han dejado escritas son las de denegar una serie de dominios en nowebbs y easylist, la lista negra.

```
24 #listas de control de acceso
25 #acl passwd proxy_auth REQUIRED
26 acl acceso src all
27 #acl nopermitidas url_regex "/etc/squid3/nopermitidas"
28 acl nowebbs dstdomain "/etc/squid3/nowebbs"
29 acl easylist dstdomain "/etc/squid3/easylist"
30 #acl extensiones urlpath_regex "/etc/squid3/extensiones"
31 #acl permitidas url_regex "/etc/squid3/permitidas"
32 #acl limit maxconn 20
33
34 #control de acceso
35 #http_access allow permitidas
36 #http_access deny maq1
37 #http_access deny extensiones
38 #http_access deny nopermitidas
39 http_access deny nowebbs
40 http_access deny easylist
41 #http_access deny !passwd
42
43 http_access allow acceso
```

De modo que los usuarios que quieran navegar necesitarán configurar su navegador estableciendo un proxy con la IP del servidor y puerto 3128.

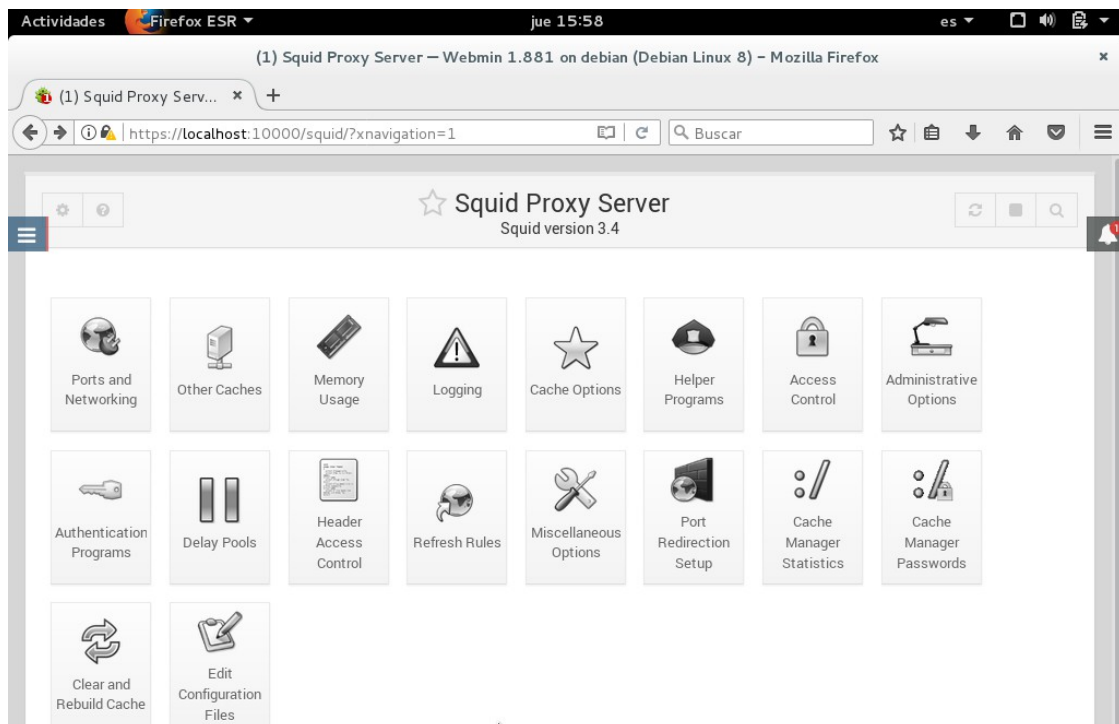


### 3.2. Webmin

Para este proyecto se incluirá una interfaz amigable web que permitirá configurar y administrar reglas para Squid, entre otras muchas funciones.

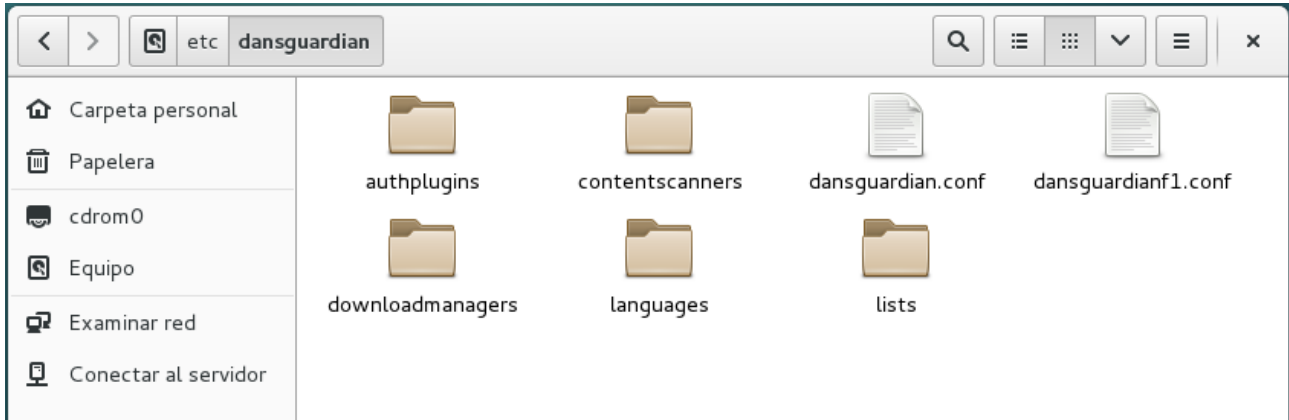
Una vez instalado y reiniciado el servicio en el servidor, Webmin es visible desde un navegador web, dentro de la red 192.168.1.0/24, accediendo a la dirección <https://192.168.1.18:10000>. Las credenciales de acceso es el usuario *root* y su contraseña (Tico\*65).

Una vez dentro de Webmin, en el menú lateral derecho dentro de *Servers* se encontrará la opción *Squid Proxy Server* que permitirá acceder a la configuración de Squid.



## 4. Dansguardian

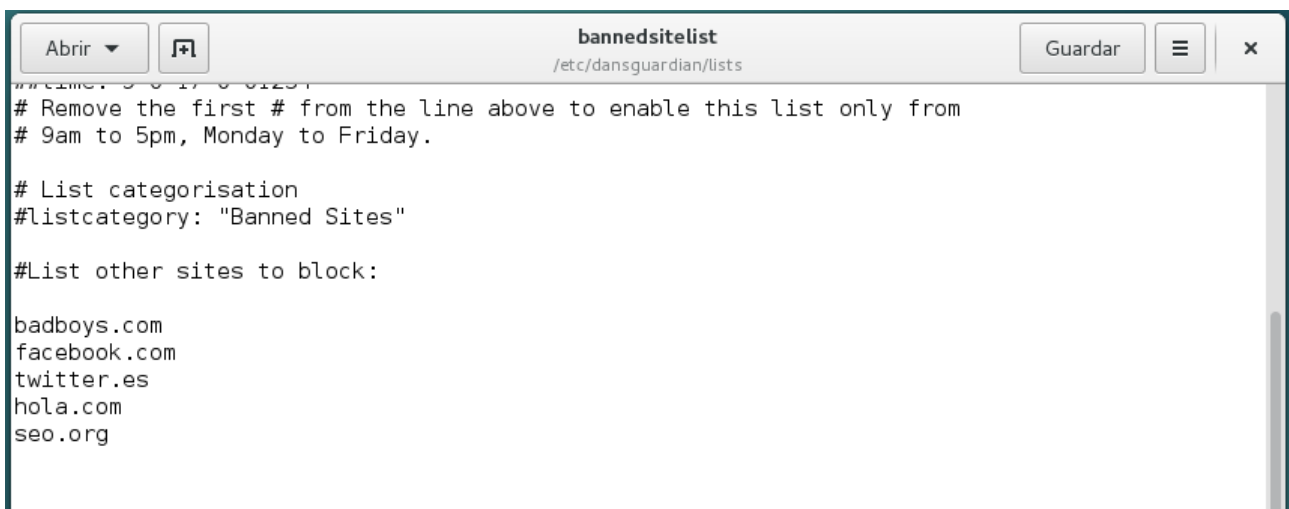
Dansguardian tiene su carpeta de configuración ubicada en */etc/dansguardian*, cuenta con un archivo de configuración llamado *dansguardian.conf* y varias carpetas que contienen diferentes configuraciones.



En el fichero de configuración se define el puerto donde escuchará Dansguardian, el puerto donde escuchará Squid y la dirección IP donde residirá Squid.

- filterport = 8080 → Puerto de Dansguardian.
- proxyip = 127.0.0.1 → Dirección IP de la maquina donde reside Squid.
- proxyport = 3128 → Puerto donde escucha Squid.

Este proxy tendrá configurado por defecto en el archivo *bannedsitelist* algunos sitios web.



## 5. Reportes de usuarios

Para obtener unos informes detallados se empleará Sarg, una herramienta de código abierto que permite analizar los logs de Squid y genera informes web en formato HTML con información sobre usuarios, direcciones IP, sitios web visitados, uso de ancho de banda total, tiempo transcurrido, descargas e sitios web denegados.

En su fichero de configuración ubicado en `/etc/sarg/`, se establecerá `access_log` para indicar donde Sarg debe tomar el log de Squid. Además será necesario indicar donde mostrará la salida de los reportes web, en línea `output_dir` indicando `/var/www/html/squid-reports`.



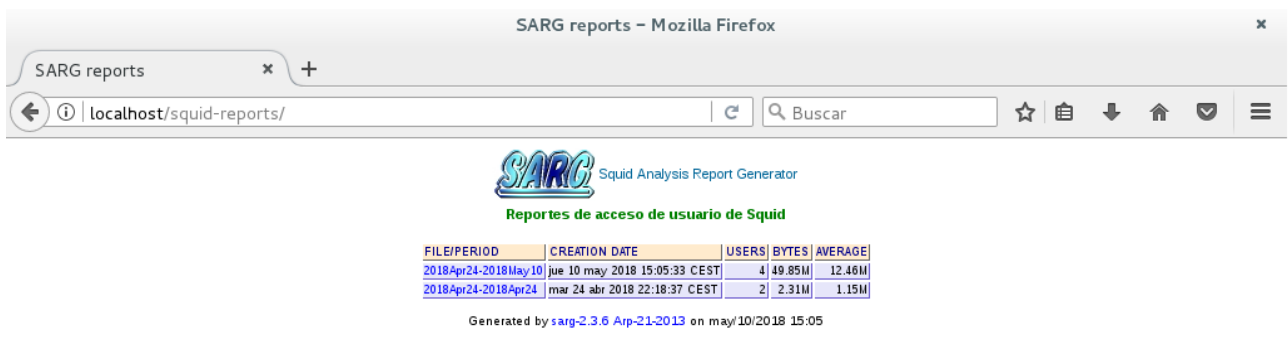
```
# sarg.conf
#
# TAG: access_log file
#      Where is the access.log file
#      sarg -l file
#
access_log /var/log/squid/access.log
```



```
# TAG: output_dir
#      The reports will be saved in that directory
#      sarg -o dir
#
output_dir /var/www/html/squid-reports
#output_dir /var/lib/sarg
```

Con Sarg -x ya comenzará a mostrar en la ruta web que se ha indicado los informes de Squid cuando al menos un cliente haya navegado hacia un sitio web.

Será accesible desde un navegador web de la red 192.168.1.0/24 accediendo a `http://192.168.1.18/squid-reports`.



SARG reports - Mozilla Firefox

SARG reports

localhost/squid-reports/

**SARG** Squid Analysis Report Generator

Reportes de acceso de usuario de Squid

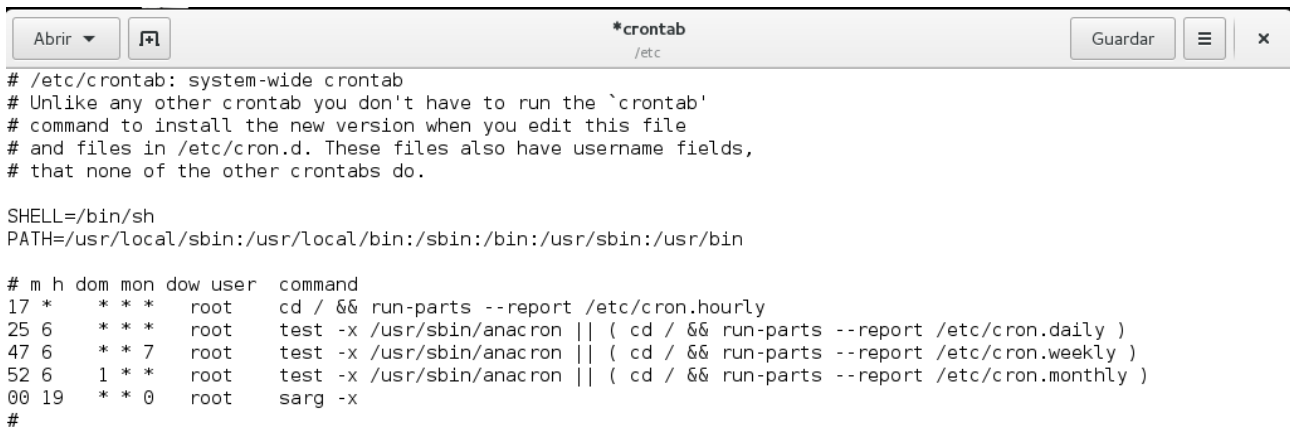
FILE/PERIOD	CREATION DATE	USERS	BYTES	AVERAGE
2018Apr24-2018May10	jue 10 may 2018 15:05:33 CEST	4	49.85M	12.46M
2018Apr24-2018Apr24	mar 24 abr 2018 22:18:37 CEST	2	2.31M	1.15M

Generated by sarg-2.3.6 Apr-21-2013 on may/10/2018 15:05

## 5.1. Sarg en Cron

Para que se tengan informes todos los días es necesario lanzar Sarg para que genere informes, de modo que para que todas las mañanas al empezar las clases se puedan ver los informes del día anterior es necesario programar la tarea.

Para ello se ha utilizado Cron, para ello hay que editar el fichero *crontab*.



```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
00 19 * * 0 root    sarg -x
#
```

Tal y como se aprecia en la imagen, todos los días a las 19h de la tarde se lanzará la tarea *Sarg -x* que generará informes accesibles mediante un navegador web visitando el sitio <http://localhost/squid-reports>.

## 6. Iptables

La ubicación de las reglas Iptables del servidor se encuentran en */etc/init.d/iptables.sh*

Las reglas del cortafuegos Iptables no se mantienen grabados en el sistema si el servidor es apagado y vuelto a encender. Para que se mantengan, es necesario incluir en el arranque de los servicios del sistema un script con las reglas que anteriormente se han descrito de Iptables.

Para ello se han aplicado los siguientes comandos:

- *update-rc.d iptables.sh defaults 99*



