Instalación y configuración

David Silva Sanmartín

Trabajo presentado para
Grado en Administración de Sistemas Informáticos en Red

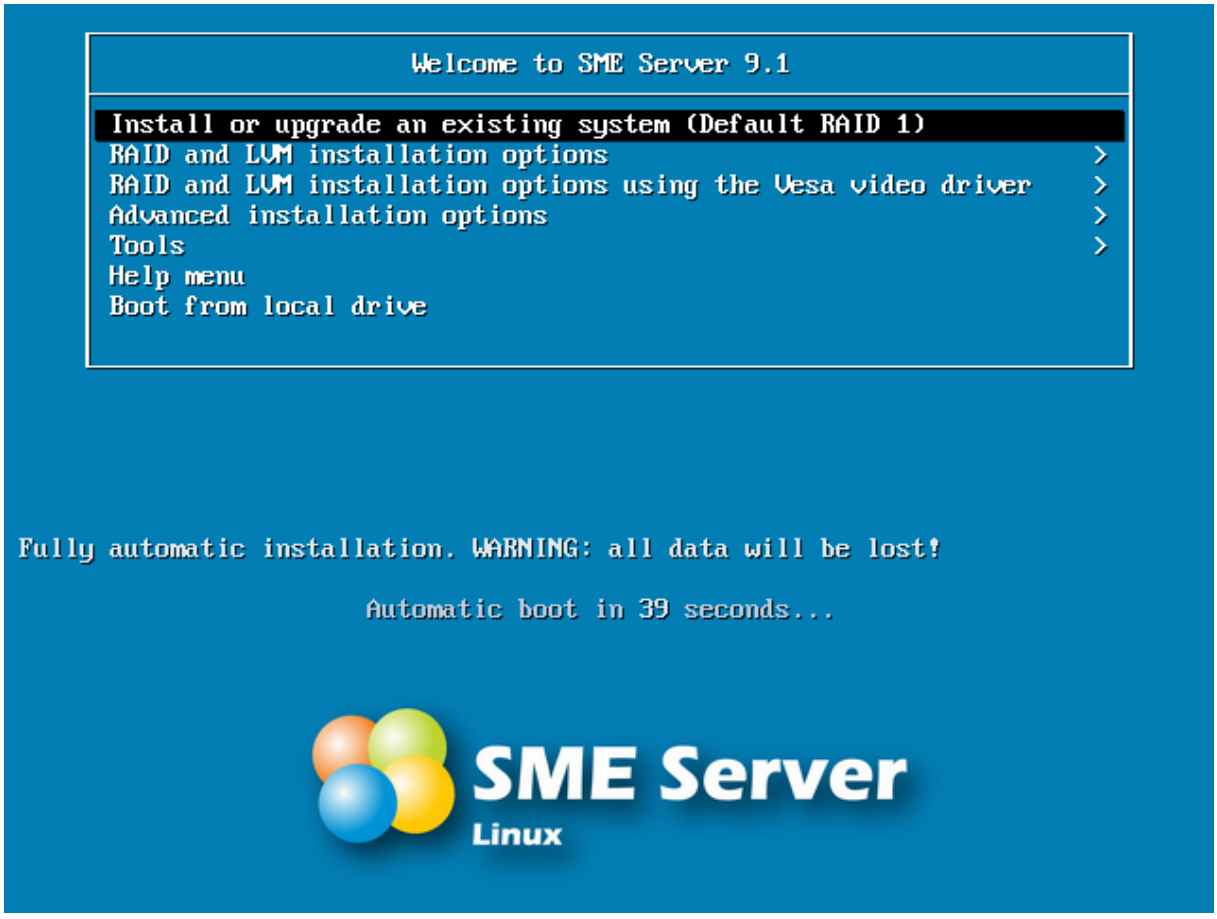
Departamento de Informática
IES Pablo Serrano
Zaragoza
06 de junio de 2016

Índice general

1 | Instalación

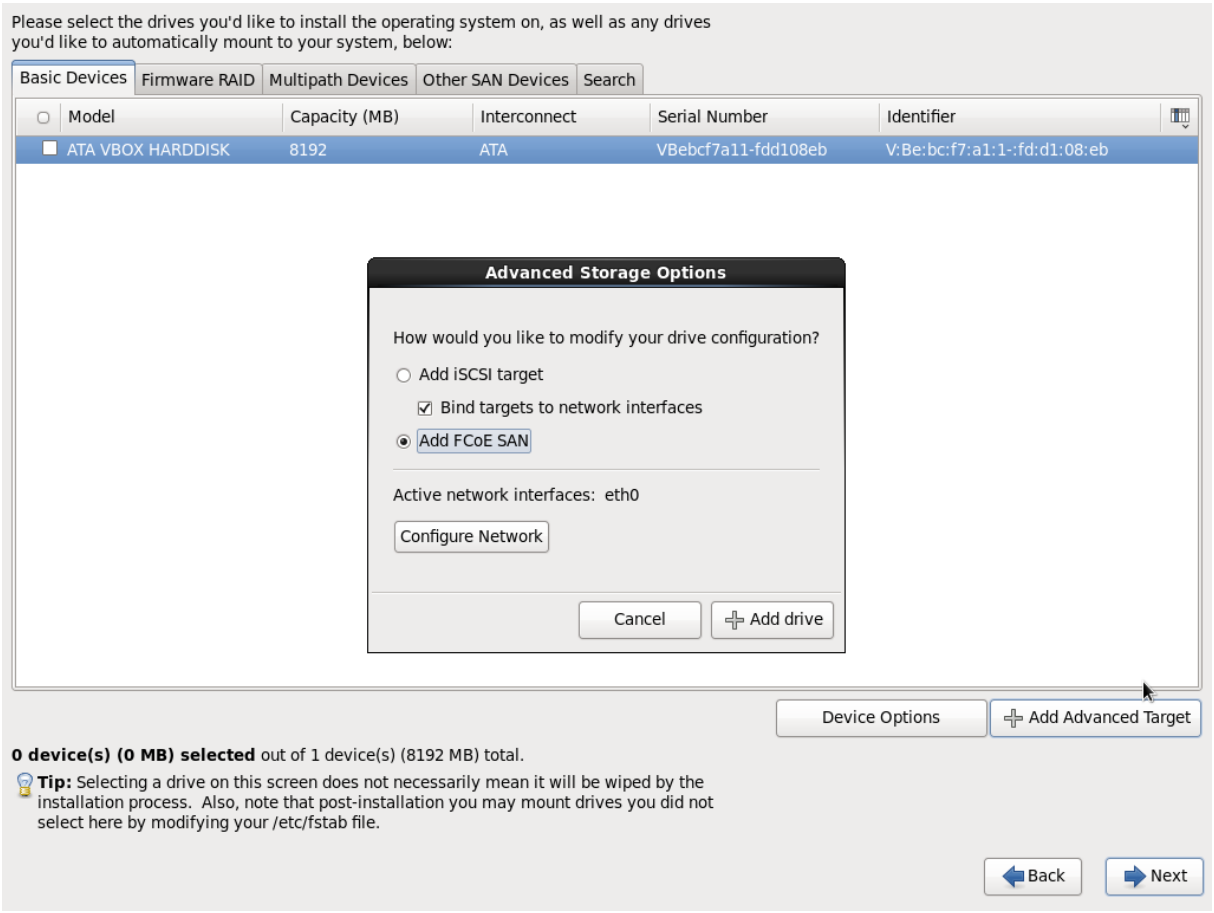
SME Server es una distribución de Linux basada en CentOS. Está pensado para actuar como servidor en pequeñas y medianas empresas. Ofrece varios servicios, como alojamiento web, compartición de archivos e impresoras, Directorio Activo (LDAP), conexión a Internet y firewall, que podemos configurar muy fácilmente desde la interfaz web. Además, para una configuración más avanzada tenemos el sistema de Templates, del que hablaremos más adelante. Está mantenido por una comunidad de desarrolladores y su uso es gratuito incluso para organizaciones comerciales. Se financia solamente a base de donaciones.

Vamos a instalarlo en una máquina virtual. La primera pantalla que vemos nada más introducir el CD de instalación es la siguiente:



Por defecto, el servidor usará los discos disponibles en modo RAID 1. Sin embargo

podemos usar la segunda opción para elegir manualmente el nivel de RAID que queremos. Si seleccionamos 'Advanced installation options', nos llevará a un instalador gráfico en el que tendremos alguna opción más, como instalar el sistema en volúmenes de alacenamiento en la red, como iSCSI.



La instalación borrará los discos completamente. No tenemos la opción de elegir el particionado manualmente. Por ejemplo, una máquina con 5 discos en RAID 5, este es el particionado que obtenemos tras la instalación:

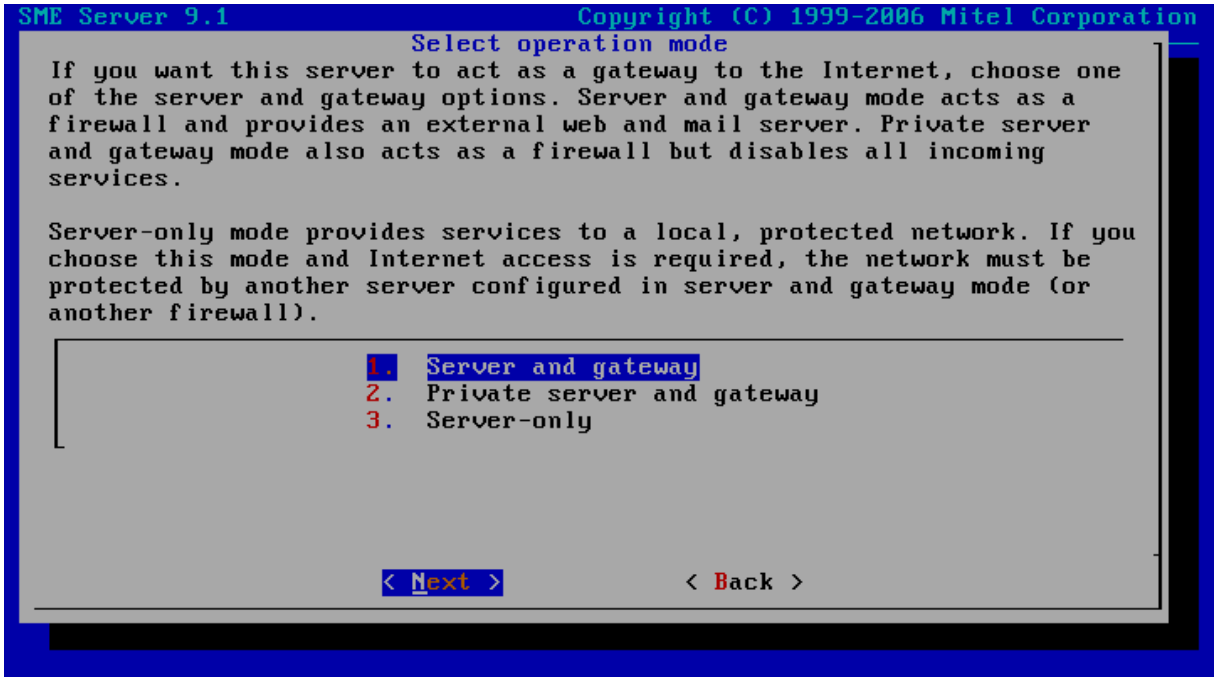
```
[root@sme ~]# df
Filesystem           1K-blocks    Used Available Use% Mounted on
/dev/mapper/main-root
                      30300188 1483636  27270724   6% /
tmpfs                   380036       0    380036   0% /dev/shm
/dev/md0                243759   30164    200799  14% /boot
```

En esta parte de la instalación solo podemos elegir el lenguaje y el tipo de teclado que vamos a utilizar. Cuando el sistema termina de instalarse, se reinicia y nos pregunta por varios aspectos de la configuración.

1.1 Configuración inicial

La primera vez que iniciamos la máquina tras instalar SME server tenemos que configurar el sistema. Nos pregunta por los siguientes parámetros:

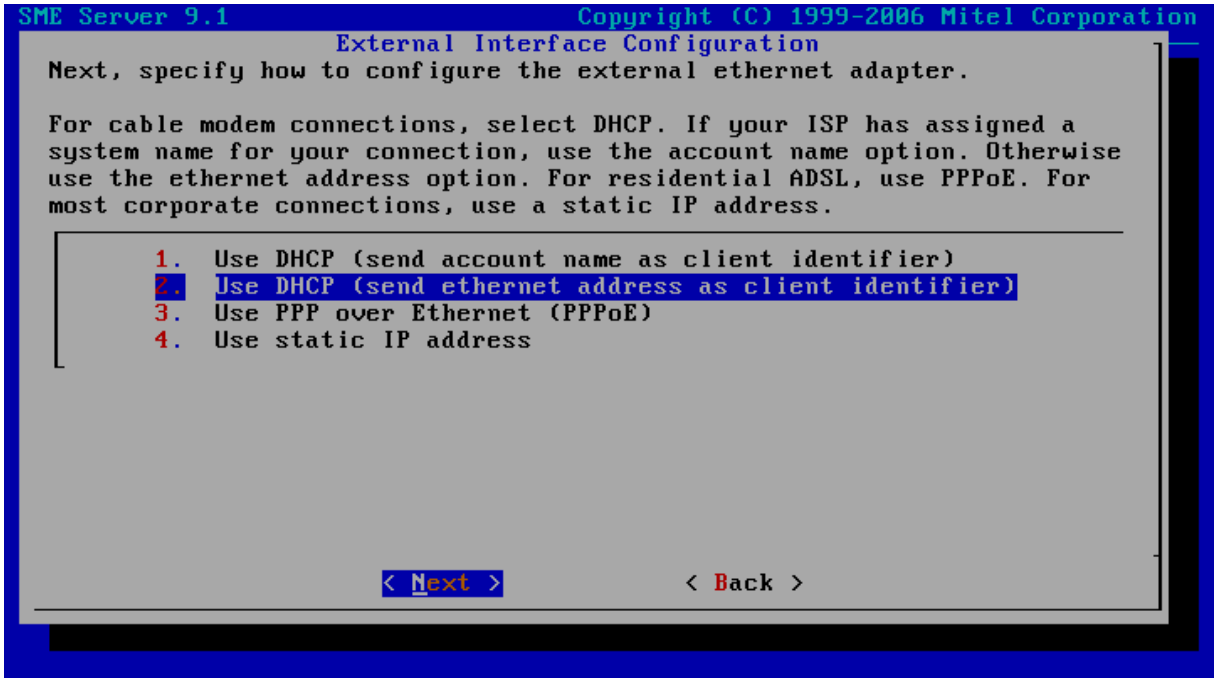
- Contraseña del sistema.
- Nombre del dominio y del sistema.
- Configuración de la interfaz de red interna, asignación de IP y máscara.
- Elegir el modo de operación del servidor.



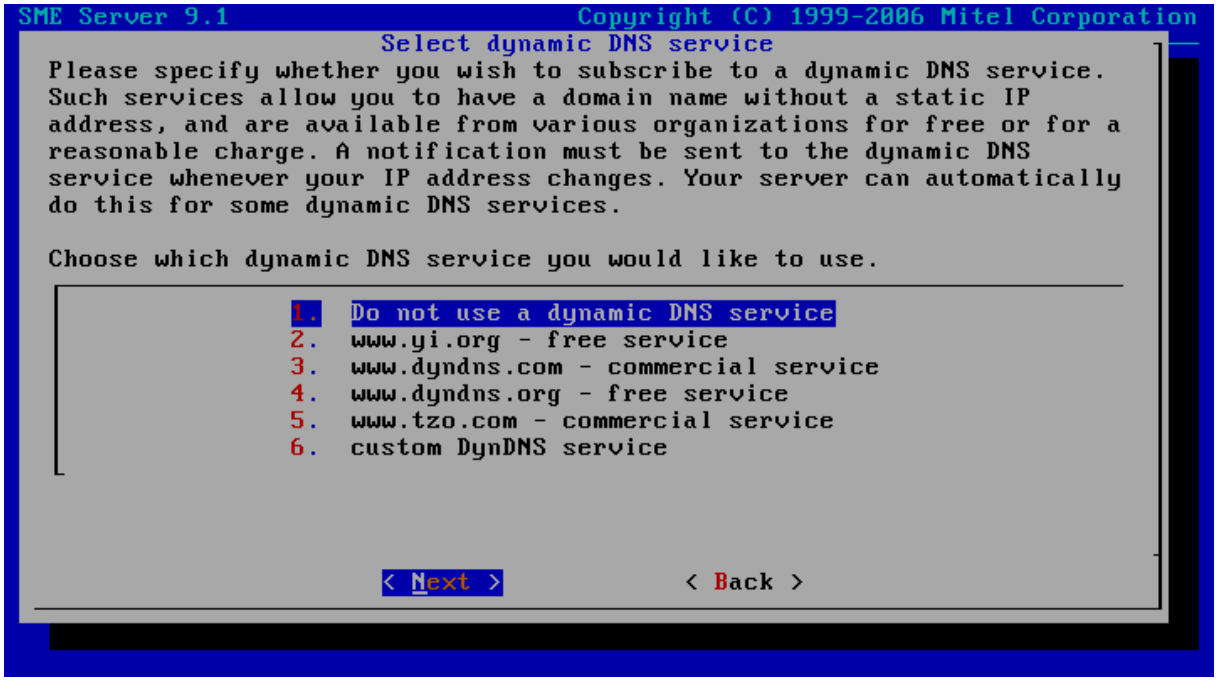
Tenemos 3 opciones:

- **Server and gateway**. El servidor provee los servicios como email, web e intercambio de archivos e impresoras a la red interna y actúa como router y gateway entre la red interna e Internet. También actúa de firewall.
- **Private server and gateway**. Esta opción se diferencia del anterior en que los servicios que proporciona el servidor no son accesibles desde la red externa, y además el firewall contiene reglas adicionales.
- **Server-only mode**. En este modo, el servidor sólo se conecta a la red interna y ofrece los servicios ahí.

- Configuración de la interfaz de red externa. Debemos decirle de qué manera se va a conectar a internet.



- Configuración de un servicio de DNS dinámico.



- Configuración del DHCP para la red interna.
- Por último nos pregunta si queremos usar otro servidor DNS que ya existiera en nuestra red interna.

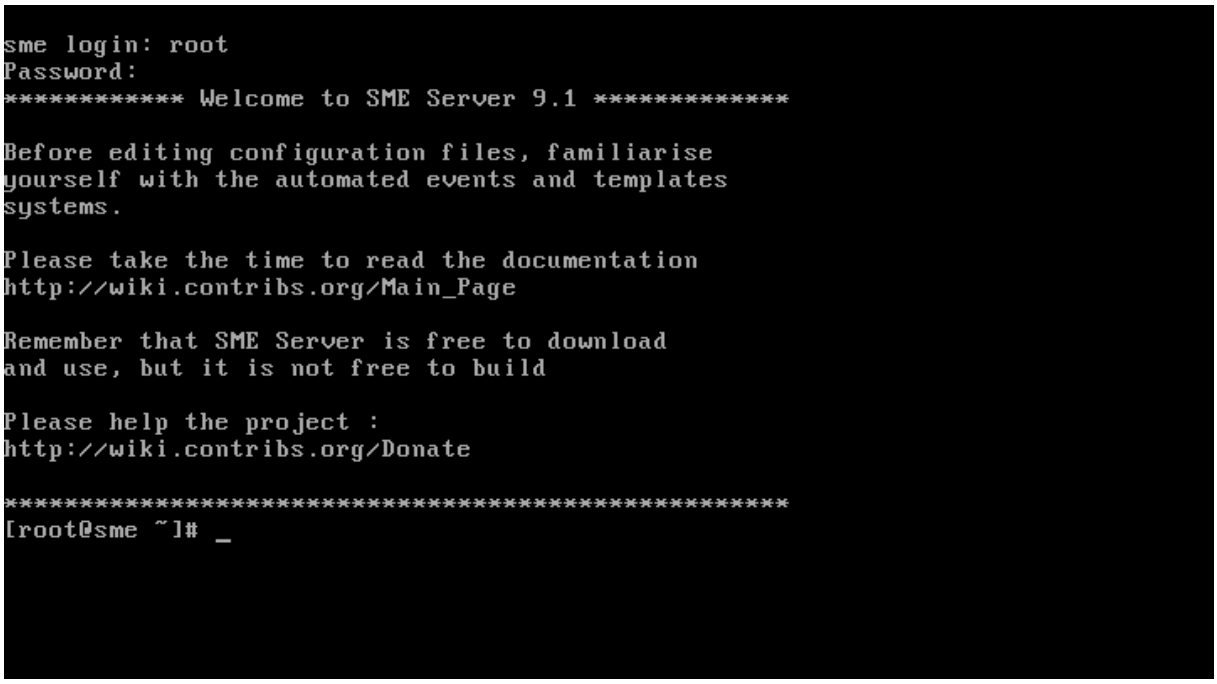
En cuanto terminamos la configuración inicial ya tenemos un servidor completamente funcional ofreciendo distintos servicios. Vamos a ver sus características por defecto y las formas que tenemos de modificarlos.

1.2 Modos de administración

Existen varios modos distintos de adminstrar el SME server.

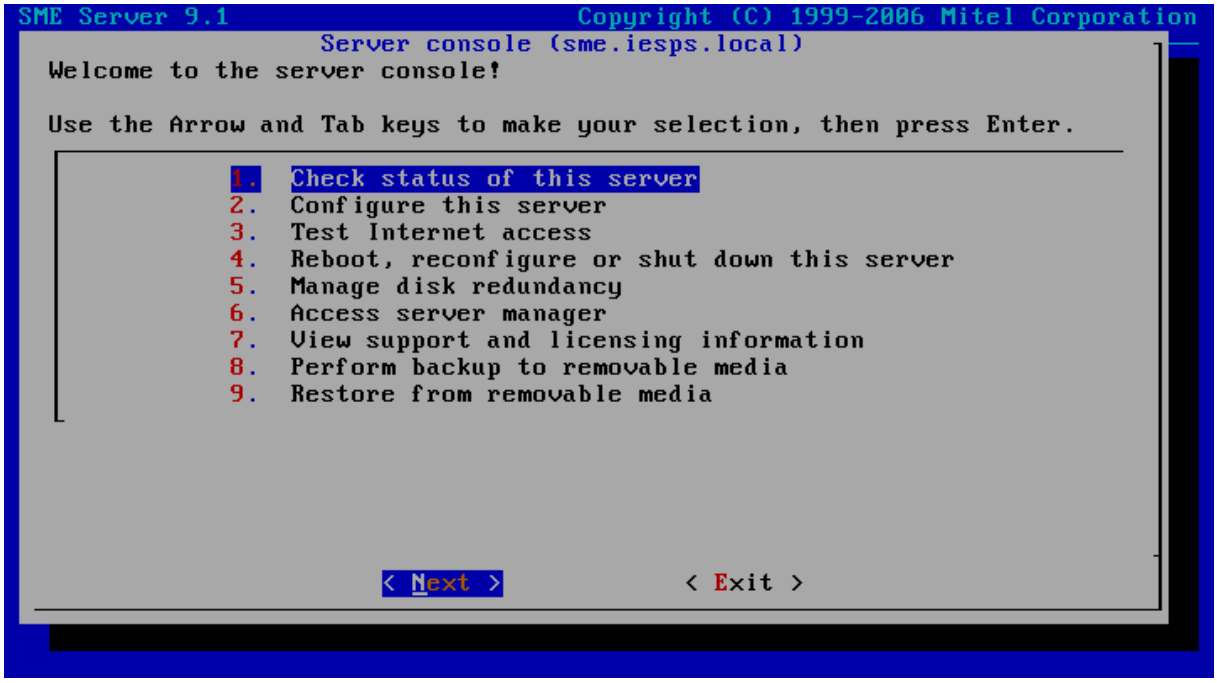
1.2.1 Consola de root de Linux

Al arrancar el sistema, accedemos con el usuario rootz la contraseña de administración. Esto nos proporciona un acceso al sistema operativo mediante la terminal de Linux.



1.2.2 Consola del servidor

Al arrancar el sistema, accedemos con el usuario usuario .ªdminz la contraseña de administración. También se puede acceder desde la consola de root, escribiendo çonsole".

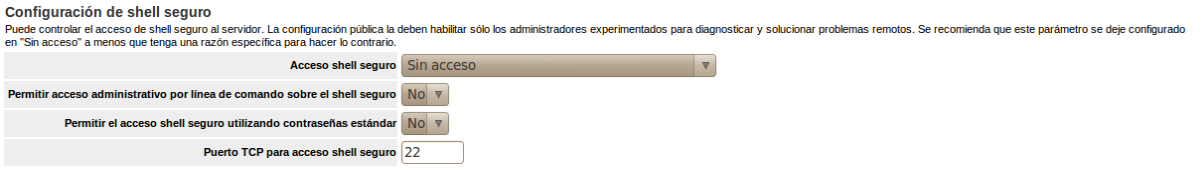


Tiene varias opciones

1. **Check status of this server**: muestra el tiempo que ha estado en marcha el servidor.
2. **Configure this server**: nos lleva otra vez a través de las pantallas de la configuración inicial por si queremos cambiar algo.
3. **Test internet access**: prueba el acceso a internet mandando datos a contribs.org.
4. **Reboot, reconfigure or shut down this server**: reiniciar, reconfigurar o apagar el servidor.
5. **Manage disk redundancy**: muestra el estado de los discos y permite administrar el tipo de RAID. En nuestro caso solo hay un disco instalado... Parece que ha hecho 2 particiones (MIRAR).
6. **Access server manager**: permite acceder a la interfaz de administración web server-manager desde el mismo servidor usando el navegador en modo texto ELinks.
7. **View Support and licensing information**: ver la licencia (GNU GPL) e información sore cómo contactar con contribs.org para el soporte.
8. **Perform backup to removable media**: permite hacer un backup del estado actual del servidor en una unidad USB. La imagen se comprime en un archivo .tgz.
9. **Restore from removable media**: permite recuperar una imagen del servidor anteriormente guardada en una unidad USB.

1.2.3 Aceso remoto

Podemos acceder a estos dos modos de administración desde otro PC a través de SSH, pero está desactivado por defecto:

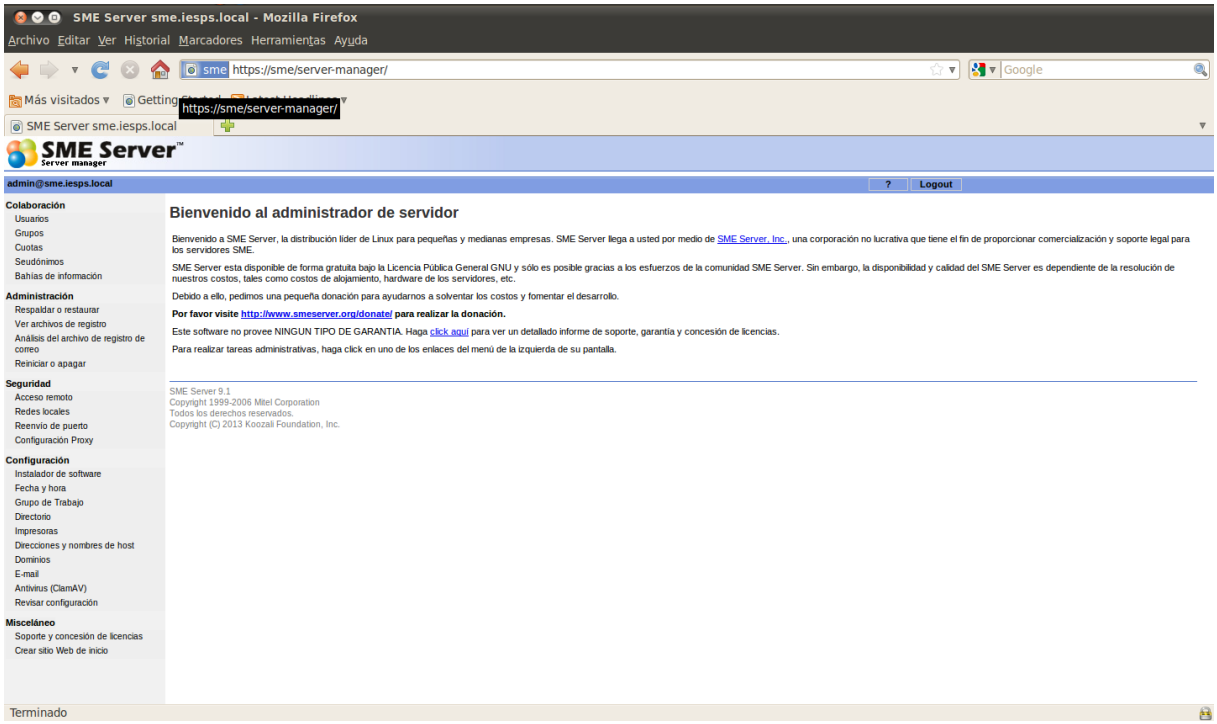


En esta práctica permitiremos el acceso por SSH desde las redes locales para una administración más sencilla. Una vez activado, tendremos disponibles las dos primeras formas de administración que hemos explicado, dependiendo de si nos conectamos como 'root' o como 'admin'. SME server permite también el acceso mediante PPTP para la administración.

1.2.4 Interfaz web server-manager

SME Server provee una interfaz web de administración. Se accede desde el navegador de cualquier ordenador de la red interna con cualquiera de las siguientes URL:

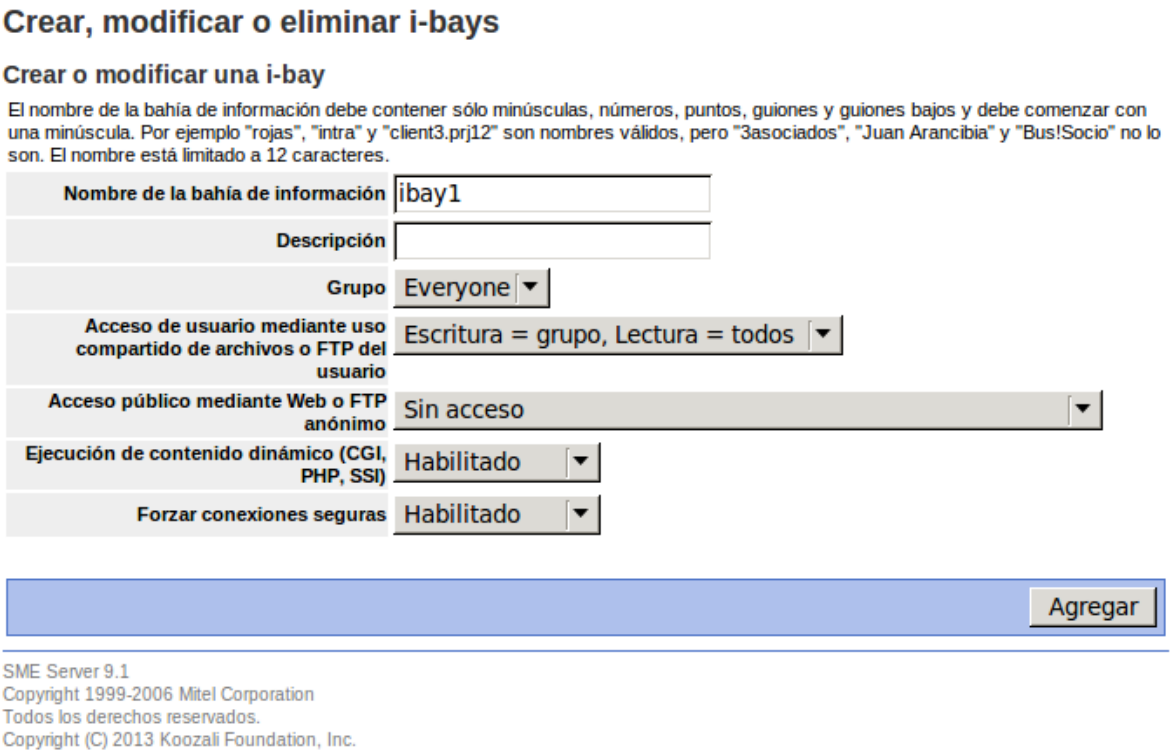
- https://sme/server-manager
- https://192.168.0.254/server-manager



1.3 Servicios y caracerísticas

- Actúa como gateway, y proporciona un firewall, que luego veremos en más detalle. También actúa como servidor DNS.

- Servidor Web.
- Cuentas de usuario y grupos. Cada cuenta de usuario en SME server incluye una cuenta de email y un área de almacenamiento, en principio sin límite máximo, aunque se puede establecer.
- Servidor de Directorio Activo.
- Email. Existe la opción de reenviar los emails de un usuario o grupo a una cuenta externa. Se pueden crear varios pseudónimos para cada cuenta de email. SME nos da la opción de redirigir estos emails a una cuenta externa.
- Almacenamiento. El área de almacenamiento de cada usuario está disponible con Samba. Aparte de esto, se pueden establecer bahías de información, o i-bays, que son directorios compartidos en el disco duro sobre los que podemos establecer fácilmente el control de acceso y los podemos proteger con contraseña.



Los contenidos se encuentran en `/home/e-smith/files/ibays/nombre_ibay/files`.

2 | Herramientas LAT

Las herramientas LAT (Lazy Admin Tools en inglés) son una serie de scripts diseñados para automatizar ciertas tareas de administración en SME server. En este momento las herramientas disponibles son:

Comando	Descripción
lat-users	Añadir/borrar usuarios (y sus directorios)
lat-groups	Añadir/borrar grupos
lat-pseudonyms	Añadir/borrar pseudónimos de email para usuarios indivuduales
lat-ibays	Añadir/borrar ibays (y sus directorios)
lat-quota	Determinar la cuota de uso de disco para usuarios individuales
lat-procmail	Activar o desactivar procmail (herramienta de filtrado de email) para usuarios individuales
lat-hosts	Añadir o quitar nombres de hosts
lat-domains	Crear dominios virtuales
lat-pptp	Activar o desactivar acceso pptp para usuarios individuales
lat-dump	Crear archivos de input para las herramientas anteriores
lat-shadow	Transferir una contraseña encriptada desde un servidor SME a otro

Cada una de ellas tiene su entrada correspondiente en el manual. A continuación instalaremos las herramientas LAT y como ejemplo de su utilización usaremos lat-users para crear varios usuarios.

2.1 Instalación

Las herramientas LAT se encuentran en el repositorio 'smecontribs'. Para instalarlas junto a los paquetes necesarios usaremos el siguiente comando:

```
yum install --enablerepo=smecontribs smeserver-lazy_admin_tools smeserver-
    userpanel smeserver-mailsorting
```

Tras instalarlo, nos advierte de que debemos reiniciar:

```
signal-event post-upgrade; signal-event reboot
```

2.2 lat-users

lat-users permite crear o eliminar cuentas de usuario. Su funcionalidad es equivalente a la opción 'User accounts' en la interfaz web, pero se puede ejecutar desde la línea de

comandos, con lo que puede ser utilizado en scripts.

Siempre que queramos crear usuarios deberemos usar esta herramienta o bien la interfaz web, nunca los comandos `adduser` o `useradd` de Linux, ya que SME Server posee una base de datos propia de todos los usuarios y grupos, que no se actualiza si no se añaden correctamente.

2.2.1 Sintaxis

Depende de cómo lo queramos usar, la sintaxis es una de las siguientes:

```
lat-users -a [-p] -c "user | first | last | password | department | company |
    street | city | tel | forward | email | uid | group1 [| group2..]"
```

```
lat-users -a [-p] -i /ruta/a/users.list
```

```
lat-users -d [-f] -c "usuario"
```

```
lat-users -d [-f] -i /ruta/a/users.list
```

2.2.2 Opciones

Comando	Descripción
`-a, --add`	Añade una cuenta de usuario
`-n`	Crea pseudónimos de email: nombre.apellido y nombre_apellido
`-c "Args.", --command-line="Args."`	Recibe argumentos de la línea de comandos. La lista de argumentos se muestra debajo
`-d, --delete`	Borra una cuenta de usuario. Acepta las wildcards * y ?
`-f`	Fuerza el borrado de usuarios, no pregunta confirmación.
`-h`	Muestra la ayuda.
`-i=ARCHIVO, --input-file=ARCHIVO`	Obtiene la información para la creación o borrado de usuarios desde un archivo.
`-p, --passwords`	Genera contraseñas aleatorias para los nuevos usuarios y las escribe en ./passwords.new.

La lista de argumentos aceptados, en orden, es la siguiente:

Argumento	Descripción
`user`	Nombre de usuario de Linux. Sólo puede contener letras minúsculas, números, puntos y barras bajas. Debe empezar con una letra minúscula. Las wildcards * y ? se pueden usar para eliminar usuarios

`first`	Nombre
`last`	Apellido
`password`	Contraseña (en texto plano)
`department`	Departamento
`company`	Compañía
`street`	Dirección: calle y número
`city`	Dirección: código postal y ciudad
`tel`	Número de teléfono
`forward`	Tipo de entrega de email. Puede ser 'local' , 'forward' o 'both'
`email`	Dirección a la que reenviar el email
`uid`	ID de usuario. Si se omite, se genera automáticamente
`group(s)`	Grupos a los que el usuario debe ser añadido. Si no existe el grupo, se creará

Los campos `user`, `first` y `last` son obligatorios.

2.2.3 Ejemplos

En el archivo /usr/doc/lazy-admin-tools/example.users tenemos varios ejemplos de sintaxis válidas para la opción `-c`

```
michiel |Michiel    |Blotwijk  |       |Development   |Altiplano bvba |Muntweide 7 |B-1
785 Brussegem |+32 2 305.70.76 |forward    |Michiel@Altiplano.Be | 9001
marco   |Marco      |Blanc     |secret |Sales         |Altiplano bvba |Muntweide 7 |B-1
785 Brussegem |+32 2 305.70.76
ejo     |Ejo        |Burger |||||||| 9002
gilbert |Gilbert    |Volk   ||||||||| allstaff| accounting | management
nathalie|Nathalie   |Butterfly
susanne |Susanne    |Rodin
delete_me*
delet?_me2
```

Figura 2.1: Ejemplos de creación y borrado de usuarios

Las líneas primera y segunda en realidad serían una única, pero se corta. Lo mismo pasa con las líneas tercera y cuarta. Las dos últimas sólo son válidas para el borrado de usuarios.

Si queremos crear una lista completa de usuarios como por ejemplo los que usamos en clase, desde alu01 hasta alu40, tenemos varias alternativas. La primera sería escribir un script que los vaya creando uno a uno usando `lat-users -a`:

```
#!/bin/bash

for I in $(seq -w 1 1 40)
do
    lat-users -a -c "alu$I|Alumno|$I|alu$I| | | | | | | |asir2d"
done
```

Los grupos alu01, ..., alu40 se crean automáticamente y se asignan como grupo primario de los correspondientes usuarios.

Otra opción es hacer un shell script que escriba un archivo como el users.list que hemos visto anteriormente, y usar luego `lat-users -a -i ./users.list`:

```
#!/bin/bash

for I in $(seq -w 1 1 40)
do
    echo "alu$I|Alumno|$I|alu$I| | | | | | | |asir2d" >> ./users.list
done
```

Puede ocurrir que hayamos escrito el comando erróneamente y se haya creado un usuario o un grupo 'a medias', de manera que no existirá como tal pero seguirá estando en la base de datos, con lo cual no podremos crearlo otra vez. Para eliminar completamente un usuario o grupo en estos casos usaremos:

```
signal-event user-delete usuario
db accounts delete usuario
```

O bien

```
signal-event group-delete grupo
db accounts delete grupo
```

Un script que viene bien para borrar completamente todos los usuarios añadidos en el ejemplo:

```
#!/bin/bash

lat-users -d -f -c "alu*"

for I in $(seq -w 1 1 40)
do
    signal-event user-delete alu$I
    db accounts delete alu$I
done

signal-event reboot
```

3 | Arquitectura del SME Server

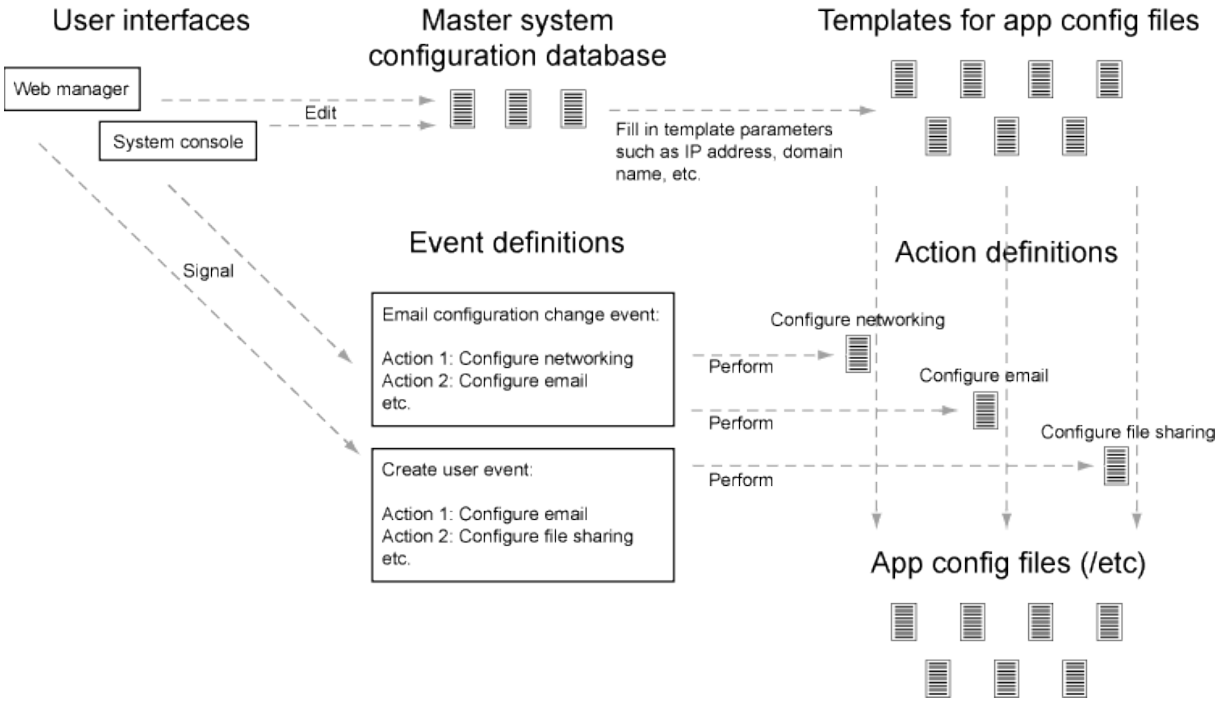
La arquitectura interna del SME Server se basa en cuatro componentes:

- Interfaces de administración, tanto por consola como web.
- Bases de datos de configuración.
- El sistema de templates, que es usado para generar archivos de configuración.
- Eventos y acciones.

Cuando un usuario configura alguno de los aspectos del servidor, la máquina configura automáticamente las aplicaciones que son relevantes a ese cambio. Esto se realiza en varios pasos:

- La interfaz de usuario cambia valores en las bases de datos de configuración. En estas bases de datos se almacena información que describe el estado del sistema, como las asignaciones de direcciones IP, los nombres de dominio, las cuentas de usuario...
- Después de haber cambiado estos valores, la interfaz de usuario envía una señal de evento para realizar los cambios en las aplicaciones. Por ejemplo, si hacemos cambios en la configuración del email, el evento que se señala es el 'email-update'. Estos eventos son colecciones de scripts, que se ejecutan en un orden determinado para producir las modificaciones deseadas.
- Estos scripts actualizan los ficheros de configuración de las aplicaciones. Los generan usando las templates, que no son más que plantillas usadas para este propósito. Los scripts de actualización leen las bases de datos de configuración, de manera que los nuevos archivos generados contienen nuestras últimas modificaciones.
- Por último las aplicaciones son avisadas de que su configuración ha sido cambiada, por lo que releen los archivos o se reinician, según convenga.

Aquí tenemos una imagen que describe este proceso:



3.1 Bases de datos de configuración

Todos los parámetros de configuración que son modificables por el usuario son almacenados aquí. Las entradas pueden ser de dos tipos:

- Simples: constan de un par clave/valor.

```
[root@sme ~]# config show AccessType
AccessType=dedicated
[root@sme ~]# config show ConsoleMode
ConsoleMode=login
[root@sme ~]# config show TimeZone
TimeZone=Europe/London
[root@sme ~]#
```

- Complejas: constan de una clave, un tipo y una colección de pares propiedad/valor.

```
[root@sme ~]# config show dhcpd
dhcpd=service
    Bootp=deny
    end=192.168.0.100
    start=192.168.0.1
    status=enabled
```

3.1.1 Acceso a las bases de datos

- **Acceso mediante la línea de comandos**

El comando para acceder a los datos de las bases de datos de configuración es `db`. Podemos ver una pequeña ayuda tecleando `db` sin más:

```
[root@sme ~]# db
usage:
    /sbin/e-smith/db dbfile keys
    /sbin/e-smith/db dbfile print [key]
    /sbin/e-smith/db dbfile show [key]
    /sbin/e-smith/db dbfile get key
    /sbin/e-smith/db dbfile set key type [prop1 val1] [prop2 val2] ...
    /sbin/e-smith/db dbfile setdefault key type [prop1 val1] [prop2 val2] ...
    /sbin/e-smith/db dbfile delete key
    /sbin/e-smith/db dbfile printtype [key]
    /sbin/e-smith/db dbfile gettype key
    /sbin/e-smith/db dbfile settype key type
    /sbin/e-smith/db dbfile printprop key [prop1] [prop2] [prop3] ...
    /sbin/e-smith/db dbfile getprop key prop
    /sbin/e-smith/db dbfile setprop key prop1 val1 [prop2 val2] [prop3 val3] ...
    /sbin/e-smith/db dbfile delprop key prop1 [prop2] [prop3] ...
```

Las bases de datos existentes son `accounts`, `networks`, `configuration`, `domains` y `hosts`. Además, tenemos el comando `config show`, que es un alias para `db configuration show`.

```
[root@sme ~]# db networks show
192.168.0.0=network
    Mask=255.255.255.0
    SystemLocalNetwork=yes
[root@sme ~]# config show LocalIP
LocalIP=192.168.0.254
[root@sme ~]# db configuration show LocalIP
LocalIP=192.168.0.254
[root@sme ~]# db accounts show alu01
alu01=user
    City=
    Company=
    Dept=
    EmailForward=local
    FirstName=Alumno
    ForwardAddress=
    LastName=01
    PasswordSet=yes
    Phone=
    Shell=/usr/bin/rssh
    Street=
    Uid=5001
    VPNClientAccess=no
```

- **Acceso desde un script de Perl**

SME Server incorpora una API para Perl para acceder a las bases de datos de configuración. Para ver ayuda y ejemplos, podemos acceder a la documentación usando el comando `perldoc`:

```
perldoc esmith::ConfigDB
perldoc esmith::AccountsDB
perldoc esmith::HostsDB
perldoc esmith::NetworksDB

perldoc esmith::DB
```

3.2 Acciones y eventos

3.2.1 Acciones

Una acción es un programa que ejecuta una única tarea, como editar un archivo de configuración o reconfigurar un servicio. Las acciones son llamadas marcando un evento, nunca directamente. Están en el directorio `/etc/e-smith/events/actions`.

```
[root@sme actions]# pwd
/etc/e-smith/events/actions
[root@sme actions]# ls
adjust-dovecot                mysql-load-tables
adjust-services               navigation-conf
cleanup-domains               pptp-interface-access
cleanup-unix-user-group       printer-create
clear-pptp-interfaces         printer-delete
conf-hostsdb                  purge-domain
conf-linktotranslations       purge-junkmail-folders
conf-migrate-hosts            purge-old-logs
conf-modules                  qmail-delete-group
conf-routes                   qmail-ipup
conf-startup                  qmail-update-group
conf-timezone                 qmail-update-user
conf-userpanelsymlinks        reboot
```

3.2.2 Eventos

Los eventos son un mecanismo que permite al sistema ejecutar una serie de acciones en respuesta a un cambio. Están asociados a una lista de acciones que se ejecutan en un orden determinado. Se encuentran en la carpeta `/etc/e-smith/events`

Por ejemplo, en el caso de un cambio de la dirección IP externa, se llamaría al evento `ip-change`. Este evento se compone de lo siguiente:

```
[root@sme ip-change]# pwd
/etc/e-smith/events/ip-change
[root@sme ip-change]# ls -l
total 8
lrwxrwxrwx 1 root root   26 Apr 17 12:24 S03set-external-ip -> ../actions/set-external-
ip
lrwxrwxrwx 1 root root   21 Apr 17 12:24 S85update-dns -> ../actions/update-dns
drwxr-xr-x 2 root root 4096 Apr 16 14:16 services2adjust
drwxr-xr-x 4 root root 4096 Jan 13 17:48 templates2expand
```

- Enlaces simbólicos a dos acciones, una que escribe la nueva IP externa en la base de datos `configuration` y otra que actualiza el servicio de DNS dinámico que tuviéramos configurado con la nueva IP de nuestra máquina. El prefijo que llevan los links sirve para establecer en qué orden se ejecutarán. Este es el cuerpo del script que cambia la IP externa:

```
package esmith;

use strict;
use Errno;
use esmith::ConfigDB;

my $db = esmith::ConfigDB->open or die "Couldn't open ConfigDB\n";

#------------------------------------------------------------
# Set $ExternalIP in configuration hash, for use by templates
#------------------------------------------------------------
my $event = $ARGV [0];
my $newip = $ARGV[1];

$db->set_value('ExternalIP', $newip);
$db->set_prop('ExternalInterface', 'IPAddress', $newip);

exit (0);
```

Aquí podemos ver cómo se usa el API de Perl para acceder a la base de datos `configuration` y cambiar valores. Casi todas las acciones son llamadas siempre con 2 argumentos: el nombre del evento que las ha llamado y el nuevo valor que se va a asignar. La IP externa del equipo es guardada en 2 registros diferentes de esta base de datos, por razones de compatibilidad con versiones anteriores de SME server:

```
[root@sme ip-change]# db configuration show ExternalIP
ExternalIP=10.0.2.15
[root@sme ip-change]# db configuration show ExternalInterface
ExternalInterface=interface
    Configuration=DHCPEthernetAddress
    Driver=e1000
    Gateway=
    IPAddress=10.0.2.15
    Name=eth0
    Netmask=255.255.255.0
```

- Acciones implícitas. La mayoría de eventos contienen dos tareas comunes: expandir los templates necesarios y reajustar los servicios implicados. Para ello se ejecutan las acciones `generic_template_expand` y `adjust-services`, cuyo código se encuentra en `/etc/e-smith/events/actions`. Los subdirectorios que encontramos en la carpeta de nuestro evento son usados por estas dos acciones:

  - Directorio `services2adjust`, para la acción `adjust-services`. Contiene links de los servicios que se tienen que reajustar y la acción que se debe efectuar sobre ellos.

```
[root@sme services2adjust]# ls -l
total 0
lrwxrwxrwx 1 root root 6 Apr 16 14:16 masq -> adjust
lrwxrwxrwx 1 root root 7 Apr 16 14:16 ntpd -> restart
lrwxrwxrwx 1 root root 7 Apr 16 14:16 pptpd -> sigterm
lrwxrwxrwx 1 root root 6 Apr 16 14:16 qmail -> sighup
lrwxrwxrwx 1 root root 7 Apr 16 14:16 tinydns -> sigusr2
```

  - Directorio `templates2expand`. Lista de los archivos de configuración que tienen que volver a ser regenerados desde las plantillas.

```
[root@sme templates2expand]# pwd
/etc/e-smith/events/ip-change/templates2expand
[root@sme templates2expand]# ls -lR
.:
total 8
drwxr-xr-x 3 root root 4096 Apr 17 12:24 etc
drwxr-xr-x 4 root root 4096 Jan 13 17:48 var

./etc:
total 4
-rw-r--r-- 1 root root    0 Feb 19  2013 dhcpd.conf
-rw-r--r-- 1 root root    0 Feb  6  2015 fetchmail
-rw-r--r-- 1 root root    0 Jan 31 22:29 hosts.allow
-rw-r--r-- 1 root root    0 Jan 31 22:29 hosts.deny
drwxr-xr-x 2 root root 4096 Apr 16 14:16 ppp
-rw-r--r-- 1 root root    0 Feb 19  2013 pptpd.conf
-rw-r--r-- 1 root root    0 Jun  7  2013 proftpd.conf
-rw-r--r-- 1 root root    0 Jan 31 22:29 securetty
-rw-r--r-- 1 root root    0 Jan 31 22:29 services
-rw-r--r-- 1 root root    0 Jan 31 22:29 shells
-rw-r--r-- 1 root root    0 Feb  6  2015 startmail

./etc/ppp:
total 0
-rw-r--r-- 1 root root 0 Feb 19  2013 ip-down.local
-rw-r--r-- 1 root root 0 Feb 19  2013 ip-up.local
-rw-r--r-- 1 root root 0 Feb 19  2013 options.pptpd
```

Por defecto, la acción `generic_template_expand` se ejecua con prioridad S05, y la acción `adjust-services` con S90. Por ello se recomienda que las acciones propias que queramos incluir estén entre S10 y S80. Sabiendo esto podemos comprender el orden en el que se ejecutan las acciones en nuestro ejemplo, el evento de cambio de IP externa:

1. Se cambia el valor de IP externa en la base de datos de configuración.
2. Se actualizan los archivos de configuración de los servicios y programas implicados.
3. Si tenemos configurado servicio de DNS dinámico, se actualiza (se advierte al servidor de nuestro cambio de IP).
4. Se reconfiguran todos los servicios con la nueva IP.

3.2.3 Señalar un evento

Para ejecutar todas las acciones de un evento usamos el comando `signal-event`, seguido del nombre del evento. Este comando además escribe todo el output en el log del sistema `messages`.

```
signal-event ip-change 216.58.210.35
```

El comando `signal-event` no suele tener más argumentos que el nombre del evento, ya que en general se prefiere hacer los cambios en los archivos de configuración antes de llamar al evento.

3.3 Templates

El sistema de templates (plantillas) presente en SME server sirve para generar los archivos de configuración de las aplicaciones. Proporciona un método uniforme para cambiar estos archivos, sin tener que preocuparnos de la sintaxis particular que usa cada uno. De hecho, en SME server no debemos editar nunca los archivos de configuración a mano, ya que se sobreescribirán con los generados por las plantillas en algunas situaciones, por ejemplo si activamos algún evento que requiera el ajuste de esas aplicaciones, o si reiniciamos del sistema.

Las templates se encuentran en el directorio `/etc/e-smith/templates`.

```
[root@sme ~]# ls /etc/e-smith/templates
boot  etc  home  root  usr  var
```

Aquí nos ecntramos una estructura de directorios que se corresponde con la del sistema. Por ejemplo, la template para `/etc/hosts` estará en `/etc/e-smith/templates/etc/hosts`.

Las plantillas están almacenadas en **fragmentos**. Es decir, en nuestro ejemplo, `/etc/e-smith/templates/etc/hosts` no es un archivo sino un directorio con diferentes archivos. Estos fragmentos se concatenan según el orden ASCII de sus nombres, y el archivo resultante es el que se encarga de generar el archivo de configuración de la aplicación.

```
[root@sme hosts]# pwd
/etc/e-smith/templates/etc/hosts
[root@sme hosts]# ls
10localhost  20hostname
```

La plantilla para `/etc/hosts` es fácil de entender.

```
[root@sme hosts]# cat 10localhost
127.0.0.1       localhost
[root@sme hosts]# cat 20hostname
{
    $OUT .= "$LocalIP\t";
    $OUT .= " ${SystemName}.${DomainName}";
    $OUT .= " ${SystemName}";
}
```

El primer archivo contiene texto, que se añade sin más al archivo de configuración. El segundo archivo contiene un pequeño script de Perl (todo lo que va entre llaves se ejecuta), que usa valores de la base de datos de configuración.

3.3.1 Expansión de templates

Para regenerar los archivos de configuración cuya plantilla hemos modificado, debemos expandir esa plantilla:

```
expand-template /etc/archivo.conf
```

También debemos reiniciar el servicio en cuestión. Se puede hacer con el siguiente comando:

```
sv t /ruta/del/servicio
```

Algunos eventos combinan la expansión de ciertas plantillas y el reinicio de todos los servicios afectados. Por ejemplo, si modificamos algo en la configuración del email, llamaremos al siguiente evento:

```
signal-event email update
```

Si dudamos qué templates debemos expandir o qué servicios debemos reiniciar, podemos marcar los siguientes eventos, que expandirán todas las plantillas y reiniciarán el sistema:

```
signal-event post-upgrade
signal-event reboot
```

3.4 Ejemplo práctico: php.ini

Si queremos modificar una de las templates o fragmentos que vienen presentes en la distribución, deberemos copiarla, junto con la estructura de directorios adecuada, desde `/etc/e-smith/templates` a `/etc/e-smith/templates-custom`, y editarla aquí. En caso de que exista una template en cada uno de estos dos directorios con el mismo nombre, solo se usará la de `templates-custom`. Esto es así para que en el caso de que cometamos algún fallo podamos revertir el cambio fácilmente, sin más que borrar la plantilla en `templates-custom`. Si queremos introducir una template nueva, la crearemos en `templates-custom`, con el nombre adecuado, teniendo en cuenta que luego se concatenará con las demás (también las de `/etc/e-smith/templates`) según el orden ASCII de sus nombres.

Veamos el aspecto que tiene el archivo `/etc/php.ini` en el sistema:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; DO NOT MODIFY THIS FILE! It is updated automatically by the
; SME Server software. Instead, modify the source template in
; an /etc/e-smith/templates-custom directory. For more
; information, see http://www.e-smith.org/custom/
;
; copyright (C) 2002 Mitel Networks Corporation
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

[PHP]

engine                                  = On
short_open_tag                          = On;
asp_tags                                = Off
precision                               = 14
y2k_compliance                          = Off
output_buffering                        = Off
output_handler                          =
implicit_flush                          = Off
allow_call_time_pass_reference          = On
safe_mode                               = Off
safe_mode_exec_dir                      =
safe_mode_allowed_env_vars              = PHP_
safe_mode_protected_env_vars            = LD_LIBRARY_PATH
disable_functions                       =
highlight.string                        = #DD0000
highlight.comment                       = #FF8000
highlight.keyword                       = #007700
highlight.bg                            = #FFFFFF
highlight.default                       = #0000BB
highlight.html                          = #000000
; Default expose_php to Off for security reasons
expose_php                              = Off
max_execution_time                      = 30
memory_limit                            = 32M
mysql.allow_persistent                  = On
error_reporting                         = E_ALL & ~E_NOTICE
display_errors                          = Off
display_startup_errors                  = Off
```

Supongamos que queremos cambiar el valor de la directiva `display_errors` a `On`. Debemos buscar la plantilla, dentro de `/etc/e-smith/templates`, en la que se encuentra esta directiva.

```
[root@sme ~]# cat /etc/e-smith/templates/etc/php.ini/30ErrorHandling
mysql.allow_persistent                  = On
error_reporting                         = E_ALL & ~E_NOTICE
display_errors                          = Off
display_startup_errors                  = Off
log_errors                              = On
error_log                               = syslog
track_errors                            = Off
warn_plus_overloading                   = Off
```

Copiamos este fragmento de plantilla en el directorio `/etc/e-smith/templates-custom/etc/php.ini` (lo creamos si no existía) y lo modificamos allí, cambiando la directiva a `On`. Así es como queda:

```
[root@sme ~]# cat /etc/e-smith/templates-custom/etc/php.ini/30ErrorHandling
mysql.allow_persistent                  = On
error_reporting                         = E_ALL & ~E_NOTICE
display_errors                          = On
display_startup_errors                  = Off
log_errors                              = On
error_log                               = syslog
track_errors                            = Off
warn_plus_overloading                   = Off
```

Para que se modifique el `php.ini` real, tenemos que expandir la plantilla y reiniciar el servicio `httpd-e-smith`, que es el servidor web.

```
expand-template /etc/php.ini
sv t /service/httpd-e-smith
```

Podemos ver que el cambio se ha realizado correctamente:

```
[root@sme ~]# cat /etc/php.ini
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; DO NOT MODIFY THIS FILE! It is updated automatically by the
; SME Server software. Instead, modify the source template in
; an /etc/e-smith/templates-custom directory. For more
; information, see http://www.e-smith.org/custom/
;
; copyright (C) 2002 Mitel Networks Corporation
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

[PHP]

engine                                  = On
short_open_tag                          = On;
asp_tags                                = Off
precision                               = 14
y2k_compliance                          = Off
output_buffering                        = Off
output_handler                          =
implicit_flush                          = Off
allow_call_time_pass_reference          = On
safe_mode                               = Off
safe_mode_exec_dir                      =
safe_mode_allowed_env_vars              = PHP_
safe_mode_protected_env_vars            = LD_LIBRARY_PATH
disable_functions                       =
highlight.string                        = #DD0000
highlight.comment                       = #FF8000
highlight.keyword                       = #007700
highlight.bg                            = #FFFFFF
highlight.default                       = #0000BB
highlight.html                          = #000000
; Default expose_php to Off for security reasons
expose_php                              = Off
max_execution_time                      = 30
memory_limit                            = 32M
mysql.allow_persistent                  = On
error_reporting                         = E_ALL & ~E_NOTICE
display_errors                          = On
display_startup_errors                  = Off
```

4 | Iptables

Iptables es un programa que proporciona una interfaz de usuario para configurar Netfilter, que es el subsistema presente en el kernel de Linux para filtrado y manipulación de paquetes. Casi cualquier firewall de terceros que podemos descargar e instalar en Linux, como UFW o Firestarter, no son más que front-ends para Iptables.

Iptables es usado para crear y administrar reglas que proporcionan, entre otras cosas, capacidad para manipular los paquetes, control de la traducción NAT y seguimiento de conexiones. Iptables es un firewall de los llamados 'stateful', ya que mantiene el control de las conexiones establecidas por nuestro PC.

4.1 Estructura y funcionamiento de iptables

Iptables permite escribir **reglas** para control y manipulación de paquetes. Estas reglas se agrupan en **cadenas**, que a su vez pertenecen a una de las **tablas** presentes.

Las tablas disponibles son las siguientes:

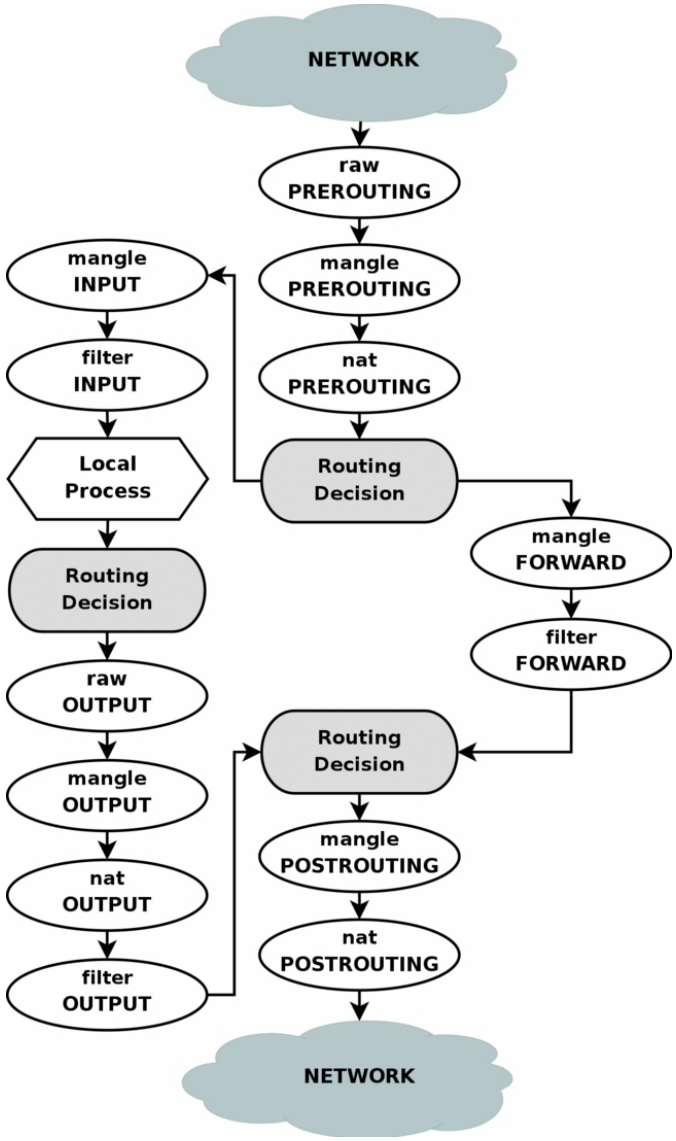
- **Filter**, la tabla por defecto, provee comandos para filtar y aceptar o rechazar paquetes.
- **Nat**, provee comandos para modificar la forma en que la máquina hace la traducción de direcciones.
- **Mangle**, permite la modificación de los encabezamientos de los paquetes.
- **Security**, proporciona reglas de control de acceso.
- **Raw**, contiene herramientas para el seguimiento de conexiones.

Cada una de las tablas tiene diferentes cadenas, a las que añadiremos nuestras reglas:

Tabla	Cadenas
filter	INPUT FORWARD OUTPUT
NAT	PREROUTING INPUT OUTPUT POSTROUTING

Tabla	Cadenas
mangle	PREROUTING INPUT FORWARD OUTPUT POSTROUTING
security	INPUT FORWARD OUTPUT
raw	PREROUTING OUTPUT

Para comprender bien el funcionamiento de Iptables debemos pensar desde el punto de vista de las cadenas en lugar de las tablas. El kernel de Linux procesa los paquetes en un orden que se corresponde con las cadenas. En esta imagen podemos ver las etapas por las que pasan los paquetes:



Hay tres posibilidades diferentes según el camino que sigan los datos:

1. **Paquetes destinados a nuestra máquina**

   Cuando un paquete entra por una interfaz, inmediatamente se compara con las reglas que tengamos en la cadena PREROUTING, en el orden indicado en la imagen: primero las que están en la tabla *raw*, luego las que están en la tabla *mangle* y por último las que están en la tabla *nat*. En este momento el paquete puede ser modificado. Por ejemplo, podemos cambiar la IP de destino y mandarlo a otro sitio, de manera que no será procesado por nuestro sistema. Se recomienda no usar nunca la cadena PREROUTING para filtrar, ya que en determinadas situaciones los paquetes no son filtrados correctamente.

   Después, se toma la decisión de enrutamiento, con lo que se sigue el camino que vemos en la imagen hacia la izquierda. Se compara primero con la cadena INPUT de las tablas que la contienen. En este momento es cuando se filtran y modifican los paquetes. Finalmente se entrega al proceso local para que lo use.

2. **Paquetes generados por nuestra máquina**

   Cuando el paquete se ha generado, se decide el enrutamiento y se elige qué dirección de origen usar y la interfaz por la que se va a enviar. Luego se compara con las reglas presentes en la cadena OUTPUT de las diferentes tablas. Con ellas podemos filtrar (tabla *filter*) y cambiar las direcciones de origen y destino (tabla *nat*), entre otras cosas.

   Posteriormente se vuelve enrutar, ya que podríamos haber modificado el paquete, y se compara con las reglas de la cadena POSTROUTING. Finalmente se envía a la interfaz adecuada para que sea transmitido.

3. **Paquetes destinados a otro host**

   En este caso el paquete entra y se compara con las reglas de la cadena PREROUTING, como hemos explicado antes, y luego se enruta. Si resulta que va dirigido a otro host, se compara con las reglas de la cadena FORWARD, en el orden indicado. Aquí podemos filtrar paquetes (tabla *filter*) y modificarlos (tabla *mangle*). Luego se vuelve a enrutar y se le aplican las reglas de la cadena POSTROUTING.

4.2 El comando iptables

Veamos unos cuantos ejemplos para comprender la sintaxis:

```
iptables -t nat -L
iptables -L
iptables -t nat -L PREROUTING
```

El primer comando muestra las reglas presentes en la tabla *nat*. La tabla por defecto es *filter*, por lo tanto se usará ésta siempre que no se especifque otra: el segundo ejemplo mostrará todas las reglas presentes en *filter*. Se puede añadir el nombre de una cadena después para que solo muestre las reglas contenidas en ella.

```
[root@sme ~]# iptables -t nat -L PREROUTING
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
PortForwarding  all  --  anywhere             anywhere
SMTPProxy  tcp  --  anywhere             anywhere            tcp dpt:smtp
TransProxy  tcp  --  anywhere             anywhere            tcp dpt:http
```

```
iptables -t mangle -F INPUT
```

Borra todas las reglas de la cadena INPUT de la tabla *mangle*. Si se omite el nombre de la tabla, se utilizará *filter*. Si se omite el nombre de la cadena, se borrarán las reglas de todas las cadenas.

```
iptables -t filter -P INPUT DROP
```

Cambia la política por defecto de la cadena INPUT de la tabla *filter* a DROP. Los paquetes que se hayan comparado con todas las reglas de esta cadena y no se hayan encontrado con una orden ACCEPT serán rechazados.

```
iptables -t nat -N miCadena
```

Crea una nueva cadena definida por el usuario, llamada *miCadena*, en la tabla *nat*. Estas cadenas deben ser llamadas desde las 5 originales, más adelante veremos cómo.

4.2.1 Añadiendo reglas

El comando básico para añadir una regla a una cadena de una tabla es

```
iptables -t tabla -A CADENA descripición
```

La descripción de una regla se especifica escribiendo el valor de varios parámetros. Los parámetros disponibles son:

Parámetro	Descripción
`-p` *protocolo*	Especifica el protocolo. Puede ser *tcp*, *udp* o *icmp*, entre otros. Se puede usar un símbolo de exclamación, !, delante del protocolo para invertir la búsqueda.
`-s` *IP/máscara*	Especificación del origen. Puede ser un nombre de host en lugar de una dirección IP. Se pueden poner varias separadas por coma. Se puede usar ! para invertir la búsqueda.
`-d` *IP/máscara*	Dirección IP o nombre de host del destino.
`-m` *parámetros*	Con `-m` o `--match` podemos dar condiciones más detalladas del paquete. Si las cumple, se ejecutará la acción en la opción `-j`. Más abajo se dará una lista con los parámetros disponibles.
`-j` *target*	Especifica qué hacer cuando el paquete cumple todas las condiciones establecidas en la regla. Los targets básicos son REJECT, DROP, ACCEPT y LOG. Algunos aceptan opciones extra. También se puede poner el nombre de una cadena personalizada, con lo que se pasará a comparar el paquete con todas las reglas de esa cadena.

`-i` *interfaz*	Nombre de la inerfaz desde la que se ha recibido un paquete. Slo disponible en las cadenas INPUT, FORWARD y PREROUTING. Se puede usar ! al principio para revertir la búsqueda, o + al final como carácter comodín.
`-o` *interfaz*	Nombre de la interfaz por la que el paquete va a ser enviado. Solo disponible en las cadenas FORWARD, OUTPUT y POSTROUTING. Se puede usar ! al principio para revertir la búsqueda, o + al final como carácter comodín.

Con la opción `-m` debemos usar lo que se llaman **módulos** de filtrado. Cada módulo tiene un nombre y acepta diferentes opciones, que se pueden escribir en la misma línea. Algunos de ellos son:

Módulo	**Opciones**	**Descripción**
`state`	`--state` *estado*	Comprueba el estado de conexión de ese paquete. El estado puede ser INVALID (no hay ninguna conexión conocida asociada a ese paquete), NEW (el paquete ha empezado una nueva conexión) y ESTABLISHED (el paquete está asociado a una conexión conocida, que ya ha enviado paquetes en ambos sentidos), entre otros.
`limit`	`--limit` *núm/ud_tiempo* `--limit-burst` *número*	La regla en la que está `limit` se comprobará como máximo una vez en el intervalo de tiempo que le digamos. `/s, /m, /h, /d` para segundos, minutos, horas y días. `--limit-burst` marca el número de veces que se permiten antes de que el límite sea efectivo. Después de una regla con `limit` se suele usar una con DROP, para descartar el resto de paquetes.
`tcp`	`--sport` *puerto[:puerto]* `--dport` *puerto[:puerto]* `--tcp-flags` *flags flags* `--syn`	Se pueden usar estas opciones directamente detrás de `-p tcp` en una regla. Permite filtrar un paquete según su puerto de origen y destino (se pueden usar rangos), y según su tipo, dependiendo de las flags TCP que tenga. El primer parámetro en `--tcp-flags` es para las flags que deben ser examinadas y el segundo para las que deben estar activadas, es decir las que estén en el primer parámetro y no en el segundo deberán estar sin activar. `--syn` es equivalente a `--tcp-flags SYN,RST ,ACK,FIN SYN`

Veamos algunos ejemplos sencillos de reglas:

```
iptables -A INPUT -s 192.168.0.0/24 -j DROP
```

Descarta los paquetes que vengan desde la red 192.168.0.0/24 por cualquier interfaz. La diferencia entre los targets DROP y REJECT es que este último envía de vuelta un mensaje ICMP de destino inalcanzable (podemos cambiar este mensaje con la opción `--reject-with`), mientras que DROP lo bloquea sin más.

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.0.0/24 --dport 22 -m state NEW,
    ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j
    ACCEPT
```

Permite todo el tráfico SSH que provenga de la red 192.168.0.0/24, siempre que haya iniciado la conexión otro host (esto último depende de la política por defecto que hayamos puesto en OUTPUT, suponemos que es DROP).

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j
    ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

Estas son las reglas necesarias para hacer traducción de direcciones. eth0 sería la interfaz externa y eth1 la interna. La primera regla es para que nuestra máquina enmascare el tráfico que sale hacia el exterior, es decir, que haga la traducción de direcciones para que parezca que el paquete ha sido enviado por ella. La segunda es para permitir todo el tráfico hacia las máquinas de la red interna, siempre que hayamos iniciado nosotros la comunicación. Está en la cadena FORWARD de la tabla *filter*, con lo cual en el momento de comprobar esta regla ya se ha hecho el enrutamiento y la dirección de destino será una de nuestra red interna. La tercera regla sirve para permitir todo el tráfico desde nuestra red interna al exterior.

5 | El firewall de SME Server

El firewall de SME server es modificado automáticamente en respuesta a los cambios que hacemos en la configuración, como activar o desactivar servicios, hacerlos públicos o privados o reenviar puertos.

Por ejemplo, si añadimos una regla de reenvío de puertos en la interfaz web:

Podemos ver cómo la regla se ha añadido a la tabla *nat*.

```
[root@sme ~]# iptables -t nat -L PREROUTING
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
PortForwarding  all  --  anywhere             anywhere
SMTPProxy  tcp  --  anywhere             anywhere            tcp dpt:smtp
TransProxy  tcp  --  anywhere             anywhere            tcp dpt:http
[root@sme ~]# iptables -t nat -L PortForwarding
Chain PortForwarding (1 references)
target     prot opt source               destination
PortForwarding_3233  all  --  anywhere            10.0.2.15
[root@sme ~]# iptables -t nat -L PortForwarding_3233
Chain PortForwarding_3233 (1 references)
target     prot opt source               destination
DNAT       tcp  --  anywhere             anywhere            tcp dpt:7777 to:192.168.0.
7:80
```

Primero consultamos la cadena PREROUTING de la tabla *nat*, que es donde se tienen que añadir las reglas de Iptables para traducción de direcciones. Vemos que hay un target de nombre PortForwarding. Al no ser un target estándar de Iptables, se refiere a una cadena personalizada que el servidor ha creado. Como no tiene filtro, todos los paquetes serán pasados por esa cadena. La consultamos y vemos que a todos los paquetes que entren con IP destino 10.0.2.15 (la externa del servidor), les aplicará el target PortForwarding_3233, que es otra cadena personalizada. Finalmente consultamos esta y vemos que está la regla que hace DNAT (en este caso se refiere a Destination NAT, cambio de la dirección de destino).

Hay dos formas en las que podemos hacer una configuración más avanzada del firewall: modificando ciertos valores en la base de datos para los distintos servicios o creando templates propias para cambiar directamente la configuración de Iptables.

5.1 Modificación del firewall

El archivo responsable de cargar toda la configuración de Iptables en el sistema es `/etc/rc.d/init.d/masq`, por tanto las plantillas se encuentran en `/etc/e-smith/templates /etc/rc.d/init.d/masq`.

```
[root@sme ~]# ls /etc/e-smith/templates/etc/rc.d/init.d/masq/
00Definitions       42CreateSSHAutoblock        90InboundTCP07FilterSSH
00Functions         42SetupPortForwarding       90InboundTCP10filter_tcp
00start             45AllowDHCPS                90InboundTCP50adjust_tcp
01localNetworks     55AllowGRE                  90InboundTCP99Finish
10flush             85PolicyForward             90InboundUDP00Start
10masq_ftp          85PolicyInput               90InboundUDP10filter_udp
10RemoveUserChains  85PolicyOutput              90InboundUDP50adjust_udp
20NewChainDenyLog   86startdone                 90InboundUDP99Finish
30AdjustTOS         89adjustStart               90local_chk00Start
30SetMasqTimeouts   90adjustDenyLog             90local_chk50networks
35SMTPProxy         90adjustGRE                 90local_chk99Finish
35transproxy        90adjustICMPIn              90PPPconnAdjust00Start
40AllowEstablished  90adjustMasq                90PPPconnAdjust50adjust_conns
40AllowGSMTP        90adjustSMTPProxy           90PPPconnAdjust99Finish
40AllowLocal        90adjustSSHAutoblock        91adjustPortForward
40AllowPPP          90adjustTransProxy          92adjustEnd
40DenyMulticast     90ForwardedTCP00Start       98MasqStart
40masqLAN           90ForwardedTCP50adjust_tcp  98MasqStop
41AllowDHCPC        90ForwardedTCP99Finish      98restart
42CheckICMPIn       90ForwardedUDP00Start       98status
42CheckTCPForward   90ForwardedUDP50adjust_udp  98stop
42CheckTCPInput     90ForwardedUDP99Finish      98trace
42CheckUDPForward   90InboundTCP00Start         98usage
42CheckUDPInput     90InboundTCP05RejectIDENT   template-begin
```

Tendremos que copiar y modificar la plantilla adecuada en `templates-custom`, o crear una nueva, y luego expandirla con `expand-template` y reiniciar el servicio: `/etc/init.d/masq restart`.

Vamos a configurar el firewall del servidor para proporcionar una seguridad más fuerte en el caso de algunos ataques comunes:

5.1.1 Denegación de servicio

Un ataque de denegación de servicio (denial-of-service, DoS) causa que un servicio o recurso sea inaccesible a sus usuarios. Se suelen generar mediante la saturación de los puertos con flujo de información, consumiendo todo el ancho de banda y bloqueando así el servidor. Existen diferentes tipos de ataques de denegación de servicio por lo que hay que elaborar una respuesta para cada uno de ellos.

- **Ping flood**

Consiste en saturar al servidor con paquetes ICMP, enviándolos lo más rápido posible sin esperar respuesta. Esto provoca que se consuma mucho ancho de banda de entrada y también de salida, puesto que el servidor intentará responder. Es más efectivo si el ancho de banda disponible para el atacante es mayor que el del servidor.

Para comprobar si SME server está configurado contra el ping flood por defecto, le he cambiado el adaptador en VirtualBox a modo bridge, para que la interfaz externa esté en la misma red que mi ordenador (la nueva IP es 192.168.1.250), y le he atacado desde ahí. Todos los pings son satisfactorios con lo que SME no está protegido contra esto.

```
david@latitude:~$ sudo ping -f 192.168.1.250
PING 192.168.1.250 (192.168.1.250) 56(84) bytes of data.
.^C
--- 192.168.1.250 ping statistics ---
44208 packets transmitted, 44208 received, 0% packet loss, time 5680ms
rtt min/avg/max/mdev = 0.039/0.103/6.771/0.134 ms, ipg/ewma 0.128/0.093 ms
```

Una solución para evitar esto es permitir solamente un ping por segundo. Podemos escribir una nueva cadena para controlar esto y llamarla desde INPUT cuando se encuentren paquetes ICMP.

```
/sbin/iptables -N icmpFlood
/sbin/iptables -A icmpFlood -p icmp --icmp-type echo-request -m limit --limit 1/
    s --limit-burst 1 -j ACCEPT
/sbin/iptables -A icmpFlood -j DROP

/sbin/iptables -A INPUT -p icmp -j icmpFlood
```

En este caso usaré la plantilla 39misReglas, para que la regla que está en la cadena INPUT se añada al principio. Las cadenas en la tabla *filter* quedan así:

```
[root@sme masq]# iptables -L INPUT
Chain INPUT (policy DROP)
target     prot opt source               destination
icmpFlood  icmp --  anywhere             anywhere
state_chk  all  --  anywhere             anywhere
local_chk  all  --  anywhere             anywhere
PPPconn    all  --  anywhere             anywhere
denylog    all  --  base-address.mcast.net/4  anywhere
denylog    all  --  anywhere             base-address.mcast.net/4
InboundTCP  tcp  --  anywhere             anywhere            tcp flags:FIN,SYN,RST,ACK
/SYN
denylog    tcp  --  anywhere             anywhere            tcp flags:FIN,SYN,RST,ACK/
SYN
InboundUDP  udp  --  anywhere             anywhere
denylog    udp  --  anywhere             anywhere
ACCEPT     udp  --  anywhere             anywhere            udp spts:bootps:bootpc
gre-in     gre  --  anywhere             anywhere
denylog    gre  --  anywhere             anywhere
denylog    all  --  anywhere             anywhere
[root@sme masq]# iptables -L icmpFlood
Chain icmpFlood (1 references)
target     prot opt source               destination
ACCEPT     icmp --  anywhere             anywhere            icmp echo-request limit: a
vg 1/sec burst 1
DROP       icmp --  anywhere             anywhere
```

Con estas reglas se soluciona el problema. Este es el resultado al volver a intentar el ping flood desde nuestro ordenador:

```
david@latitude:~$ sudo ping -f 192.168.1.250
PING 192.168.1.250 (192.168.1.250) 56(84) bytes of data.
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
..^C
--- 192.168.1.250 ping statistics ---
397 packets transmitted, 5 received, 98% packet loss, time 4828ms
rtt min/avg/max/mdev = 0.263/0.315/0.381/0.047 ms, ipg/ewma 12.193/0.347 ms
```

- **SYN flood**

Cuando queremos iniciar una conexión TCP, se produce el saludo a tres vías: el cliente envía un mensaje con la bandera SYN, el servidor responde con un mensaje SYN-ACK y finalmente el cliente responde con un mensaje ACK. El ataque SYN flood consiste en saturar el servidor mandando mensajes SYN y no respondiendo con los correspondientes ACK.

En Linux podemos usar el programa hping3 con este fin.

```
david@latitude:~$ sudo hping3 -q -i u50 -S -p 80 192.168.1.250
HPING 192.168.1.250 (wlp3s0 192.168.1.250): S set, 40 headers + 0 data bytes
^C
--- 192.168.1.250 hping statistic ---
117560 packets transmitted, 88536 packets received, 25% packet loss
round-trip min/avg/max = 1.5/17.9/32.9 ms
```

Con el comando de la imagen enviamos un mensaje con la bandera SYN al puerto 80 cada 50 microsegundos. Como resultado, mientras se está ejecutando este comando el servidor está bloqueado, impidiéndonos por ejemplo acceder a la interfaz web de administración.

Para solucionarlo de una forma distinta al anterior, se puede utilizar el módulo `recent`:

```
modprobe ipt_recent

/sbin/iptables -N synFlood
/sbin/iptables -A synFlood -m recent --set
/sbin/iptables -A synFlood -m recent --update --seconds 2 --hitcount 20 -j DROP

/sbin/iptables -A INPUT -p tcp --tcp-flags SYN SYN -j synFlood
```

La primera línea es para cargar el módulo. La última es para que los paquetes que tengan al menos la bandera SYN pasen por nuestra cadena synFlood. La primera regla de la nueva cadena añade la IP de origen del paquete a la lista. Si ya está en la lista, actualiza la entrada. La siguiente línea se encarga de actualizar el tiempo en el que se ha visto esa IP y rechazar los paquetes si la IP ha sido vista en la tabla en los últimos dos segundos y si tiene al menos 20 paquetes enviados. La lista que guarda los paquetes enviados no tiene en cuenta el tiempo en el que han llegado. Esto quiere decir que el control solo se reiniciará cuando la IP de origen haya estado 2 segundos sin enviar ningún paquete, o al

menos ninún paquete que se haya comparado contra esta regla. Como podemos ver, ahora solo llegan 19 paquetes de vuelta:

```
david@latitude:~$ sudo hping3 -q -i u50 -S -p 80 192.168.1.250
HPING 192.168.1.250 (wlp3s0 192.168.1.250): S set, 40 headers + 0 data bytes
^C
--- 192.168.1.250 hping statistic ---
95881 packets transmitted, 19 packets received, 100% packet loss
round-trip min/avg/max = 2.4/3.0/3.7 ms
```

5.1.2 Suplantación de identidad

Este tipo de ataque consiste en que un host o aplicación se hace pasar por otro. Por ejemplo, nos pueden llegar paquetes desde la interfaz externa con IPs de nuestra red interna. En estos casos el atacante no se preocupa de la respuesta que origina el paquete ya que no le llegará.

Podemos lanzar un ataque SYN flood de manera que cada paquete que enviemos se envíe con una dirección IP de origen aleatoria, con lo que la regla que hemos creado contra estos ataques no serviría para nada. El programa hping3 permite hacerlo de manera sencilla, sin más que añadir la opción `--rand-source`:

```
david@latitude:~$ sudo hping3 --flood -S -p 80 --rand-source 192.168.1.250
HPING 192.168.1.250 (wlp3s0 192.168.1.250): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.250 hping statistic ---
9244 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Estamos provocando que nuestro servidor mande mensajes SYN-ACK a esas IPs aleatorias.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	91.176.31.238	192.168.1.250	TCP	54	2561→80 [SYN] Seq=0 Win=512 Len=0
2	0.000259000	192.168.1.250	91.176.31.238	TCP	60	80→2561 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
3	0.100329000	149.92.82.224	192.168.1.250	TCP	54	2562→80 [SYN] Seq=0 Win=512 Len=0
4	0.100798000	192.168.1.250	149.92.82.224	TCP	60	80→2562 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
5	0.200424000	192.168.1.250	187.53.61.3	TCP	60	80→2552 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
6	0.200448000	39.115.138.196	192.168.1.250	TCP	54	2563→80 [SYN] Seq=0 Win=512 Len=0
7	0.200612000	192.168.1.250	203.129.227.53	TCP	60	80→2551 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
8	0.200781000	192.168.1.250	39.115.138.196	TCP	60	80→2563 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
9	0.300544000	205.11.144.138	192.168.1.250	TCP	54	2564→80 [SYN] Seq=0 Win=512 Len=0
10	0.300903000	192.168.1.250	205.11.144.138	TCP	60	80→2564 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
11	0.400730000	134.187.230.227	192.168.1.250	TCP	54	2565→80 [SYN] Seq=0 Win=512 Len=0
12	0.401108000	192.168.1.250	134.187.230.227	TCP	60	80→2565 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
13	0.401293000	192.168.1.250	91.109.92.149	TCP	60	80→2550 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
14	0.401383000	192.168.1.250	220.33.77.58	TCP	60	80→2555 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
15	0.401420000	192.168.1.250	153.27.104.227	TCP	60	80→2554 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
16	0.500844000	130.11.115.246	192.168.1.250	TCP	54	2566→80 [SYN] Seq=0 Win=512 Len=0
17	0.501222000	192.168.1.250	130.11.115.246	TCP	60	80→2566 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
18	0.600954000	161.155.174.101	192.168.1.250	TCP	54	2567→80 [SYN] Seq=0 Win=512 Len=0
19	0.601324000	192.168.1.250	89.199.212.229	TCP	60	80→2557 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
20	0.601458000	192.168.1.250	161.155.174.101	TCP	60	80→2567 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
21	0.701058000	129.241.112.143	192.168.1.250	TCP	54	2568→80 [SYN] Seq=0 Win=512 Len=0
22	0.701313000	192.168.1.250	129.241.112.143	TCP	60	80→2568 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460

Esto puede ser evitado con una regla de las del primer tipo, la que vimos para el ping flood.

```
/sbin/iptables -N synFlood2
/sbin/iptables -A synFlood2 -m limit --limit 1/s --limit-burst 1 -j RETURN
/sbin/iptables -A synFlood2 -j DROP

/sbin/iptables -A INPUT -p tcp --tcp-flags SYN SYN -j synFlood2

/sbin/iptables -A INPUT -s 10.0.0.0/8 -i eth0 -j DROP
/sbin/iptables -A INPUT -s 172.16.0.0/12 -i eth0 -j DROP
/sbin/iptables -A INPUT -s 192.168.0.0/16 -i eth0 -j DROP
/sbin/iptables -A INPUT -s 224.0.0.0/4 -i eth0 -j DROP
/sbin/iptables -A INPUT -s 240.0.0.0/5 -i eth0 -j DROP
/sbin/iptables -A INPUT -s 127.0.0.0/8 -i eth0 -j DROP
/sbin/iptables -A FORWARD -s 10.0.0.0/8 -i eth0 -j DROP
/sbin/iptables -A FORWARD -s 172.16.0.0/12 -i eth0 -j DROP
/sbin/iptables -A FORWARD -s 192.168.0.0/16 -i eth0 -j DROP
/sbin/iptables -A FORWARD -s 224.0.0.0/4 -i eth0 -j DROP
/sbin/iptables -A FORWARD -s 240.0.0.0/5 -i eth0 -j DROP
/sbin/iptables -A FORWARD -s 127.0.0.0/8 -i eth0 -j DROP
```

También hemos añadido varias reglas para rechazar paquetes que vengan desde la interfaz eth0, que es la de la red externa, con dirección IP de origen de una red privada, de multicast o de loopback. Podemos comprobar que las reglas contra el SYN flood son efectivas analizando el tráfico y viendo que solo hay una respuesta SYN-ACK del servidor cada segundo.

```
    43 2.803477000 24.196.207.124   192.168.1.250   TCP   54 2795→80 [SYN] Seq=0 Win=512 Len=0
    44 2.903650000 165.187.228.202  192.168.1.250   TCP   54 2796→80 [SYN] Seq=0 Win=512 Len=0
    45 3.003747000 182.234.149.98   192.168.1.250   TCP   54 2797→80 [SYN] Seq=0 Win=512 Len=0
    46 3.103861000 132.114.166.42   192.168.1.250   TCP   54 2798→80 [SYN] Seq=0 Win=512 Len=0
    47 3.203966000 196.145.8.107    192.168.1.250   TCP   54 2799→80 [SYN] Seq=0 Win=512 Len=0
    48 3.304530000 92.149.140.36    192.168.1.250   TCP   54 2800→80 [SYN] Seq=0 Win=512 Len=0
    49 3.404647000 224.120.110.31   192.168.1.250   TCP   54 2801→80 [SYN] Seq=0 Win=512 Len=0
    50 3.504819000 60.98.12.97      192.168.1.250   TCP   54 2802→80 [SYN] Seq=0 Win=512 Len=0
    51 3.505130000 192.168.1.250    60.98.12.97     TCP   60 80→2802 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
    52 3.605017000 0.54.187.71      192.168.1.250   TCP   54 2803→80 [SYN] Seq=0 Win=512 Len=0
    53 3.705187000 33.184.183.33    192.168.1.250   TCP   54 2804→80 [SYN] Seq=0 Win=512 Len=0
    54 3.805307000 149.202.162.27   192.168.1.250   TCP   54 2805→80 [SYN] Seq=0 Win=512 Len=0
```

Bibliografía

- **Manuales de SME Server**
  https://wiki.contribs.org/SME_Server:Documentation:Administration_Manual
  https://wiki.contribs.org/SME_Server:Documentation:Developers_Manual
  https://wiki.contribs.org/Template_Tutorial

- **Manuales de Iptables**
  http://www.tony-hill.info/app/download/1367500/IPTABLES+Tutorial+V1-3.pdf
  https://www.frozentux.net/iptables-tutorial/iptables-tutorial.html
  http://es.tldp.org/Manuales-LuCAS/doc-iptables-firewall/doc-iptables-firewall.pdf

- **Otros recursos y ejemplos**
  http://es.tldp.org/Manuales-LuCAS/GARL2/garl2/index.html
  https://help.ubuntu.com/community/IptablesHowTo
  http://www.cyberciti.biz/tips/linux-iptables-examples.html
  http://www.thegeekstuff.com/2011/01/iptables-fundamentals/
  http://blog.desdelinux.net/ddos-y-otros-ataques-vs-iptables-seguridad-anti-ddos-en-iptables/