Apache - Servidor web del SeRMN

Descripció general

El servidor web està allotjat a l'ordinador `sermnserver.sermn.net` de la xarxa local (LAN) del SeRMN. Alhora, el servidor web té configurats dos servidors virtuals bassats en l'adreça IP: `sermnserver.sermn.net` i `mruiserver.sermn.net` que contenen les pàgines web i serveis d'Internet relacionats respectivament amb el SeRMN i amb el programa MRUI.

L'accés a aquests dos servidors virtuals es fa a través d'un encaminador/tallafocs (Router/Firewall) que té assignades dues adreces IP a la interfície de WAN, on cadascuna de les IP de la WAN es correspon amb una de les IP de la LAN,

WAN Hostname / IP address	LAN Hostname / IP address
sermn.uab.cat / 158.109.58.175	sermnserver.sermn.net / 192.168.1.2
oldmrui.uab.cat / 158.109.58.236	mruiserver.sermn.net / 192.168.1.129

Al tallafocs s'han definit filtres que permenten el tràfic entre les adreces equivalents de la taula anterior,

Configuració de la xarxa

Aquesta és la configuració actual del servidor,

Programes de configuració (semi)automàtica

La configuració del servidor és estàtica, no canvia habitualment per tal d'evitar interferir amb les connexions, o sigui que no volem cap dels programes que *ajuden* a configurar la xarxa. Específicament, no han d'instal·lar-se (o s'han d'eliminar si estan instal·lats) els següents paquets de Debian:

- resolvconf
- network-manager

Fitxer: /etc/udev/rules.d/70-persistent-net.rules

Comentar l'entrada corresponent a la segona targeta ethernet perquè no la intenti configurar cap programa. El fitxer original es guarda amb el nom `70-persistent-net.rules.BKUP`.

```
# This file was automatically generated by the /lib/udev/write_net_rules
# program run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single line.
```

```
# PCI device 0x14e4:0x165a (tg3)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:25:b3:b8:73:a3", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth0"

# PCI device 0x10ec:0x8169 (r8169)
# SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="00:17:3f:d0:d6:5b", ATTR{type}=="1", KERNEL=="eth*",
NAME="eth1"
```

Fitxer: /etc/network/interfaces

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

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface

# allow-hotplug eth0
# replace allow-hotplug eth0 with auto eth0; otherwise restarting
# the network doesn't work, and we'd have to reboot the whole system

auto eth0 eth0:1

iface eth0 inet static
    address 192.168.2.2
    netmask 255.255.255.0
    network 192.168.2.0
    broadcast 192.168.2.255
    gateway 192.168.2.1
    ## dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 192.168.2.1
    dns-search sermn.net

# Another IP address at the same NIC. Intended for the MRUI server.
iface eth0:1 inet static
    address 192.168.2.129
    netmask 255.255.255.0
    network 192.168.2.0
    # broadcast 192.168.2.255
    # gateway 192.168.2.1
    ## dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 192.168.2.1
    dns-search sermn.net
```

Configuració de les interfícies de xarxa

Aquesta és la configuració de les interfícies de xarxa que obtenim amb els fitxers de configuració de més amunt,

```
# /bin/netstat -ie
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 00:25:b3:b8:73:a3
          inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::225:b3ff:feb8:73a3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11208 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18909 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1985783 (1.8 MiB)  TX bytes:2870754 (2.7 MiB)
          Interrupt:17

eth0:1    Link encap:Ethernet  HWaddr 00:25:b3:b8:73:a3
          inet addr:192.168.2.129  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:17

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1826 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1826 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:857452 (837.3 KiB)  TX bytes:857452 (837.3 KiB)
```

Taula de rutes

Aquesta és la taula de rutes que obtenim amb els fitxers de configuració de més amunt,

```
# /bin/netstat -arn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt
Iface
192.168.2.0     0.0.0.0         255.255.255.0   U         0 0          0
eth0
0.0.0.0         192.168.2.1     0.0.0.0         UG        0 0          0
eth0
```

Actualitzant a Apache 2

- Upgrading to Apache 2

Apache i els hostes virtuals

Configuració

La configuració general de l'apache es troba al fitxer `apache2.conf`.

La configuració adicional per a determinades aplicacions que depenen de l'apache es troba als diferents fitxers `.conf` de la carpeta `/etc/apache2/conf.d/`

La configuració dels diferents virtual hosts es troba a la carpeta `/etc/apache2/sites-available/` on hi ha un fitxer de configuració per cada virtual hosts definit. Per a que estiguin operatius, s'han d'habilitar amb la comanda `a2ensite <file>`, la qual crea un enllaç a la carpeta `/etc/apache2/sites-enabled/`.

Enllaços

- Creating Name and IP Based Virtual hosts in debian, article a DebianHelp.
- Configuración de Apache, lliçó del Curso Administración de Servidores Linux.
- Apache 2 configuration of clean URLs on Debian, al lloc web de Drupal.

Apache+SSL i els hosts virtuals

Aconseguir tenir més d'un servidor virtual basat en nom sota HTTPS (HTTP + SSL) no és una tasca fàcil. El problema esdevé perquè només hi pot haver un certificat SSL per adreça IP, i aquest certificat només pot correspondre a un dels noms associats a l'adreça IP. De forma que, tan bon punt intentem servir més d'un hoste virtual amb el mateix certificat, el client comença a tenir problemes de validesa del certificat.

Aquest són alguns dels enllaços que he trobat sobre aquest tema,

- A *Debian-Administration* hi ha un article sobre Setting up an SSL server with Apache2. És especialment interessant la discussió al final de l'article, on hi ha recomanacions que ens podrien servir, com per exemple, el comentari #63

Tanmateix, sembla ser que hi ha formes *d'enganyar* el servidor,

- Hosting multiple SSL vhosts on a single IP/Port/Certificate with Apache2.
- HOWTO: Apache 2 SSL Name-Based Virtual Hosting

si bé em sembla que aquesta és la més elegant,

- Shared IP, multiple vhosts and multiple SSL certificates on Apache

Alternativament, com que el nostre servidor estarà en una xarxa privada on no tindrem restriccions en el número d'adreces IP que li podem assignar, una solució més “canònica” seria fer servir IP diferents per cadascun dels SSL Virtual Hosts allotjats. Llavors s'hauria de configurar el router perquè

dirigís el tràfic a l'adreça IP correcta en funció de l'adreça web.

Hostes virtuals basats en l'adreça IP

Ateses les dificultats per fer compatible el protocol HTTPS amb els hostes virtuals basats en el nom, vàrem decidir-nos per configurar els hostes virtuals en base a l'adreça IP, seguint l'exemple *Setup 2* descrit a http://httpd.apache.org/docs/1.3/vhosts/examples.html,

```
...
Port 80
ServerName sermnserver.sermn.net

<VirtualHost 192.168.1.2>
  DocumentRoot /var/www/sermn
  ServerName sermn.uab.cat
  ...
</VirtualHost>

<VirtualHost 192.168.1.129>
  DocumentRoot /var/www/mrui
  ServerName mrui.uab.cat
  ...
</VirtualHost>
```

També vàrem definir un hoste per defecte a on s'enviaran totes les connexions que no vagin adreçades a un hoste i port vàlids,

```
...
<VirtualHost _default_:*>
  DocumentRoot /www/default
  ...
</VirtualHost>
```

això impedeix de forma efectiva que cap petició arribi al servidor principal.

Redirigint HTTP cap a HTTPS

- Apache: Redirect http to https Apache secure connection – force HTTPS Connections
- Redirect To SSL Using Apache's .htaccess
    - FORCE HTTPS AND NO DOUBLE LOGIN
    - Redirecting All or Part of a Server to SSL
    - Apache SSL in htaccess examples
- Use Apache's mod_rewrite to make URLs more user friendly
- Using Apache's RewriteEngine to redirect requests to other URLS and to HTTPS
- phpMyAdmin Web Administrator
- How To: 5 Steps to Install phpMyAdmin on Linux

Securing Apache

- HowTo: Secure your Ubuntu Apache Web Server

Antic servidor web a rmn3

- Transferint fitxers de rmn3 a sermn