

CURSO 2016-2017
PRIMER SEMESTRE



INFORMES PRÁCTICOS

GESTIÓN Y ADMINISTRACIÓN DE REDES

HÉCTOR MARTÍNEZ TOBAR - 1391872

GRUPO 11
DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y SISTEMAS OPERATIVOS

Contenido del documento

0. Introducción.....	4
1. Primer Informe – Preparación del entorno	4
1.1 Introducción	4
1.2 Preparación de las máquinas virtuales con VBOX.....	4
1.2.3 Creación del master	4
1.2.4 Clonaje de las máquinas virtuales	5
1.3.4 Configuración de las redes internas.....	6
1.3 Usuarios y grupos del sistema.....	7
1.3.1 Creación grupo	7
1.4 Recursos del sistema	8
1.4.1 Arquitectura	8
1.4.2 CPU	8
1.4.3 Memoria principal	9
1.5 Hardware.....	9
1.5.1 Particiones y FS.....	9
1.5.2 Recursos disponibles	10
1.6 Servicios en ejecución	10
1.7 Niveles de ejecución.....	12
1.7.1 Servicios asociados	12
1.7.2 Cambiando el runlevel.....	16
1.8 Entorno de trabajo	17
1.8.1 Con GUI sin X-Window	17
1.8.2 Sin GUI con X-Windows	17
1.9 BootManager	17
1.10 Sistema de paquetes.....	18
1.10.1 Paquetes instalados en el sistema	18
1.10.2 Verificar actualizaciones	18
1.11 Kernel.....	19
1.11.1 El kernel de nuestro entorno virtual	19
1.11.2 Operar con distintos kernels.....	19
1.11.3 Módulos activos	19
1.12 Red	19
1.12.1 Dispositivos y redes activos en nuestro entorno	19
1.13 Configuración de la red.....	20

1.13.1	Configuración del master	21
1.13.2	Configuración slave1	22
1.13.3	Configuración slave2	23
1.14	Verificaciones de funcionamiento	23
1.14.1	Ping de master a ambos slaves	23
1.14.2	Ping de master a Internet	24
1.14.3	Ping de slave1 a master y slave2.....	24
1.14.4	Ping de slave1 a Internet.....	25
1.14.5	Ping de slave2 a master y a slave1	25
1.14.6	Ping de slave 2 a Internet.....	25
2.	Segundo Informe – SSH, DHCP y DNS	26
2.1	OpenSSH Server	26
2.1.1	Instalación	26
2.1.2	Pruebas de funcionamiento básico	26
2.1.3	Configuración de acceso mediante pubkey	27
2.1.4	Pruebas de funcionamiento avanzado (con pubkey)	28
2.1.5	Mejoras acceso vía SSH	30
2.2	DHCP.....	30
2.2.1	Instalación	30
2.2.2	Pasos previos.....	31
2.2.3	Configuración	32
2.2.4	Asignación IP a Slave2. Problemas, isc-dhcp-relay y config de interfaces	32
2.2.5	Asignación estática para Slave1 y Slave2	32
2.3	DNS.....	32
2.3.1	Instalación	32
2.3.2	Configuración	32
3.	Tercer Informe – Servicios	32
3.1	Introducción	32
3.2	NFS	32
3.2.1	Instalación	32
3.2.2	Configuración	33
3.3	Apache.....	35
3.3.1	Instalación	35
3.3.2	Edición página por defecto.....	36
3.3.3	Logs de Apache.....	37

3.3.4	Módulos.....	37
3.3.5	VirtualHosts por IP.....	39
3.3.6	Certificados y HTTPS.....	41
4.	Cuarto Informe – Seguridad	44
4.1	Cuestiones previas	44
4.2	Ejercicio 1	45
4.3	Ejercicio 2	47
4.4	Ejercicio 3	51
4.5	Ejercicio 4	56
5.	Quinto Informe – Monitorización.....	60
5.1	Introducción	60
5.2	Nagios.....	60
5.2.1	Instalación sobre master	60
5.2.2	Instalación sobre los clientes.....	71
5.2.3	Pruebas de monitorización.....	78
5.3	Ganglia.....	83
5.3.1	Instalación sobre master y configuración del servidor	83
5.3.2	Configuración clientes	84
5.4	Comparativa y conclusiones.....	87

0. Introducció

Durante el desarrollo de esta práctica se pretende preparar el entorno de virtual con el que se irá trabajando a lo de los distintos laboratorios.

El entorno en cuestión consta de tres máquinas virtuales corriendo bajo la distribución Debian de 64 bits. Una de éstas, la *Master*, cuenta con dos tipos de red: NAT, para poder salir a Internet y la red interna, que será con la que establezca comunicación con las otras dos máquinas denominadas *Slave1* y *Slave2*.

Además de la preparación del propio entorno, incluyendo la red, se irán mostrando una serie de explicaciones y justificaciones sobre la gestión de los usuarios del sistema, dispositivos y paquetes, copias de seguridad, kernel, run-levels...

1. Primer Informe – Preparación del entorno

1.1 Introducció

En este informe se trata toda la preparación del entorno la cual incluye la administración básica local de los sistemas UNIX/Linux, que corresponde a los niveles, gestión de usuarios y dispositivos, copias, kernel, etc y, también, a la gestión de la red.

1.2 Preparación de las máquinas virtuales con VBOX

Lo primero que debemos hacer es preparar las máquinas virtuales que usaremos durante todos los informes. Se elige el software VirtualBox.

1.2.3 Creación del *master*

Para crear la primera máquina virtual que nos servirá como plantilla de las demás, se debe clicar sobre el botón *Nueva* que vemos a continuación y seguir el asistente.




Il·lustració 1 - Botón creación máquina virtual


Seleccionamos el nombre y el sistema operativo que correrá la nueva creación.

Nombre y sistema operativo

Seleccione un nombre descriptivo para la nueva máquina virtual y seleccione el tipo de sistema operativo que tiene intención de instalar en ella. El nombre que seleccione será usado por VirtualBox para identificar esta máquina.

Nombre:

Tipo: 

Versión: 

Il·lustració 2 - Nombre y SO de la nueva máquina

Ahora se debe asignar la memoria RAM de la que dispondrá dicha máquina.

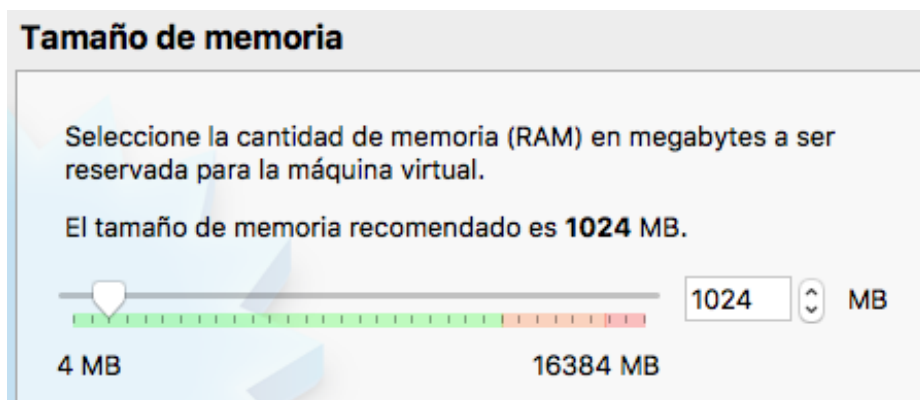


Ilustración 3 - RAM asignada a la máquina virtual

Por último, se procede a la asignación del disco y se crea la máquina virtual.

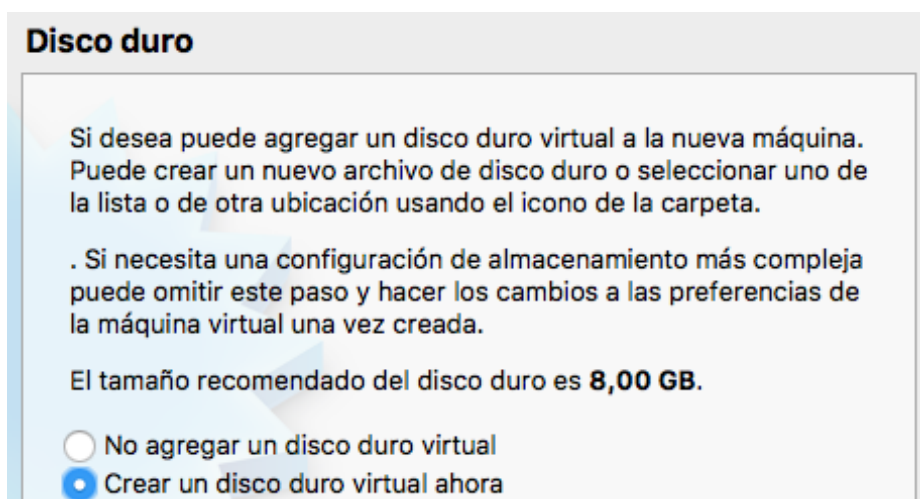


Ilustración 4 - Asignación de disco y creación

Una vez creada nos debe aparecer en el menú de la izquierda nuestra nueva máquina *Master*.

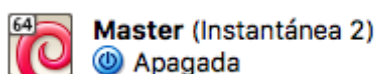


Ilustración 5 - Máster creada

1.2.4 Clonaje de las máquinas virtuales

El proceso de clonación es muy simple, botón derecho sobre la máquina que se desea clonar y seguir el asistente. Se recomienda reinicializar las direcciones MAC para que no tengamos ningún conflicto con la máquina original.

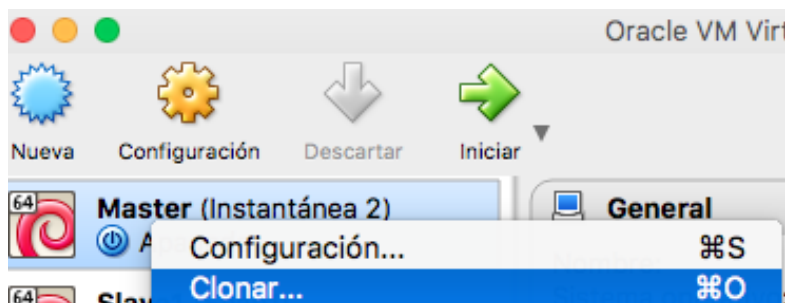


Ilustración 6 - Clonaje máquina máster

Una vez finalizado el proceso, junto a la máquina *master* debemos tener la nueva máquina resultado de la clonación de la anterior, en este caso, *slave1*. Repetimos el proceso una vez más y generamos *slave2*.

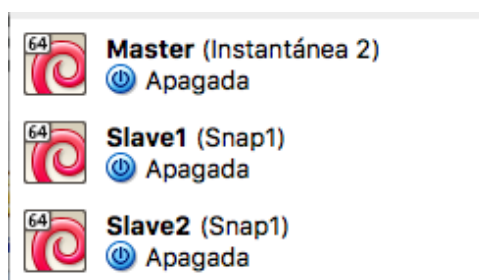


Ilustración 7 - Lista de máquinas virtuales

1.3.4 Configuración de las redes internas

Con el objetivo de conectar las máquinas entre ellas emulando una nueva interfaz de red, se usan las redes internas de VirtualBox.

Para poder realizar la configuración, se selecciona la máquina virtual en el panel izquierdo y pulsamos sobre el botón de configuración. Seguidamente seleccionamos la pestaña de red, habilitamos el adaptador y añadimos la red interna a la que debe pertenecer cada máquina virtual.

Se muestra el proceso de configuración del *Slave1*.



Ilustración 8 - Acceso a la configuración

Slave1 debe conectar con *Master* y con *Slave2*, éstas dos están en redes distintas, es por eso que se usan dos adaptadores distintos con la siguiente configuración.

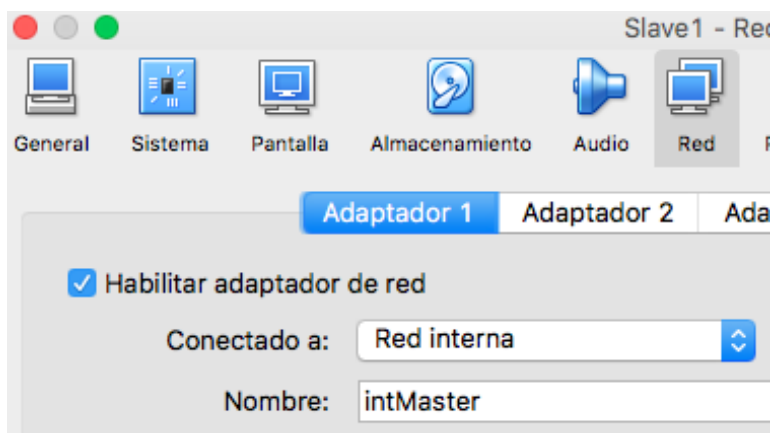


Ilustración 9 - Red interna intMaster

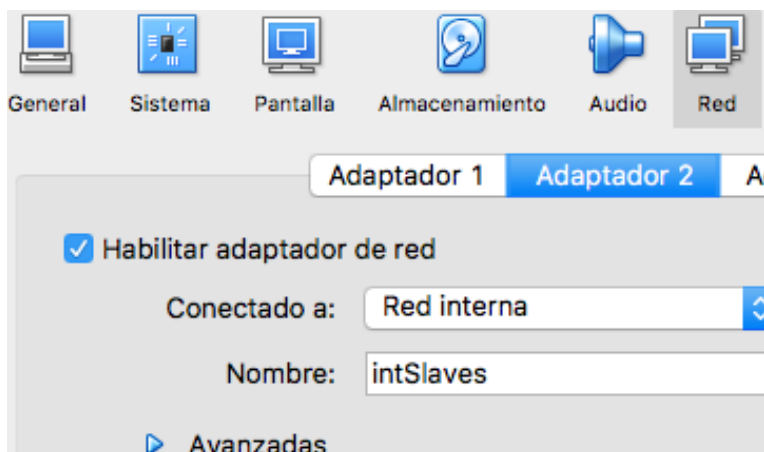


Ilustración 10 - Red interna intSlaves

1.3 Usuarios y grupos del sistema

Ahora que ya disponemos de las máquinas y de su configuración de red, podemos empezar a trabajar con ellas. Se procede a iniciarlas y a crear grupos y usuarios para poder operar con ellos.

1.3.1 Creación grupo

Para crear un grupo usamos el comando:

```
# groupadd <nombre_del_grupo>
```

```
root@master:~# groupadd MyFirstGroup4GAX
root@master:~# _
```

Ilustración 11 - Snippet creación grupo

Para verificar que se ha creado correctamente, buscamos el grupo recién creado en `/etc/group`

```
# cat /etc/group | grep MyFirst
```

```
root@master:~# cat /etc/group | grep MyFirst
MyFirstGroup4GAX:x:1003:
```

Ilustración 12 - Verificación existencia de grupo

1.4 Recursos del sistema

Procedemos a verificar las siguientes características de nuestra máquina.

1.4.1 Arquitectura

Para determinar la arquitectura nos serviremos del comando `uname` con el parámetro `m`. El parámetro `-m` nos da información sobre el hardware name de nuestra máquina.

```
# uname -m
```

```
root@master:~# uname -m
x86_64
```

Ilustración 13 - Snippet arquitectura máquina

1.4.2 CPU

Para determinar la CPU y toda su información relativa consultamos el fichero `/proc/cpuinfo` con el comando `less`, de este modo podemos navegar entre el mismo desde el terminal.

```
# less /proc/cpuinfo
```

```
root@master:~# less /proc/cpuinfo
```

Ilustración 14 - Snippet para conocer CPU

Cuyo output es el siguiente:

```
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 42
model name     : Intel(R) Core(TM) i5-2435M CPU @ 2.40GHz
stepping       : 7
cpu MHz        : 2394.542
cache size     : 3072 KB
physical id    : 0
siblings       : 1
core id        : 0
cpu cores      : 1
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 13
wp             : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 syscall nx rdtscp lm constant_tsc rep_good n
opl xtopology nonstop_tsc pni pclmulqdq monitor ssse3 cx16 sse4_1 sse4_2 popcnt
aes xsave avx  hypervisor lahf_lm
bogomips       : 4789.08
clflush size   : 64
```

Ilustración 15 - Output cpuinfo

1.4.3 Memoria principal

Para conocer la memoria de la que dispone nuestra máquina, usamos el comando *free* con los parámetros *m* y *h*. Dichos parámetros harán que la información se muestre en megas y que la información que sea impresa, pueda ser fácilmente entendida por el humano.

```
# free -mh
```

```
root@master:~# free -mh
              total        used        free      shared    buffers     cached
Mem:           1.0G         151M         849M          4.4M          22M          73M
-/+ buffers/cache:          55M         945M
Swap:          217M           0B         217M
```

Ilustración 16 - Snippet free

1.5 Hardware

Procedemos a verificar las siguientes características de nuestra máquina.

1.5.1 Particiones y FS

Para conocer las particiones de nuestro sistema usamos el comando *fdisk -l*.

```
# fdisk -l
```

```
root@master:~# fdisk -l

Disk /dev/sda: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x32cff0a3

Device     Boot    Start        End    Sectors    Size Id Type
/dev/sda1  *         2048    7938047    7936000    3.8G 83 Linux
/dev/sda2                7940094    8386559    446466     218M  5 Extended
/dev/sda5                7940096    8386559    446464     218M 82 Linux swap / Solaris
```

Ilustración 17 - Snippet fdisk

Para conocer los *filesystems* de nuestro sistema usamos el comando *df*.

```
# df
```

```
root@master:~# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1       3840152 1436916   2188452   40% /
udev             10240         0      10240    0% /dev
tmpfs           204880     4520    200360    3% /run
tmpfs           512196         0    512196    0% /dev/shm
tmpfs            5120         0       5120    0% /run/lock
tmpfs           512196         0    512196    0% /sys/fs/cgroup
tmpfs           102440         0    102440    0% /run/user/0
```

Ilustración 18 - Snippet df

1.5.2 Recursos disponibles

Si queremos conocer los recursos disponibles de nuestra máquina, usamos el comando *htop*.

```
# htop
```

```
CPU [ | 0.5%] Tasks: 20, 10 thr; 1 running
Mem [ | 59/1000MB] Load average: 0.16 0.08 0.06
Swp [ 0/217MB] Uptime: 00:29:54
```

Ilustración 19 - Snippet htop

Con este comando agrupamos los recursos de cpu y de memoria.

Con el comando *df -h* somos capaces de ver la capacidad disponible de nuestros filesystems.

```
# df -h
```

```
root@master:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        3.7G  1.4G  2.1G  40% /
udev             10M    0   10M   0% /dev
tmpfs            201M   4.5M  196M   3% /run
tmpfs            501M    0   501M   0% /dev/shm
tmpfs            5.0M    0   5.0M   0% /run/lock
tmpfs            501M    0   501M   0% /sys/fs/cgroup
tmpfs           101M    0   101M   0% /run/user/0
```

Ilustración 20 - Snippet df -h

1.6 Servicios en ejecución

Para conocer los servicios que están siendo ejecutados en la máquina, podemos usar el comando *htop*.

```
# htop
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1214	root	20	0	24244	3360	2840	R	0.0	0.3	0:00.03	htop
534	root	20	0	228M	1684	1392	S	0.0	0.2	0:00.35	/usr/sbin/VBoxSer
1	root	20	0	28568	4620	3108	S	0.0	0.5	0:00.82	/sbin/init
143	root	20	0	28872	2952	2664	S	0.0	0.3	0:00.06	/lib/systemd/syst
151	root	20	0	41464	3744	2764	S	0.0	0.4	0:00.17	/lib/systemd/syst
412	root	20	0	25400	7720	812	S	0.0	0.8	0:00.00	dhclient -v -pf /
480	root	20	0	37080	2692	2260	S	0.0	0.3	0:00.00	/sbin/rpcbind -w
489	statd	20	0	37280	2884	2300	S	0.0	0.3	0:00.00	/sbin/rpc.statd
504	root	20	0	23356	208	4	S	0.0	0.0	0:00.00	/usr/sbin/rpc.idm
505	daemon	20	0	19024	1764	1596	S	0.0	0.2	0:00.00	/usr/sbin/atd -f
506	root	20	0	27504	2732	2456	S	0.0	0.3	0:00.00	/usr/sbin/cron -f
507	root	20	0	55184	5244	4572	S	0.0	0.5	0:00.01	/usr/sbin/sshd -D
510	root	20	0	28268	2936	2596	S	0.0	0.3	0:00.01	/lib/systemd/syst
513	messagebu	20	0	42248	3412	3000	S	0.0	0.3	0:00.13	/usr/bin/dbus-dae
532	root	20	0	228M	1684	1392	S	0.0	0.2	0:00.00	/usr/sbin/VBoxSer
533	root	20	0	228M	1684	1392	S	0.0	0.2	0:00.03	/usr/sbin/VBoxSer
535	root	20	0	228M	1684	1392	S	0.0	0.2	0:00.00	/usr/sbin/VBoxSer
536	root	20	0	228M	1684	1392	S	0.0	0.2	0:00.02	/usr/sbin/VBoxSer

Ilustración 21 - Output comando htop

De la misma manera, podríamos usar también el comando *ps -aux*, pero htop nos muestra los datos de una manera más clara.

Hay varias maneras para identificar los servicios que se están ejecutando en el sistema. Usamos el nuevo método el cual corresponde a **systemd**.

Para ello nos servimos del comando `systemd-cgtop`

```
# systemd-cgtop
```

Path	Tasks	%CPU	Memory	Input/s	Output/s
/	65	4.3	-	-	-
/system.slice/acpid.service	1	-	-	-	-
/system.slice/atd.service	1	-	-	-	-
/system.slice/cron.service	1	-	-	-	-
/system.slice/dbus.service	1	-	-	-	-
/system.slice/networking.service	1	-	-	-	-
/system.slice/nfs-common.service	2	-	-	-	-
/system.slice/rpcbind.service	1	-	-	-	-
/system.slice/rsyslog.service	1	-	-	-	-
/system.slice/ssh.service	1	-	-	-	-
/system.slice/systemd-journald.service	1	-	-	-	-
/system.slice/systemd-logind.service	1	-	-	-	-
/system.slice/systemd-udev.service	1	-	-	-	-
/system.slice/box-guest-utils.service	1	-	-	-	-
/user.slice/...0.slice/session-1.scope	3	-	-	-	-
/user.slice/...-0.slice/user@0.service	2	-	-	-	-

Ilustración 22 - Snippet `systemd-cgtop`

También podemos usar el comando `systemctl list-unit-files | grep enabled`.

```
# systemctl list-unit-files | grep enabled
```

```
root@master:~# systemctl list-unit-files | grep enabled
acpid.path                                enabled
anacron-resume.service                   enabled
anacron.service                         enabled
atd.service                             enabled
cron.service                           enabled
getty@.service                          enabled
hwclock-save.service                   enabled
netfilter-persistent.service            enabled
rsyslog.service                        enabled
ssh.service                            enabled
sshd.service                           enabled
syslog.service                         enabled
acpid.socket                           enabled
remote-fs.target                       enabled
```

Ilustración 23 - Snippet `systemctl list-unit-files | grep enabled`

1.7 Niveles de ejecución

Podemos determinar fácilmente los runlevel de nuestra máquina listando el directorio que los contiene, que es el /etc.

Para ello hacemos lo siguiente:

```
# ls -h /etc/rc y tabulamos
```

```
root@master:~# ls -h /etc/rc
rc0.d/      rc2.d/      rc4.d/      rc6.d/      rcS.d/
rc1.d/      rc3.d/      rc5.d/      rc.local
```

Ilustración 24 - Runlevels presentes en nuestro sistema

1.7.1 Servicios asociados

Para determinar los servicios asociados que tiene cada runlevel usamos el comando `ls -h /etc/rc*`. Al ejecutarlo nos mostrará el contenido de cada carpeta y dicho contenido serán los servicios asociados.

```
# ls -h /etc/rc*
```

1.7.1.1 Runlevel 0

```
/etc/rc0.d:
K01atd
K01isc-dhcp-server
K01netfilter-persistent
K01saned
K01urandom
K01virtualbox-guest-utils
K02sendsigs
K03rsyslog
K04umountnfs.sh
K05nfs-common
K05rpcbind
K06hwclock.sh
K06networking
K07umountfs
K08umountroot
K09halt
README
```

Ilustración 25 - Servicios rc0

1.7.1.2 Runlevel 1

```
/etc/rc1.d:  
K01atd  
K01isc-dhcp-server  
K01netfilter-persistent  
K01saned  
K01virtualbox-guest-utils  
K03rsyslog  
K05nfs-common  
K05rpcbind  
README  
S01killprocs  
S01motd  
S03bootlogs  
S04single
```

Il·lustració 26 - Servici rc1

1.7.1.3 Runlevel 2

```
/etc/rc2.d:  
README  
S01motd  
S01rsyslog  
S01virtualbox-guest-utils  
S02acpid  
S02anacron  
S02atd  
S02cron  
S02dbus  
S02isc-dhcp-server  
S02ssh  
S03bootlogs  
S03saned  
S04rc.local  
S04rmnologin
```

Il·lustració 27 - Servici rc2

1.7.1.4 Runlevel 3

```
/etc/rc3.d:  
README  
S01motd  
S01rsyslog  
S01virtualbox-guest-utils  
S02acpid  
S02anacron  
S02atd  
S02cron  
S02dbus  
S02isc-dhcp-server  
S02ssh  
S03bootlogs  
S03saned  
S04rc.local  
S04rmnologin
```

Il·lustració 28 - Servici rc3

1.7.1.5 Runlevel 4

```
/etc/rc4.d:  
README  
S01motd  
S01rsyslog  
S01virtualbox-guest-utils  
S02acpid  
S02anacron  
S02atd  
S02cron  
S02dbus  
S02isc-dhcp-server  
S02ssh  
S03bootlogs  
S03saned  
S04rc.local  
S04rmnologin
```

Il·lustració 29 - Servici rc4

1.7.1.6 Runlevel 5

```
/etc/rc5.d:  
README  
S01motd  
S01rsyslog  
S01virtualbox-guest-utils  
S02acpid  
S02anacron  
S02atd  
S02cron  
S02dbus  
S02isc-dhcp-server  
S02ssh  
S03bootlogs  
S03saned  
S04rc.local  
S04rmnologin
```

Il·lustració 30 - Servis rc5

1.7.1.7 Runlevel 6

```
/etc/rc6.d:  
K01atd  
K01isc-dhcp-server  
K01netfilter-persistent  
K01saned  
K01urandom  
K01virtualbox-guest-utils  
K02sendsigs  
K03rsyslog  
K04umountnfs.sh  
K05nfs-common  
K05rpcbind  
K06hwclock.sh  
K06networking  
K07umountfs  
K08umountroot  
K09reboot  
README
```

Il·lustració 31 - Servis rc6

1.7.1.8 Runlevel System

```
/etc/rcS.d:
README
S01hostname.sh
S01mountkernfs.sh
S02udev
S03keyboard-setup
S04mountdevsubfs.sh
S05hwclock.sh
S06checkroot.sh
S07checkfs.sh
S08checkroot-bootclean.sh
S08kmod
S09mountall.sh
S10mountall-bootclean.sh
S11procps
S11udev-finish
S11urandom
S12networking
S13rpcbind
S14nfs-common
S15mountnfs.sh
S16mountnfs-bootclean.sh
S17kbd
S18console-setup
```

Il·lustració 32 - Servicios rcS

1.7.2 Cambiando el *runlevel*

Para cambiar de *runlevel* tan sólo es necesario ejecutar un comando, por ejemplo *telinit*.

Primero verificamos en qué *runlevel* nos encontramos con el comando *who -r*.

```
# who -r
```

```
root@master:~# who -r
      run-level 5  2016-12-06 11:42
```

Il·lustració 33 - Snippet *who -r*

La ilustración 33 nos muestra que estamos en el *runlevel* 5.

Ahora, si queremos trabajar con el *runlevel* 3 ejecutamos el siguiente comando:

```
# telinit 3
```

Ejecutamos y verificamos el *runlevel* actual con *who -r*

```
root@master:~# telinit 3
root@master:~# who -r
      run-level 3  2016-12-06 11:58      last=5
root@master:~# _
```

Il·lustració 34 - Snippet *telinit*

1.8 Entorno de trabajo

1.8.1 Con GUI sin X-Window

Para poder trabajar con una GUI sin necesidad de usar X-Window, deberíamos instalar el entorno gráfico deseado. Xfce, KDE, Gnome... Entre otros.

1.8.2 Sin GUI con X-Windows

Lo mismo que en el paso anterior pero desinstalando el entorno gráfico instalado y modificando los parámetros pertinentes en el boot manager.

1.9 BootManager

Por defecto nuestra infraestructura usa GRUB.

Para poder cambiar los parámetros del grub, nos debemos dirigir a `/etc/default/grub`.

```
# vi /etc/default/grub
```

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480
```

Ilustración 35 - Parámetros GRUB

Los eventos del sistema quedan registrados en el log del sistema.

Para verlo, usamos el comando cat.

```
# cat /var/log/syslog
```

```
Dec 6 12:18:41 master acpid: starting up with netlink and the input layer
Dec 6 12:18:41 master acpid: 1 rule loaded
Dec 6 12:18:41 master acpid: waiting for events: event logging is off
Dec 6 12:19:02 master systemd[553]: Starting Paths.
Dec 6 12:19:02 master systemd[553]: Reached target Paths.
Dec 6 12:19:02 master systemd[553]: Starting Timers.
Dec 6 12:19:02 master systemd[553]: Reached target Timers.
Dec 6 12:19:02 master systemd[553]: Starting Sockets.
Dec 6 12:19:02 master systemd[553]: Reached target Sockets.
Dec 6 12:19:02 master systemd[553]: Starting Basic System.
Dec 6 12:19:02 master systemd[553]: Reached target Basic System.
Dec 6 12:19:02 master systemd[553]: Starting Default.
Dec 6 12:19:02 master systemd[553]: Reached target Default.
Dec 6 12:19:02 master systemd[553]: Startup finished in 56ms.
Dec 6 13:17:01 master CRON[740]: (root) CMD ( cd / && run-parts --report /etc
/cron.hourly)
Dec 6 13:17:41 master kernel: [ 9102.711296] e1000: eth0 NIC Link is Down
Dec 6 14:40:52 master systemd[553]: Time has been changed
Dec 6 14:40:57 master kernel: [ 9108.723930] e1000: eth0 NIC Link is Up 1000 Mb
ps Full Duplex, Flow Control: RX
```

Ilustración 36 - Log del system

1.10 Sistema de paquetes

1.10.1 Paquetes instalados en el sistema

Si conocer los paquetes instalados en nuestro sistema, debemos usar el comando *dpkg-query*-, este comando nos listará todos ellos de forma detallada. El output es muy largo por lo que se muestra una parte del mismo.

```
# dpkg-query-l
```

```
ii  xscreensaver  5.30-1+deb8u amd64      Screensaver daemon and frontend f
ii  xscreensaver-d 5.30-1+deb8u amd64      Screen saver modules for screensa
ii  xserver-common 2:1.16.4-1   all          common files used by various X se
ii  xserver-xorg   1:7.7+7      amd64       X.Org X server
ii  xserver-xorg-c 2:1.16.4-1   amd64       Xorg X server - core server
ii  xserver-xorg-i 1:7.7+7      amd64       X.Org X server -- input driver me
ii  xserver-xorg-i 1:2.9.0-2    amd64       X.Org X server -- evdev input dri
ii  xserver-xorg-i 1:1.9.1-1    amd64       X.Org X server -- mouse input dri
ii  xserver-xorg-i 1.8.1-1     amd64       Synaptics TouchPad driver for X.O
ii  xserver-xorg-i 1:13.0.0-1+b amd64       X.Org X server -- VMMouse input d
ii  xserver-xorg-i 0.26.0+20140 amd64       X.Org X server -- Wacom input dri
ii  xserver-xorg-v 1:2.3.3-1+b3 amd64       X.Org X server -- VESA display dr
ii  xterm          312-2       amd64       X terminal emulator
ii  xz-utils       5.1.1alpha+2 amd64       XZ-format compression utilities
ii  zlib1g:amd64   1:1.2.8.dfsg amd64       compression library - runtime
```

Ilustración 37 - Output dpkg-query-l

1.10.2 Verificar actualizaciones

Para verificar si hay actualizaciones de nuestra distro, usamos el comando *apt-get upgrade -s*.

Lo que hará es hacer una simulación del comando sin llegar a escribir nada en el disco, por lo que de detectar que hay actualizaciones lo podríamos ver.

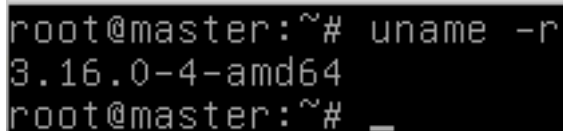
```
# apt-get upgrade -s
```

1.11 Kernel

1.11.1 El kernel de nuestro entorno virtual

Usamos el comando `uname -r` para determinar la versión actual de nuestro kernel.

```
# uname -r
```



```
root@master:~# uname -r
3.16.0-4-amd64
root@master:~# _
```

Ilustración 38 - Snippet `uname -r`

1.11.2 Operar con distintos kernels

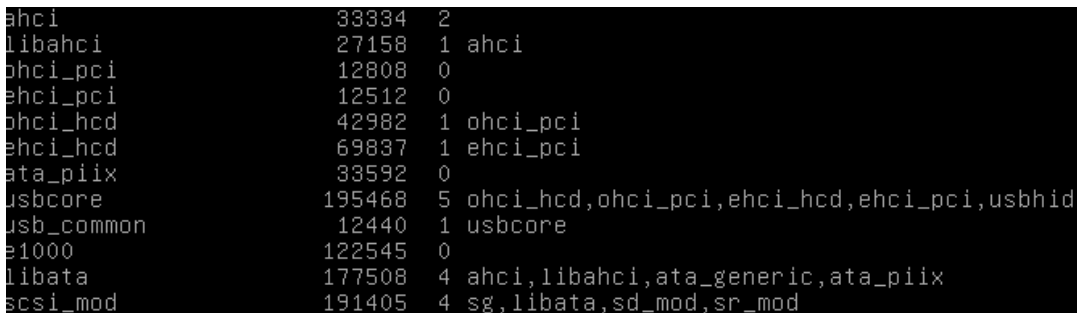
Para instalar otro kernel en nuestro sistema usamos el comando `apt-get search linux-image`.

```
# apt-get search Linux-image
```

1.11.3 Módulos activos

Si lo que queremos es ver los módulos activos que está usando nuestro kernel ejecutamos el comando `lsmod`. El output es bastante largo por lo que se muestra una parte.

```
# lsmod
```



```
ahci                33334      2
libahci             27158      1 ahci
ohci_pci            12808      0
ehci_pci            12512      0
ohci_hcd            42982      1 ohci_pci
ehci_hcd            69837      1 ehci_pci
ata_piix             33592      0
usbcore             195468     5 ohci_hcd,ohci_pci,ehci_hcd,ehci_pci,usbhid
usb_common           12440      1 usbcore
e1000                122545     0
libata              177508     4 ahci,libahci,ata_generic,ata_piix
scsi_mod            191405     4 sg,libata,sd_mod,sr_mod
```

Ilustración 39 - Output `lsmod`

1.12 Red

1.12.1 Dispositivos y redes activos en nuestro entorno

Para determinar los dispositivos de red y la configuración de cada uno de ellos, nos dirigimos a `/etc/network/` y printamos el archivo `interfaces`.

```
# cat /etc/network/interfaces
```

```
root@master:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
# INAT CONNECTION!
auto eth0
iface eth0 inet dhcp

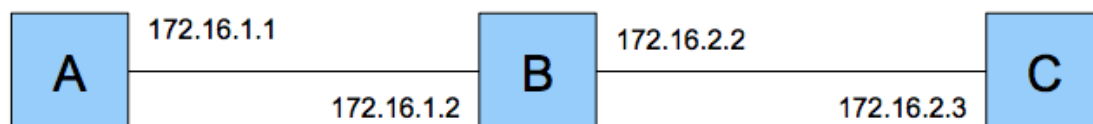
# The secondary network
# Refers to <intMaster>
auto eth1
iface eth1 inet static
address 172.16.1.1
netmask 255.255.255.0

up ip route add 172.16.2.0/24 via 172.16.1.2 dev eth1
```

Il·lustració 40 - Snippet cat

1.13 Configuración de la red

Se pretende seguir el esquema de a continuación.



Il·lustració 41 - Esquema red

1.13.1 Configuración del master

```
root@master:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
# INAT CONNECTION!
auto eth0
iface eth0 inet dhcp

# The secondary network
# Refers to <intMaster>
auto eth1
iface eth1 inet static
address 172.16.1.1
netmask 255.255.255.0

up ip route add 172.16.2.0/24 via 172.16.1.2 dev eth1
```

Ilustración 42 - Configuración del master

El master debe ser capaz de enrutar por lo que se tiene que setear un parámetro en el archivo */proc/sys/net/ipv4/ip_forward*.

Mostramos el contenido:

```
root@master:~# cat /proc/sys/net/ipv4/ip_forward
1
```

Ilustración 43 - Opción enrutador

Además, se deben añadir las rutas para el master

```
root@master:~# ip route
default via 10.0.2.2 dev eth0
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15
169.254.0.0/16 dev eth0 scope link metric 1000
172.16.1.0/24 dev eth1 proto kernel scope link src 172.16.1.1
172.16.2.0/24 via 172.16.1.2 dev eth1
```

Ilustración 44 - Rutas master

1.13.2 Configuración slave1

```
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*
##### ----- SLAVE 1 ----- #####
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 172.16.1.2
netmask 255.255.255.0
gateway 172.16.1.1

# The secondary network interface
# Refers to:
#     Internal network:
#     - intSlaves

auto eth1
iface eth1 inet static
address 172.16.2.2
netmask 255.255.255.0
root@slave1:~# _
```

Ilustración 45 - Configuración Slave1

Slave1 debe ser capaz de enrutar por lo que se tiene que setear un parámetro en el archivo `/proc/sys/net/ipv4/ip_forward`.

Mostramos el contenido:

```
root@slave1:~# cat /proc/sys/net/ipv4/ip_forward
1
```

Ilustración 46 - Opción enrudador slave1

Mostramos las rutas de slave1:

```
root@slave1:~# ip route
default via 172.16.1.1 dev eth0
169.254.0.0/16 dev eth1 scope link metric 1000
172.16.1.0/24 dev eth0 proto kernel scope link src 172.16.1.2
172.16.2.0/24 dev eth1 proto kernel scope link src 172.16.2.2
```

Ilustración 47 - Rutas slave1

1.13.3 Configuración slave2

```
permitted by applicable law.
/etroot@slave2:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

##### ----- SLAVE 2 ----- #####

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface

# Refers to:
#     Internal Network:
#     - intSlaves

allow-hotplug eth0
iface eth0 inet static
address 172.16.2.3
netmask 255.255.255.0
gateway 172.16.2.2
```

Ilustración 48 - Configuración slave2

En este caso, slave2 no debe tener la capacidad de enrutar nada, por lo que la opción de `ip_forward` no la tiene activada.

Las rutas de slave2 son las siguientes:

```
root@slave2:~# ip route
default via 172.16.2.2 dev eth0
169.254.0.0/16 dev eth0 scope link metric 1000
172.16.2.0/24 dev eth0 proto kernel scope link src 172.16.2.3
```

Ilustración 49 - Rutas slave2

1.14 Verificaciones de funcionamiento

1.14.1 Ping de master a ambos slaves

```
root@master:~# ping slave1a.gm.org
PING slave1a.gm.org (172.16.1.2) 56(84) bytes of data.
64 bytes from slave1a.gm.org (172.16.1.2): icmp_seq=1 ttl=64 time=0.543 ms
64 bytes from slave1a.gm.org (172.16.1.2): icmp_seq=2 ttl=64 time=0.697 ms
64 bytes from slave1a.gm.org (172.16.1.2): icmp_seq=3 ttl=64 time=0.795 ms
^C
--- slave1a.gm.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.543/0.678/0.795/0.105 ms
```

Ilustración 50 - Ping de master a slave1 extremo A


```
root@master:~# ping slave1c.gm.org
PING slave1c.gm.org (172.16.2.2) 56(84) bytes of data.
64 bytes from slave1c.gm.org (172.16.2.2): icmp_seq=1 ttl=64 time=0.538 ms
64 bytes from slave1c.gm.org (172.16.2.2): icmp_seq=2 ttl=64 time=0.360 ms
64 bytes from slave1c.gm.org (172.16.2.2): icmp_seq=3 ttl=64 time=0.414 ms
^C
--- slave1c.gm.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.360/0.437/0.538/0.076 ms
```

Il·lustració 51 - Ping de master a slave1 Extremo C

```
root@master:~# ping slave2.gm.org
PING slave2.gm.org (172.16.2.3) 56(84) bytes of data.
64 bytes from slave2.gm.org (172.16.2.3): icmp_seq=1 ttl=63 time=1.32 ms
64 bytes from slave2.gm.org (172.16.2.3): icmp_seq=2 ttl=63 time=1.08 ms
64 bytes from slave2.gm.org (172.16.2.3): icmp_seq=3 ttl=63 time=0.684 ms
^C
--- slave2.gm.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.684/1.031/1.329/0.268 ms
```

Il·lustració 52 - Ping de master a slave2

1.14.2 Ping de master a Internet

```
root@master:~# ping google.es
PING google.es (216.58.210.163) 56(84) bytes of data.
64 bytes from mad06s10-in-f3.1e100.net (216.58.210.163): icmp_seq=1 ttl=63 time=43.6 ms
64 bytes from mad06s10-in-f3.1e100.net (216.58.210.163): icmp_seq=2 ttl=63 time=42.7 ms
64 bytes from mad06s10-in-f3.1e100.net (216.58.210.163): icmp_seq=3 ttl=63 time=42.6 ms
^C
--- google.es ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 42.683/43.050/43.682/0.479 ms
```

Il·lustració 53 - Ping de master a Google

1.14.3 Ping de slave1 a master y slave2

```
root@slave1:~# ping master.gm.org
PING master.gm.org (172.16.1.1) 56(84) bytes of data.
64 bytes from master.gm.org (172.16.1.1): icmp_seq=1 ttl=64 time=0.380 ms
64 bytes from master.gm.org (172.16.1.1): icmp_seq=2 ttl=64 time=0.346 ms
64 bytes from master.gm.org (172.16.1.1): icmp_seq=3 ttl=64 time=0.784 ms
^C
--- master.gm.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.346/0.503/0.784/0.199 ms
root@slave1:~# ping slave2.gm.org
PING slave2.gm.org (172.16.2.3) 56(84) bytes of data.
64 bytes from slave2.gm.org (172.16.2.3): icmp_seq=1 ttl=64 time=0.364 ms
64 bytes from slave2.gm.org (172.16.2.3): icmp_seq=2 ttl=64 time=0.780 ms
64 bytes from slave2.gm.org (172.16.2.3): icmp_seq=3 ttl=64 time=0.409 ms
^C
--- slave2.gm.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.364/0.517/0.780/0.188 ms
```

Il·lustració 54 - Pings de slave1 a master y a slave2

1.14.4 Ping de slave1 a Internet

```
root@slave1:~# ping google.es
PING google.es (216.58.201.131) 56(84) bytes of data.
64 bytes from mad06s25-in-f131.1e100.net (216.58.201.131): icmp_seq=1 ttl=61 time=77.6 ms
64 bytes from mad06s25-in-f131.1e100.net (216.58.201.131): icmp_seq=2 ttl=61 time=73.3 ms
64 bytes from mad06s25-in-f131.1e100.net (216.58.201.131): icmp_seq=3 ttl=61 time=46.5 ms
^C
--- google.es ping statistics ---
^X3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 46.581/65.853/77.612/13.737 ms
```

Il·lustració 55 - Ping de slave1 a Internet

1.14.5 Ping de slave2 a master y a slave1

```
root@slave2:~# ping slave1c.gm.org
PING slave1c.gm.org (172.16.2.2) 56(84) bytes of data.
64 bytes from slave1c.gm.org (172.16.2.2): icmp_seq=1 ttl=64 time=0.563 ms
64 bytes from slave1c.gm.org (172.16.2.2): icmp_seq=2 ttl=64 time=0.682 ms
64 bytes from slave1c.gm.org (172.16.2.2): icmp_seq=3 ttl=64 time=0.537 ms
^C
--- slave1c.gm.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.537/0.594/0.682/0.063 ms
root@slave2:~# ping slave1a.gm.org
PING slave1a.gm.org (172.16.1.2) 56(84) bytes of data.
64 bytes from slave1a.gm.org (172.16.1.2): icmp_seq=1 ttl=64 time=0.709 ms
64 bytes from slave1a.gm.org (172.16.1.2): icmp_seq=2 ttl=64 time=0.342 ms
64 bytes from slave1a.gm.org (172.16.1.2): icmp_seq=3 ttl=64 time=0.423 ms
^C
--- slave1a.gm.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.342/0.491/0.709/0.158 ms
root@slave2:~# _
```

Il·lustració 56 - Ping de slave2 a slave1 (ambos extrems)

```
root@slave2:~# ping master.gm.org
PING master.gm.org (172.16.1.1) 56(84) bytes of data.
64 bytes from master.gm.org (172.16.1.1): icmp_seq=1 ttl=63 time=0.605 ms
64 bytes from master.gm.org (172.16.1.1): icmp_seq=2 ttl=63 time=0.806 ms
64 bytes from master.gm.org (172.16.1.1): icmp_seq=3 ttl=63 time=1.71 ms
^C
--- master.gm.org ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.605/1.040/1.711/0.482 ms
root@slave2:~# _
```

Il·lustració 57 - Ping de slave2 a master

1.14.6 Ping de slave 2 a Internet

```
root@slave2:~# ping google.es
PING google.es (216.58.201.131) 56(84) bytes of data.
64 bytes from mad06s25-in-f131.1e100.net (216.58.201.131): icmp_seq=1 ttl=59 time=54.0 ms
64 bytes from mad06s25-in-f131.1e100.net (216.58.201.131): icmp_seq=2 ttl=59 time=59.4 ms
64 bytes from mad06s25-in-f131.1e100.net (216.58.201.131): icmp_seq=3 ttl=59 time=58.3 ms
^C
--- google.es ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 54.025/57.271/59.410/2.341 ms
root@slave2:~# _
```

2. Segundo Informe – SSH, DHCP y DNS

2.1 OpenSSH Server

2.1.1 Instalación

Por defecto, los paquetes de OpenSSH ya vienen instalados, sí que es cierto que se puede realizar una configuración adicional pero para el funcionamiento básico, no es necesario hacer ningún cambio sobre el fichero de configuración.

A continuación se puede apreciar como, efectivamente, el paquete *openssh-server* ya se encuentra en nuestro equipo:

```
# apt-get install openssh-server
```

```
root@master:~# apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 85 not upgraded.
```

Ilustración 58 - Paquetes OpenSSH Server

2.1.2 Pruebas de funcionamiento básico

Para verificar que funciona correctamente, se procede a realizar una conexión desde el master a los demás equipos. Con esta comprobación bastará para verificar que se permite la conexión bidireccional entre los nodos. Cierto es que en un entorno cuya configuración fuese más avanzada, esta afirmación no la podríamos suponer ya que se pueden establecer distintas políticas de conexión.

Como en el escenario actual, todo está por defecto, podemos asumirlo.

```
root@master:~# ssh hector@slave1a.gm.org
hector@slave1a.gm.org's password: _
```

Ilustración 59 - Conexión SSH master - slave1a

```
root@master:~# ssh hector@slave1c.gm.org
hector@slave1c.gm.org's password: _
```

Ilustración 60 - Conexión SSH master - slave1c

```
root@master:~# ssh hector@slave2.gm.org
hector@slave2.gm.org's password: _
```

Ilustración 61 - Conexión SSH master - slave2

Tal y como se ve en las ilustraciones anteriores, se recibe el comportamiento esperado. Al lanzar el comando *ssh*, el sistema nos pide la contraseña.

En los siguientes pasos se configurará el sistema para que no sea necesaria la entrada de una contraseña.

2.1.3 Configuración de acceso mediante pubkey

Como ya se ha dicho en la sección anterior, se procede a realizar la configuración necesaria para que el sistema no nos pida contraseña cuando nos conectemos a un host conocido vía SSH.

El primer paso que debemos hacer es generar un par de claves (pública y privada) para el master con el usuario root. Lo dejamos todo por defecto.

Para poder generar dicho par de claves, usamos el comando `ssh-keygen`.

```
# ssh-keygen -b 4096 -t rsa
```

```
root@master:~# ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
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:
fb:94:aa:e2:ca:5f:6e:32:70:ed:de:8d:30:36:f2:fa root@master.gm.org
The key's randomart image is:
+---[RSA 4096]---+
|
|      . S
|    . . . . .
|   . . * . .
|  . +*O= *
| o++OE.+ o
+-----+

```

Ilustración 62 - Generación par de llaves

Ahora que ya disponemos del par de claves debemos exportar las públicas para que sean añadidas en el archivo `authorized_keys` de cada slave, archivo que se muestra a continuación.

```
root@master:~# ls -h .ssh/authorized_keys
.ssh/authorized_keys
root@master:~# cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACUirfIv6Cs4GoFjd1BBV5fTGrM21ohCkKqyBJBI9/B
/JCpJi9KX53eF1okq386Jem8wi/JbWpM0aAgFtdFWIDAmd2PHLn8k8gjo3JhfjJSnRbk3Hcaf7TneQtd
pWxgsxU0/E9whsX55H+v2phd/mxd+3mX0n+9kR2WOL5HwYDhXB1W/KF9kEMVL9occk4MJT+PC151rDFv
daXK1MW1DBPDVVLJaDr6t9nyU7f8NyB0VeWpLDD5M3TKihixwzSFoBtez6XDH2j4hImyoMRwPToYHjP
12R0C7PBf3B0z5Y/x2VWzg2aMVGj0eYBnYKrWp8CMdoabq0xHvdc+DwvQcNh39REpQ1eqfP5+x6pTDKe
yFR061vPz28qCQ+wzMQHPLmRJTzv/OJa0qN0p2yJfVwoT79wHhXdfi4SkMeq1U34fPaAR01G3ad5KJcny
+XLtonTj0JMSw6D9oYxbyPyra6fvczbibD38BYuAGoRCl1J4oMv7tvpq3/yKSCyBHWeHnQV61GuifqqK
QvMCay4d5MRUNRYZvSp191H737Tr001sr0nKRRJU+u0rq1C70gwks8/H0bu0g/qy0PBdA1PSxp15D1nC
FT0Fiux6hMM6FsX1hWqwmMLdHmKHAHbMra821kg21kj2A+zR787GwsgBEhgtx7LVspw/SJiGh48p+y1
8Q== root@slave1.gm.org
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACQDQr6TfRnMsfnUeAzTSnUjHndyNrZwXpsyqh9maN1a
YurHH5WshH5H59AgPs0ZKF1wq/6pn7cTYPG1L01Wgpg/cJwSvksirkij1R5QddiX1/GF9mHjwE3aq2t
YEU0xMdRud2o5KN4cQ0i/K1LLgBTfsKzDQ4d8eVfvou9BGj+TLFd0c8GM6NN2vNGwctJGmxNgWWJgXFY
sDm7Ge1buJK55b99VxHAKe/BdvSkUHIJkrWR0+1Dqu1GLU7yp4BPv2T0nXGhQ+231lpMqpE2xNVcmJPd
vz4gp4oduy016ebhA+/X19MJSe2sy1qIHX55201yNyZsTeBDjLVG9Uf+roEJJX1RyMSq5D9va1Cq7qKW
NWK0cTxBychhcL+nuUPemsawy1rkYC+e5Pwu8R1Bd3k/F54MR4G2zMsEi1B2Ddd8x1s3v2xTP5Hd/v6
wStwTbhHu6yLbsegeia3p1U1fqr0cUG7QPRDfTaijcd/H61x/NKdIwyz+wrp413MM+35aI4s0JMapBfg
pQNNukyJgf+8ksYIPwkrT/J7bFXN1fu7UDeTJ07Ih6JMCW1j61d378S3v03E/g09ntcNt2mYp5F5x300K
8kmPbCT8WG1DiYKGTJ06X0GkxAv1wr9eF0y6UaHtLF705mHjajhkdIbdr1bft4CTX1UaaILTddGKtUng
IQ== root@slave2.gm.org
```

Ilustración 63 - Listamos y mostramos el contenido de authorized_keys

Para ello, utilizamos el comando `ssh-copy-id user@server` donde user irá root y en server el slave deseado.

```
# ssh-copy-id root@slave1a.gm.org
# ssh-copy-id root@slave1c.gm.org
```

Se trata del mismo equipo, por lo que con una pubkey ya basta pese a que tenga dos interfaces de red.

```
root@master:~# ssh-copy-id root@slave1a.gm.org
/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@slave1a.gm.org's password:
Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@slave1a.gm.org'"
and check to make sure that only the key(s) you wanted were added.
```

Ilustración 64 - Copia de llaves a slave1 desde master

```
# ssh-copy-id root@slave2.gm.org
```

```
root@master:~# ssh-copy-id root@slave2.gm.org
/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@slave2.gm.org's password:
Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@slave2.gm.org'"
and check to make sure that only the key(s) you wanted were added.
```

Ilustración 65 - Copia de llaves a slave2 desde master

Este proceso se debe repetir para cada host y hacer todas las combinaciones posibles.

Se ha mostrado de master a slave1a/c y slave2.

También se lleva a cabo la configuración de slave1a a master, slave1c (aunque pueda parecer absurdo) y slave2. Después, desde slave1c a slave1a, master y slave2. Por último, desde slave2 a slave1c/a y master.

2.1.4 Pruebas de funcionamiento avanzado (con pubkey)

2.1.4.1 De master a slave1a/c y slave2

```
root@master:~# ssh slave1a.gm.org

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  8 11:04:29 2016 from master.gm.org
root@slave1:~# _
```

Ilustración 66 - De master a slave1a

```
root@master:~# ssh slave1c.gm.org

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  8 11:07:01 2016 from master.gm.org
root@slave1:~# _
```

Il·lustració 67 - De master a slave1c

```
root@master:~# ssh slave2.gm.org

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  8 10:19:13 2016
root@slave2:~# _
```

Il·lustració 68 - De master a slave2

2.1.4.2 De slave1 a slave2 y master

```
root@slave1:~# ssh slave2.gm.org

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  8 11:08:44 2016 from master.gm.org
root@slave2:~# _
```

Il·lustració 69 - De slave1 a slave2

```
root@slave1:~# ssh master.gm.org

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  8 10:19:22 2016 from slave2.gm.org
root@master:~# _
```

Il·lustració 70 - De slave1 a master

2.1.4.3 De slave2 a slave1 y master

```
root@slave2:~# ssh slave1a.gm.org

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  8 11:10:58 2016 from slave2.gm.org
root@slave1:~# _
```

Il·lustració 71 - De slave2 a slave1

```
root@slave2:~# ssh master.gm.org

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  8 11:09:21 2016 from slave1a.gm.org
root@master:~# _
```

Il·lustració 72 - De slave2 a master

2.1.5 Mejoras acceso vía SSH

Hay varios métodos para mejorar el acceso vía SSH. Una buena práctica es no permitir el acceso ssh con la cuenta root. Para ello se debe modificar el archivo de configuración de ssh que se encuentra en `/etc/ssh/sshd_config` una vez allí, modificamos el parámetro referente al log del root.

Modificamos de *yes* a *no*.

```
# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes
```

Il·lustració 73 - Root auth

Además, también hay la posibilidad de acceder con clave pública tal y como ha sido configurado el sistema en las secciones anteriores pero añadiendo otro reto al usuario mediante un passphrase en la clave pública. Eso supone que, además de necesitar la clave pública, a dicho usuario se le requerirá una contraseña o de lo contrario, pese a disponer de la clave, no podrá acceder al sistema.

2.2 DHCP

2.2.1 Instalación

Para usar este servicio es necesaria la instalación de una serie de paquetes. Para nuestro entorno se decide instalar el paquete `isc-dhcp-server`.

Sobre el master, que es el servidor que realizará las tareas de DHCP se ejecuta el comando de instalación.

```
# apt-get install isc-dhcp-server
```

```
root@master:~# apt-get install isc-dhcp-server
```

Ilustración 74 - Snippet apt-get install

2.2.2 Pasos previos

2.2.2.1 Recolección de datos de los clientes DHCP

Slave1

```
root@slave1:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:a0:c5:d6 brd ff:ff:ff:ff:ff:ff
    inet 172.16.1.2/24 brd 172.16.1.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fea0:c5d6/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:2e:fb:b7 brd ff:ff:ff:ff:ff:ff
    inet 172.16.2.2/24 brd 172.16.2.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe2e:fb7/64 scope link
        valid_lft forever preferred_lft forever
root@slave1:~# _
```

Ilustración 75 - Datos relativos a Slave1

Slave2

```
root@slave2:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:2e:2b:68 brd ff:ff:ff:ff:ff:ff
    inet 172.16.2.3/24 brd 172.16.2.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe2e:2b68/64 scope link
        valid_lft forever preferred_lft forever
root@slave2:~# _
```

Ilustración 76 - Datos relativos a Slave2

De este modo conocemos las IPs que deben tener y, también, las direcciones físicas (MAC) que serán necesarias para la posterior configuración.

2.2.3 Configuración

Una vez tenemos todos los datos ya podemos proceder a la configuración del servicio dhcp.

2.2.4 Asignación IP a Slave2. Problemas, isc-dhcp-relay y config de interfaces

2.2.5 Asignación estática para Slave1 y Slave2

Desde el master, nos dirigimos al archivo *dhcp.conf* localizado en */etc/dhcp* y lo abrimos.

```
# vi /etc/dhcp/dhcp.conf
```

Nos desplazamos hasta el final y añadimos nuestra configuración deseada, que es la siguiente.

```
host slave1-aside {
    option host-name "slave1a.gm.org";
    hardware ethernet 08:00:27:a0:c5:d6;
    fixed-address 172.16.1.2;
    option domain-name-servers 172.16.1.1;
}
host slave1-cside{
    option host-name "slave1c.gm.org";
    hardware ethernet 08:00:27:2e:fb:b7;
    fixed-address 172.16.2.2
    option domain-name-servers 172.16.1.1;
}
host slave2 {
    option host-name "slave2.gm.org";
    hardware ethernet 08:00:27:2e:2b:68;
    fixed-address 172.16.2.3;
    option domain-name-servers 172.16.1.1;
}
```

Ilustración 77 - Configuración DHCP en master

De este modo, el servidor DHCP asignará las IPs, servidor DNS y hostname a los hosts que cumplan los parámetros requeridos, en este que la MAC sea la que aparece en el fichero de configuración.

2.3 DNS

2.3.1 Instalación

2.3.2 Configuración

3. Tercer Informe – Servicios

3.1 Introducción

En este informe se explica la instalación y configuración de los servicios NFS, DNS, apache y como gestionarlos.

3.2 NFS

3.2.1 Instalación

El proceso de instalación es muy simple. Se debe hacer es descargar los paquetes necesarios para ejecutar correctamente el servicio.

```
# apt-get install nfs-kernel-server nfs-common
```

```
root@slave1:~# apt-get install nfs-kernel-server nfs-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
nfs-common is already the newest version.
The following NEW packages will be installed:
  nfs-kernel-server
0 upgraded, 1 newly installed, 0 to remove and 77 not upgraded.
Need to get 115 kB of archives.
After this operation, 515 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

Il·lustració 78 - Proceso instalació NFS

3.2.2 Configuració

Una vez tenemos los paquetes descargados e instalados correctamente, procedemos a la configuración. Se deben crear los directorios que quieren ser compartidos.

Se define la ruta `/var/nfsdir` como punto de acceso común por y para los equipos A (master), C (Slave2) y cualquier otro dentro del rango IP x.2.x.

Para ello, se crea el directorio y se le asigna los permisos pertinentes.

```
# mkdir /var/nfsdir
```

```
root@slave1:~# mkdir /var/nfsdir_
```

Il·lustració 79 - Creació recurs compartit

```
# chown -R nobody:nogroup /var/nfsdir
```

```
root@slave1:~# chown -R nobody:nogroup /var/nfsdir/_
```

Il·lustració 80 - Asignación de usuario y grupo

```
root@slave1:~# ll /var/ | grep nfs
drwxr-xr-x  2 nobody nogroup 4.0K Dec 12 17:23 nfsdir
root@slave1:~# _
```

Il·lustració 81 - Comprobación asignación

```
# chmod 755 /var/nfsdir
```

```
root@slave1:~# chmod 755 /var/nfsdir/
root@slave1:~# ll /var/ | grep nfs
drwxr-xr-x  2 nobody nogroup 4.0K Dec 12 17:23 nfsdir
root@slave1:~# _
```

Il·lustració 82 - Cambio de permisos y comprobación

Ahora que ya tenemos los directorios creados, debemos añadir los shares en el archivo localizado en */etc/exports*.

```
# vi /etc/exports
```

```
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
# /var/nfsdir 172.16.1.1(ro,sync,no_subtree_check,no_root_squash) 172.16.2.0/24(ro,sync,no_subtree_check,no_root_squash)
#
### NFS DE GM.ORG HECTOR & GERARD_
```

Ilustración 83 - Directorio compartido a master y a cualquier host de la subred 2

Una vez tenemos lista la configuración del servidor, pasamos al cliente. De nuevo, se muestra sólo la configuración de un cliente, ya que para ambos es la misma.

Verificamos que el cliente tenga los paquetes *nfs-common*.

```
root@master:~# apt-get install nfs-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
nfs-common is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 84 not upgraded.
root@master:~# _
```

Ilustración 84 - Instalación NFS cliente

Se procede a la creación del directorio en el cliente para poder montar el recurso NFS.

```
# mkdir -p /mnt/nfs/home
root@master:~# mkdir -p /mnt/nfs/shared
root@master:~# ll /mnt/nfs/shared/
-bash: ll: command not found
root@master:~# alias ll="ls -lh"
root@master:~# ll /mnt/nfs/shared/
total 0
root@master:~# _
```

Ilustración 85 - Directorios creados en cliente

Tan sólo falta montar el recurso sobre el directorio creado recientemente.

```
root@master:~# ll /mnt/nfs/shared/
total 0
root@master:~# mount 172.16.1.2:/var/nfsdir /mnt/nfs/shared/
root@master:~# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/sda1                  3.7G      1.6G   2.0G  45% /
udev                      10M         0   10M   0% /dev
tmpfs                     201M      4.5M   196M   3% /run
tmpfs                     501M         0   501M   0% /dev/shm
tmpfs                     5.0M         0    5.0M   0% /run/lock
tmpfs                     501M         0   501M   0% /sys/fs/cgroup
tmpfs                     101M         0   101M   0% /run/user/0
172.16.1.2:/var/nfsdir    3.7G      1.5G   2.1G  42% /mnt/nfs/shared
root@master:~# _
```

Ilustración 86 - Montamos el recurso y verificamos que está accesible (muestra Master)

```
root@slave2:~# mount 172.16.1.2:/var/nfsdir /mnt/nfs/shared/
root@slave2:~# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/sda1                  3.7G      1.5G   2.1G  42% /
udev                      10M         0   10M   0% /dev
tmpfs                     201M      4.5M   196M   3% /run
tmpfs                     501M         0   501M   0% /dev/shm
tmpfs                     5.0M         0    5.0M   0% /run/lock
tmpfs                     501M         0   501M   0% /sys/fs/cgroup
tmpfs                     101M         0   101M   0% /run/user/0
172.16.1.2:/var/nfsdir    3.7G      1.5G   2.1G  42% /mnt/nfs/shared
root@slave2:~# _
```

Ilustración 87 - Montamos el recurso y verificamos que está accesible (muestra Slave2)

3.3 Apache

3.3.1 Instalación

Para instalar apache en nuestro sistema (slave1) debemos ejecutar el comando de a continuación:

```
# apt-get install apache2
```

```
root@slave1:~# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 79 not upgraded.
Need to get 1,942 kB of archives.
After this operation, 6,643 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

Ilustración 88 - Instalación Apache2 en slave1

Verificamos el estado del servicio

```
# systemctl status apache2.service
```

```
root@slave1:~# systemctl status apache2.service
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─forking.conf
   Active: active (running) since Thu 2016-12-08 12:15:47 GMT; 1min 52s ago
     CGroup: /system.slice/apache2.service
            └─2087 /usr/sbin/apache2 -k start
               2090 /usr/sbin/apache2 -k start
               2091 /usr/sbin/apache2 -k start

Dec 08 12:15:44 slave1.gm.org apache2[2066]: Starting web server: apache2AH0...g
Dec 08 12:15:44 slave1.gm.org apache2[2066]: AH00558: apache2: Could not rel...e
Dec 08 12:15:47 slave1.gm.org apache2[2066]: .
Hint: Some lines were ellipsized, use -l to show in full.
```

Ilustración 89 - Estado apache2

Comprobamos desde un navegador

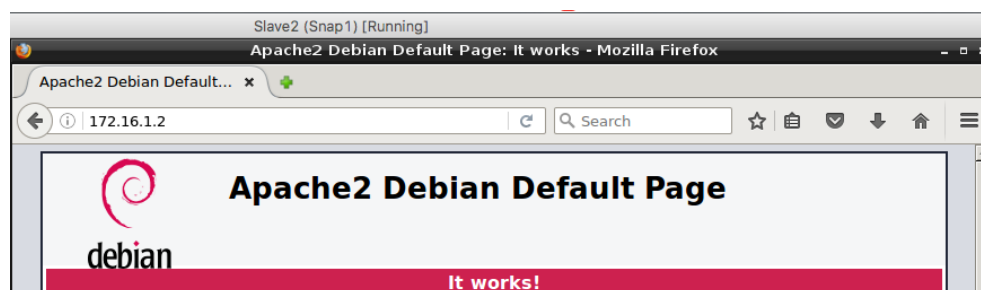


Ilustración 90 - Accedemos al directorio default de apache desde navegador

3.3.2 Edición página por defecto

Para editar la página por defecto debemos dirigirnos al directorio `/var/www/html/` que contiene la página por defecto `index.html`.

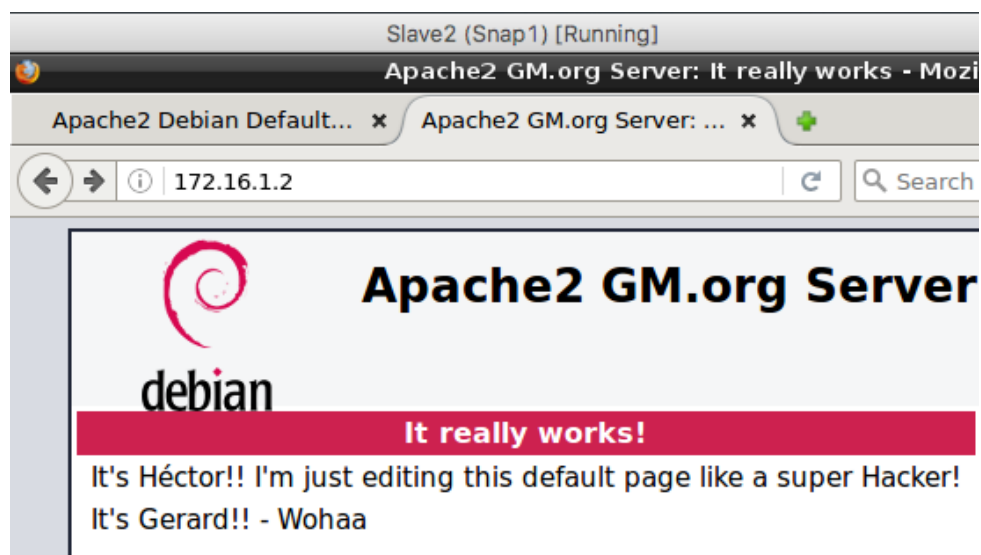


Ilustración 91 - Edición página por defecto

3.3.3 Logs de Apache

Los logs de apache2 nos van a permitir conocer la información de los eventos de nuestros virtualhosts, módulos o todo lo referente al servicio. Es por ello que debemos conocer la ubicación de los logs. Éstos se encuentran en el directorio */var/log/apache2*.

```
root@slave1:/var/log/apache2# pwd
/var/log/apache2
root@slave1:/var/log/apache2# _
```

Ilustración 92 - ubicación logs apache2

```
root@slave1:/var/log/apache2# ls -lh
total 8.0K
-rw-r----- 1 root adm 939 Dec  8 12:28 access.log
-rw-r----- 1 root adm 279 Dec  8 12:15 error.log
-rw-r----- 1 root adm  0 Dec  8 12:15 other_vhosts_access.log
root@slave1:/var/log/apache2# _
```

Ilustración 93 - Contenido carpeta logs de apache2

3.3.4 Módulos

3.3.4.1 PHP

Para que nuestro servidor web interprete el código PHP debemos instalar una serie de módulos. Dicha instalación la haremos siguiendo los comandos que se muestran a continuación:

```
# apt-get install php5 libapache2-mod-php5
root@slave1:/# apt-get install php5 libapache2-mod-php5
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libonig2 libqdbm14 php5-cli php5-common php5-json php5-readline
Suggested packages:
  php-pear php5-user-cache
The following NEW packages will be installed:
  libapache2-mod-php5 libonig2 libqdbm14 php5 php5-cli php5-common php5-json
  php5-readline
0 upgraded, 8 newly installed, 0 to remove and 79 not upgraded.
Need to get 5,422 kB of archives.
After this operation, 21.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

Ilustración 94 - Instalación módulos php5 para apache2

Para verificar que nuestro servidor ya es capaz de interpretar PHP, creamos una nueva página con un print y una función PHP. De este modo comprobaremos rápido si la instalación y configuración se realizó correctamente.

```
# vi /var/www/html/info.php
```

```
<?php
    print "Hello, this is PHP! -GM.org";
    phpinfo();
?>
```

Ilustración 95 - Contenido info.php

Navegamos hasta dicho archivo

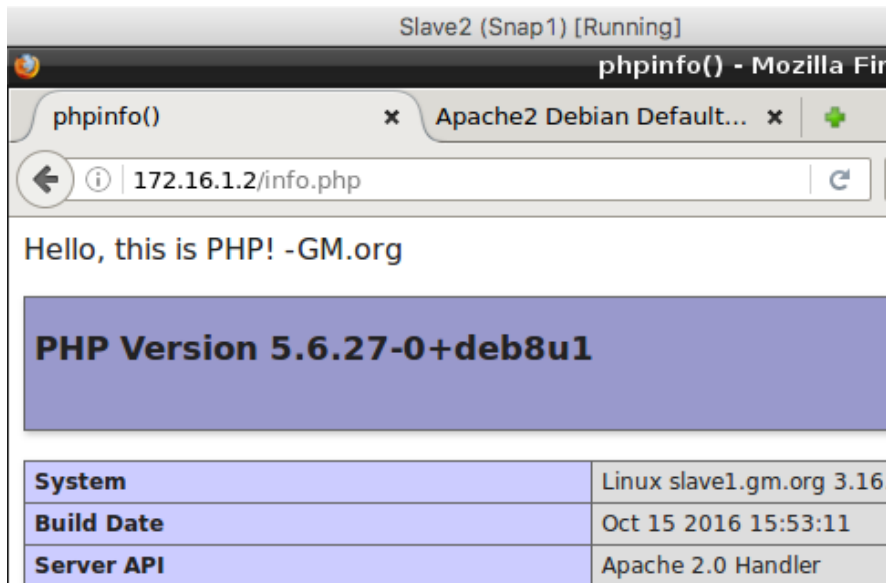


Ilustración 96 - Print e información de nuestro servidor

3.3.4.2 Python

De la misma manera, para que nuestro servidor web interprete el código python debemos instalar una serie de módulos. Dicha instalación la haremos siguiendo los comandos que se muestran a continuación:

```
# apt-get install Python
```

```
root@slave1:/# apt-get install python
Reading package lists... Done
Building dependency tree
Reading state information... Done
python is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 79 not upgraded.
root@slave1:/# _
```

Ilustración 97 - Instalación de Python

En este caso, nuestro servidor ya dispone del intérprete de Python, ahora sólo faltan los módulos para apache2.

```
# apt-get install libapache2-mod-python
```

```
root@slave1:/# apt-get install libapache2-mod-python
Reading package lists... Done
Building dependency tree
Reading state information... Done
libapache2-mod-python is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 79 not upgraded.
root@slave1:/#
```

Ilustración 98 - Instalación módulo python apache2

Lo que se ha hecho con el snippet anterior es habilitar el módulo Python para apache2.

3.3.4.2.1 Configuración Python

Procedemos a la creación del script Python y navegamos hasta la página.

```
# vi /var/www/html/index.py
```

```
print "Content-Type: text/html\n\n"  
print 'Hola Mon des de Python! - GM.org'
```

Ilustración 99 - Contenido index.py

3.3.5 VirtualHosts por IP

3.3.5.1 Preparación

```
root@slave1:/var/www# mkdir -p eth0/public_html eth1/public_html
```

Ilustración 100 - Creación rutas vHosts

```
root@slave1:/var/www# ll  
total 16K  
drwxr-xr-x 3 root root 4.0K Dec 10 13:21 eth0  
drwxr-xr-x 3 root root 4.0K Dec 10 13:21 eth1  
drwxr-xr-x 3 root root 4.0K Dec 10 12:29 html  
drwxr-xr-x 2 root root 4.0K Dec 10 12:30 python  
root@slave1:/var/www# ll *  
eth0:  
total 4.0K  
drwxr-xr-x 2 root root 4.0K Dec 10 13:21 public_html  
eth1:  
total 4.0K  
drwxr-xr-x 2 root root 4.0K Dec 10 13:21 public_html
```

Ilustración 101 - Contenido directorios

```
root@slave1:/var/www# chmod -R 755 eth*  
root@slave1:/var/www# _
```

Ilustración 102 - Asignación de permisos sobre los directorios

```
root@slave1:/var/www# cp html/info.php eth1/public_html/  
root@slave1:/var/www# _
```

Ilustración 103 - Copia de info.php al nuevo vHost eth1

3.3.5.2 Creación vHosts

```
root@slave1:/var/www# vi /etc/apache2/sites-available/eth1_
```

Ilustración 104 - Creación documento vHost eth1

```
root@slave1:/var/www# vi /etc/apache2/sites-available/eth0_
```

Ilustración 105 - Creación documento eth0


```
<VirtualHost 172.16.1.2:80>
    ServerAdmin hector@gm.org
    ServerName py-gm.org
    DocumentRoot /var/www/eth0/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Il·lustració 106 - Virtual Host eth0

```
<VirtualHost 172.16.2.2:80>
    ServerAdmin hector@gm.org
    ServerName php-gm.org
    DocumentRoot /var/www/eth1/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Il·lustració 107 - VirtualHost eth1

```
root@slave1:/var/www# mv /etc/apache2/sites-available/eth0 /etc/apache2/sites-av
ailable/eth0.conf
root@slave1:/var/www# mv /etc/apache2/sites-available/eth1 /etc/apache2/sites-av
ailable/eth1.conf
root@slave1:/var/www# _
```

Il·lustració 108 - Modificació extensió ya que olvidé el .conf en ambos vhosts

3.3.5.3 Habilitando vHosts

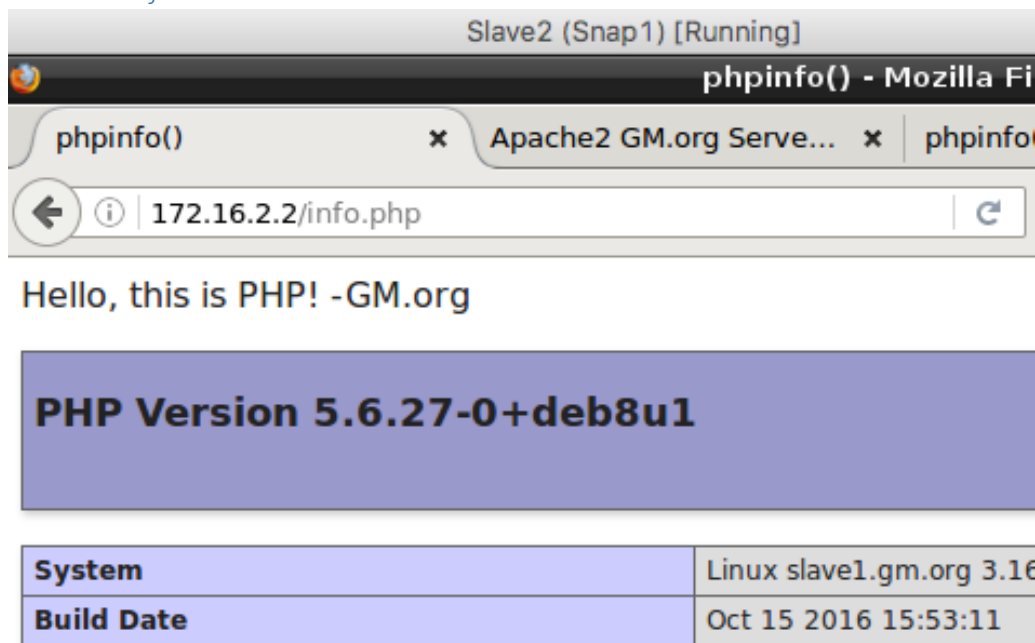
```
root@slave1:/var/www# a2ensite eth0.conf
Enabling site eth0.
To activate the new configuration, you need to run:
    service apache2 reload
root@slave1:/var/www# a2ensite eth1.conf
Enabling site eth1.
To activate the new configuration, you need to run:
    service apache2 reload
root@slave1:/var/www# _
```

Il·lustració 109 - Habilitando ambos vHosts

```
root@slave1:/var/www# service apache2 restart
root@slave1:/var/www# _
```

Il·lustració 110 - Reinicio de Apache2

3.3.5.4 Verificación



3.3.6 Certificados y HTTPS

3.3.6.1 Configuración de Apache

Para poder aceptar peticiones SSL y asignar certificados a los distintos sites de nuestro apache2, vamos a tener que realizar unas configuraciones previas sobre el servidor.

Lo primero será habilitar el SSL y el site por defecto de SSL, después reiniciamos apache2.

```
# a2ensite default-ssl
```

```
# a2enmod ssl
```

```
root@slave1:/var/www# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@slave1:/var/www# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
elf-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
root@slave1:/var/www# _
```

Ilustración 111 - Habilitando SSL y default site de SSL

```
# systemctl restart apache2.service
```

```
root@slave1:/var/www# systemctl restart apache2.service
root@slave1:/var/www# _
```

Ilustración 112 - Reinicio Apache2

Ahora revisamos el sitio desde el navegador y comprobamos que navegamos usando https.



Ilustración 113 - SSL en funcionamiento

3.3.6.2 Generando certificados auto-firmados

Lo primero que vamos a necesitar para poder generar dichos certificados será tener instalados los paquetes de *OpenSSL*.

Usamos el siguiente snippet para instalar o actualizar dichos paquetes en caso de tenerlos instalados:

```
# apt-get install openssl
```

```
root@slave1:/var/www# apt-get install openssl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  openssl
1 upgraded, 0 newly installed, 0 to remove and 78 not upgraded.
Need to get 665 kB of archives.
After this operation, 1,024 B of additional disk space will be used.
Get:1 http://security.debian.org/ jessie/updates/main openssl amd64 1.0.1t-1+deb8u5 [665 kB]
Fetched 665 kB in 0s (776 kB/s)
Reading changelogs... Done
(Reading database ... 64432 files and directories currently installed.)
Preparing to unpack .../openssl_1.0.1t-1+deb8u5_amd64.deb ...
Unpacking openssl (1.0.1t-1+deb8u5) over (1.0.1t-1+deb8u2) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up openssl (1.0.1t-1+deb8u5) ...
root@slave1:/var/www# _
```

Ilustración 114 - Actualización openssl

Una vez disponemos de los paquetes, ya podemos generar certificados.

Primero creamos un directorio para alojar los certificados.

```
root@slave1:~# mkdir /etc/apache2/ssl
root@slave1:~# hostname
slave1.gm.org
root@slave1:~# █
```

Ilustración 115 - Creación directorio ssl

Ahora ya podemos crear el certificado. Usamos el siguiente snippet:

```
# sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
```

```
root@slave1:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.  
key -out /etc/apache2/ssl/apache.crt  
Generating a 2048 bit RSA private key  
.....++  
.....++  
writing new private key to '/etc/apache2/ssl/apache.key'  
-----  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value,  
If you enter '.', the field will be left blank.  
-----  
Country Name (2 letter code) [AU]:ES  
State or Province Name (full name) [Some-State]:BCN  
Locality Name (eg, city) []:Castellar del Valles  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:GM  
Organizational Unit Name (eg, section) []:IT  
Common Name (e.g. server FQDN or YOUR name) [HectorMartinez]:
```

Ilustración 116 - Creación certificado con openssl

A continuación, mostramos el contenido de `/etc/apache2/ssl`.

```
root@slave1:/var/www# ll /etc/apache2/ssl/  
total 8.0K  
-rw-r--r-- 1 root root 1.4K Dec 10 14:06 apache.crt  
-rw-r--r-- 1 root root 1.7K Dec 10 14:06 apache.key  
root@slave1:/var/www# _
```

Ilustración 117 - Key y certificado

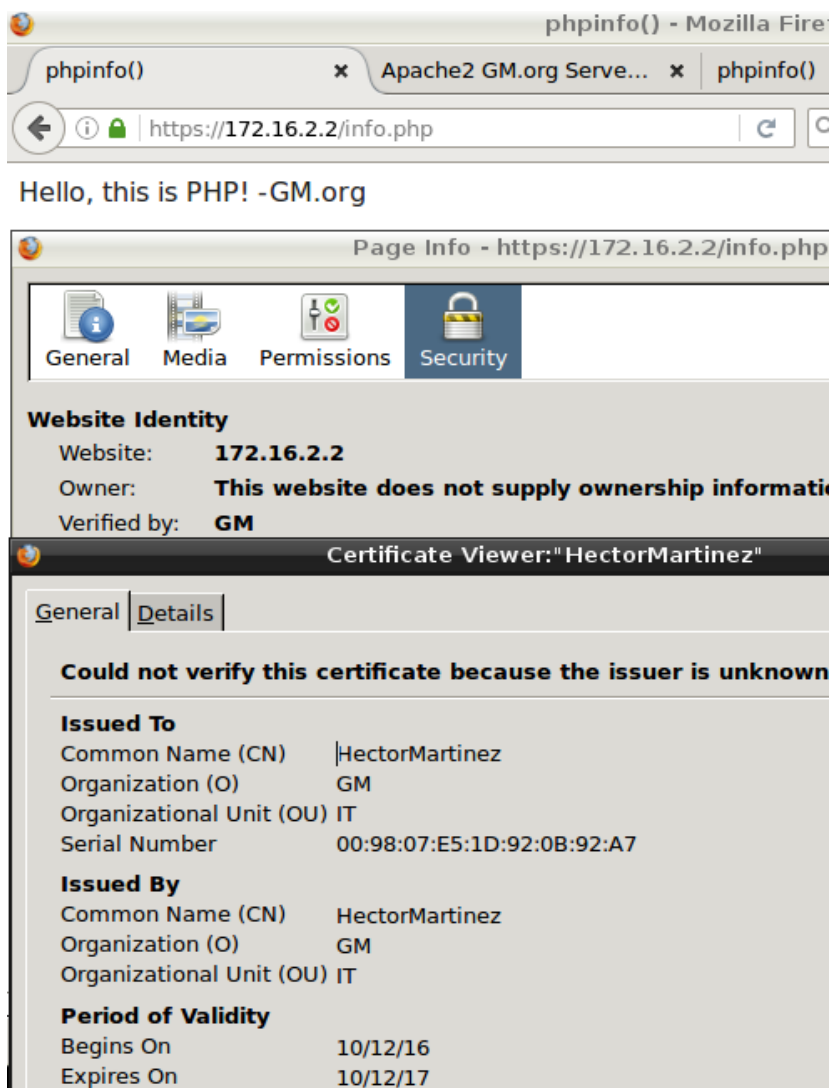
```
<VirtualHost 172.16.2.2:443>  
    SSLEngine on  
    SSLCertificateFile /etc/apache2/ssl/apache.crt  
    SSLCertificateKeyFile /etc/apache2/ssl/apache.key  
    ServerAdmin hector@gm.org  
    ServerName php-gm.org  
    DocumentRoot /var/www/eth1/public_html  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
</VirtualHost>
```

Ilustración 118 - Modificación vHost que muestra PHP

```
root@slave1:/etc/apache2/sites-available# systemctl restart apache2.service  
root@slave1:/etc/apache2/sites-available# _
```

Ilustración 119 - Reiniciamos servicio apache2

A continuación, nos dirigimos al navegador, accedemos a la página que acabamos de modificar para permitir el acceso vía ssl con el certificado recién creado y mostramos la información del mismo.



Il·lustració 120 - Verificació Site PHP funcionant amb certificat auto-firmat

4. Cuarto Informe – Seguridad

4.1 Cuestiones previas

Quina comanda ens permet veure els límits del usuari de processos, file descriptors, memòria, etc?

Comando `prctl`

Si els arxius de log del sistema comencen a créixer indefinidament, com evitariem que ens consumeixi tot l'espai del disc?

Para evitar que los archivos log crezcan indefinidamente con los mensajes, se hace que sean circulares, de forma que la información se mantiene un cierto tiempo.

El paquete logrotate contiene una tarea de cron que hace circular automáticamente los archivos de log.

Com podem evitar que un usuari concret pugui deixar d'executar un binari que es troba a /usr/bin?

Con el archivo `sudoers`.

Què és un atac de DoS? És Apache susceptible d'un atac de DoS? Com el podríem evitar?

Los ataques DDoS se generan mediante la saturación de los puertos con múltiples flujos de información, haciendo que el servidor se sobrecargue y no pueda seguir prestando su servicio.

Los desarrolladores del servicio de servidores web Apache han reconocido que existe una vulnerabilidad que permite recibir ataques de denegación de servicio (DDoS). La amenaza se llama Apache Killer y aprovecha una brecha de seguridad en el módulo mod_deflate.

Por el momento no existe una solución definitiva al fallo, pero si existe un manual con una serie de pasos para evitar la intromisión que deja los sitios caídos.

Gracias al módulo de apache mod_evasive conseguiremos redirigir el tráfico de estas peticiones ilegítimas hacia un error 403 (prohibido).

Quines són les diferents taules de Iptables? Com limitaries o habilitaries un servei concret (per exemple FTP)? Quins són els paràmetres més comuns a Iptables?

Las diferentes tablas de Iptables son; tablas MANGLE, de filtrado y tablas NAT.

Las reglas de firewall sólo estarán activas si se está ejecutando el servicio iptables. Se usa el comando iptables y se edita para habilitar o limitar el servicio concreto.

Describimos algunos de los comandos más comunes:

- A Agregar nueva regla a la cadena especificada.
- I Insertar nueva regla antes de la regla número_regla(rulenum) en la cadena especificada de acuerdo a los parámetros sometida.
- R Reemplazar la regla (rulenum) en la cadena especificada.
- E Modifica el nombre de la cadena.

Para ver el resto de comandos utilizar el comando: man iptables

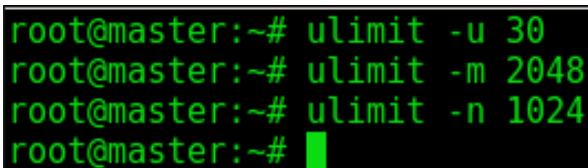
4.2 Ejercicio 1

Usaremos *ulimit* para prevenir distintos tipos de ataque comunes en los sistemas. Limitaremos el uso máximo de memoria, el número máximo de procesos (para evitar las forkbombs) y también el número de file descriptors abiertos simultáneamente.

Para ello ejecutaremos los siguientes comandos:

Limitando máximo de procesos (forkbomb):

```
# ulimit -u 30
# ulimit -m 2048
# ulimit -n 1024
```



```
root@master:~# ulimit -u 30
root@master:~# ulimit -m 2048
root@master:~# ulimit -n 1024
root@master:~#
```

Ilustración 121 - Restricciones con ulimit

Mostramos ahora todas las restricciones aplicadas con ulimit.

```
root@master:~# ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority     (-e) 0
file size               (blocks, -f) unlimited
pending signals         (-i) 3935
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) 2048
open files              (-n) 1024
pipe size               (512 bytes, -p) 8
POSIX message queues    (bytes, -q) 819200
real-time priority      (-r) 0
stack size              (kbytes, -s) 8192
cpu time                (seconds, -t) unlimited
max user processes      (-u) 30
virtual memory          (kbytes, -v) unlimited
file locks              (-x) unlimited
```

Ilustración 122 - Restricciones ulimit

En redhat se podría usar cgroups. En distribuciones Debian también pero la implementación de este sistema en Debian tiene poca documentación.

Aquí vemos el resultado de aplicar los límites:

```
troll@master:~$ :(){ :|:& };;
[1] 695
troll@master:~$ bash: fork: retry: Resource temporarily unavailable
bash: fork: retry: No child processes
bash: fork: retry: No child processes
```

Ilustración 123 - Intento fallido de fork bomb

Para limitar el uso de espacio de disco usaremos cuotas de discos.

Con tal objetivo, verificamos que tengamos los paquetes necesarios instalados en nuestro sistema.

```
# apt-get install quota quotatool
```

En nuestro caso tenemos ya dicho software. Para aplicar la regla que no permita al usuario ocupar más de un determinado tamaño, usamos este comando:

```
# quotatool -u troll -bq 500Mb -l '500 Mb' /home/troll
```

Con la finalidad de prevenir el tamaño de los logs, usamos logrotate. El script con la configuración especificada en /etc/logrotate.conf se ejecutará diariamente por haberlo puesto en cron, en este caso, cron.daily.

```
root@master:/etc/cron.daily# ll logrotate
-rwxr-xr-x 1 root root 89 Nov  8 2014 logrotate
```

Ilustración 124 - Logrotate en cron.daily

4.3 Ejercicio 2

Volem saber quines connexions hi han establertes al nostre sistema i quins ports s'estan utilitzant. Quina comanda utilitzaries?

El comando *netstat* es el que nos va a permitir conocer las conexiones establecidas, ya sean entrantes o salientes sobre el ordenador.

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 master.gm.org:ssh      slave2.gm.org:45426    ESTABLISHED
tcp        0      0 localhost:8649         localhost:38497        TIME_WAIT
udp        0      0 localhost:37032        localhost:8649         ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags               Type                   State                  I-Node  Path
unix    2      [ ]                 DGRAM                  8228                  /run/systemd/journal/
syslog
unix    2      [ ]                 DGRAM                  7761                  /run/systemd/notify
unix    2      [ ]                 DGRAM                  7779                  /run/systemd/shutdown
unix   11      [ ]                 DGRAM                  7781                  /run/systemd/journal/
dev-log
unix    6      [ ]                 DGRAM                  7790                  /run/systemd/journal/
socket
unix    2      [ ]                 DGRAM                  13516                 /run/user/0/systemd/n
otify
unix    3      [ ]                 STREAM                 CONNECTED              11867
unix    3      [ ]                 STREAM                 CONNECTED              11578                 /usr/local/nagios/var
```

Il·lustració 125 - netstat sobre master (el contenido sigue pero no es mostrado en la imagen)

Apreciamos que hay distintas direcciones y puertos en los que se pueden identificar los de nagios, ganglia, ssh, entre otros.

Crieu que hi ha un ordinador dintre de la nostra xarxa local que té algunes connexions sospitoses. Exploreu amb nmap aquesta iP.

Como conexión sospechosa seleccionamos una abierta por *slave1a.gm.org*, así que se procede a realizar el nmap sobre dicha dirección.

```
root@master:~# nmap slave1a.gm.org

Starting Nmap 6.47 ( http://nmap.org ) at 2017-01-04 11:35 GMT
Nmap scan report for slave1a.gm.org (172.16.1.2)
Host is up (0.00068s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
443/tcp   open  https
2049/tcp  open  nfs
5666/tcp  open  nrpe
8649/tcp  open  unknown
MAC Address: 08:00:27:A0:C5:D6 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 2.81 seconds
```

Il·lustració 126 - Resultado nmap

Un usuari mirarà de crear errors a Apache per a omplir els seus logs. Haurem d'evitar que aquesta acció també saturi el nostre sistema.

Una de las soluciones más viables para este problema es configurar apache para que deje de registrar los logs de acceso y también de error.

Para ello debemos desplaarnos al fichero `apache2.conf` (`/etc/apache2/apache2.conf`) y modificar las líneas `LogLevel warn` y `LogLevel info` por `LogLevel Emerg`.

```
#LogLevel warn
LogLevel emerg
```

Il·lustració 127 - Prevenció de saturació de logs de apache2

Además, también debemos modificar el formato del log, de `common` a `combined`.

```
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %D \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %D \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%{Referer}i -> %U" referer
#LogFormat "%h %l %u %t \"%r\" %>s %D" common
LogFormat "%h %l %u %t \"%r\" %>s %D" combined
LogFormat "%{User-agent}i" agent
```

Il·lustració 128 - Prevenció de saturació de logs de Apache2

Des de fora ha preparat un atac de denegació de servei (DDoS) a Apache. Apache té eines per evitar ser susceptible a aquests atacs. Descriu quina llògica segueix per evitar que tombin el servidor i com es carrega.

Una de las herramientas que nos ofrece apache es un módulo llamado `mod_evasive`, el cual nos permite trazar la ruta origen de la petición y, en caso de que supere el número permitido de peticiones, bloquear dicho origen.

Hay más herramientas o estrategias configurables desde `sysctl.conf` pero nos centramos en el módulo de apache.

Para poder cargar la herramienta en nuestro servidor, deberemos realizar un proceso de instalación normal con el snippet `apt-get`. Sin duda, esta será la opción más sencilla, también se puede incluir a mano en el directorio correspondiente a los módulos.

```
# apt-get install libapache2-mod-evasive
```

```
root@master:~# apt-get install libapache2-mod-evasive
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  bsd-mailx exim4-base exim4-config exim4-daemon-light liblockfile1
Suggested packages:
  eximon4 exim4-doc-html exim4-doc-info spf-tools-perl swaks
Recommended packages:
  mailx
The following NEW packages will be installed:
  bsd-mailx exim4-base exim4-config exim4-daemon-light libapache2-mod-evasive
  liblockfile1
0 upgraded, 6 newly installed, 0 to remove and 92 not upgraded.
Need to get 2,296 kB of archives.
After this operation, 4,157 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

Il·lustració 129 - Instalando mod_evasive

Ahora necesitamos crear los directorios para los logs que genere este módulo.

```
# mkdir -p /var/log/apache2/evasive
# chown -R www-data:root /var/log/apache2/evasive
```

```
root@master:~# mkdir -p /var/log/apache2/evasive
root@master:~# chown -R www-data:root /var/log/ap
apache2/ apt/
root@master:~# chown -R www-data:root /var/log/apache2/evasive/
root@master:~# _
```

Ilustración 130 - Creación directorio

Una vez creado el directorio y asignada correctamente la membresía, creamos la configuración en el directorio `/etc/apache2/mods-available`.

```
# vi /etc/apache2/mods-available/mod-evasive.load
```

```
LoadModule evasive20_module /usr/lib/apache2/modules/mod_evasive20.so

<IfModule mod_evasive20.c>
    DOSHashTableSize 3097
    DOSPageCount 2
    DOSSiteCount 50
    DOSPageInterval 5
    DOSSiteInterval 1
    DOSBlockingPeriod 10
    DOSLogDir "/var/log/apache2/evasive"
</IfModule>
```

Ilustración 131 - Configuración mod_evasive

Después de esto, reiniciamos apache y ya tendremos `mod_evasive` en funcionamiento.

Els TCP Wrappers ens permeten crear regles que embolcallen les operacions del protocol TCP. Els arxius `/etc/hosts.allow` i `/etc/hosts.deny` ens permeten definir regles de TCP Wrapper.

Deshabilitar tot tipus de connexions per a una IP concreta, i per al servei de SSH per a una altre IP.

```
# /etc/hosts.deny: list of hosts that are _not_ allowed to access the system.
# See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: some.host.name, .some.domain
#            ALL EXCEPT in.fingerd: other.host.name, .other.domain
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
# The PARANOID wildcard matches any host whose name does not match its
# address.
#
# You may wish to enable this to ensure any programs that don't
# validate looked up hostnames still leave understandable logs. In past
# versions of Debian this has been the default.
# ALL: PARANOID
ALL: 172.16.1.150
sshd: 172.16.1.90_
```

Ilustración 132 - Deny para todos los servicios sobre una IP y sshd para otra

No contestar missatges de ICMP.

Añadimos 1 a `icmp_echo_ignore_all`. Lo podemos hacer manualmente a través del fichero `/etc/sysctl.conf` o bien de la siguiente manera:

```
# echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

```
root@slave1:~# ping master.gm.org
PING master.gm.org (172.16.1.1) 56(84) bytes of data.
64 bytes from master.gm.org (172.16.1.1): icmp_seq=1 ttl=64 time=0.355 ms
64 bytes from master.gm.org (172.16.1.1): icmp_seq=2 ttl=64 time=0.374 ms
64 bytes from master.gm.org (172.16.1.1): icmp_seq=3 ttl=64 time=0.360 ms
64 bytes from master.gm.org (172.16.1.1): icmp_seq=4 ttl=64 time=0.330 ms
^C
--- master.gm.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.330/0.354/0.374/0.028 ms
root@slave1:~# _
```

Slave2 (Snap1) [Running]

LXTerminal

File Edit Tabs Help

```
root@master:~# echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
root@master:~# █
```

```
root@slave1:~# ping master.gm.org
PING master.gm.org (172.16.1.1) 56(84) bytes of data.
64 bytes from master.gm.org (172.16.1.1): icmp_seq=1 ttl=64 time=0.355 ms
64 bytes from master.gm.org (172.16.1.1): icmp_seq=2 ttl=64 time=0.374 ms
64 bytes from master.gm.org (172.16.1.1): icmp_seq=3 ttl=64 time=0.360 ms
64 bytes from master.gm.org (172.16.1.1): icmp_seq=4 ttl=64 time=0.330 ms
^C
--- master.gm.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.330/0.354/0.374/0.028 ms
root@slave1:~# ping master.gm.org
PING master.gm.org (172.16.1.1) 56(84) bytes of data.
```

Com que volem castigar durament els nostres usuaris bloquejarem twitter.com, facebook.com i youtube.com.

Bloquejarem el acceso a dichos sitios web usando el archivo hosts (/etc/hosts).

```
0.0.0.0    www.facebook.com
0.0.0.0    www.youtube.com
0.0.0.0    www.twitter.com
0.0.0.0    facebook.com
0.0.0.0    youtube.com
0.0.0.0    twitter.com
::0        www.facebook.com
::0        www.youtube.com
::0        www.twitter.com
::0        facebook.com
::0        youtube.com
::0        twitter.com
```

No volem que hi hagin més de tres connexions simultànies per SSH

Usamos el archivo de configuración de ssh (sshd_config). Le seteamos el parámetro

MaxSessions a 3 y ya tendremos el máximo de conexiones concurrentes parametrizado.

Evitar que algunas IPs concretas se conecten. Mirar rangos de IPs que se donen a un país o conjunt de països i no permetre l'accés.

Los podemos bloquear usando el htaccess, aunque sería más recomendable usar un módulo con una base de datos de geolocalización, como maxmind con geo2location.

Aún así, usamos htaccess.

En el archivo htaccess localizado en el DocumentRoot de apache, añadimos las siguientes líneas:

```
order allow,deny
allow from all
deny from 65.19.146.2 220.248.0.0/14
```

Esas líneas bloquean el tráfico procedente de China.

4.4 Ejercicio 3

El objetivo de este ejercicio es denegar un ataque DoS sobre nuestro sistema.

Instalaremos una herramienta llamada *slowhttptest* que nos permitirá hacer el ataque.

```
# apt-get install slowhttptest
```

```
root@master:~# apt-get install slowhttptest
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  slowhttptest
0 upgraded, 1 newly installed, 0 to remove and 92 not upgraded.
Need to get 27.6 kB of archives.
After this operation, 115 kB of additional disk space will be used.
Get:1 http://ftp.uk.debian.org/debian/ jessie/main slowhttptest amd64 1.6-1 [27.6 kB]
Fetched 27.6 kB in 0s (97.9 kB/s)
Selecting previously unselected package slowhttptest.
(Reading database ... 70737 files and directories currently installed.)
Preparing to unpack .../slowhttptest_1.6-1_amd64.deb ...
Unpacking slowhttptest (1.6-1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up slowhttptest (1.6-1) ...
```

Ilustración 133 - Instalación slowhttptest

Para ejecutar el ataque sobre la máquina local:

```
# /root/slow/slowhttptest -g
```

```

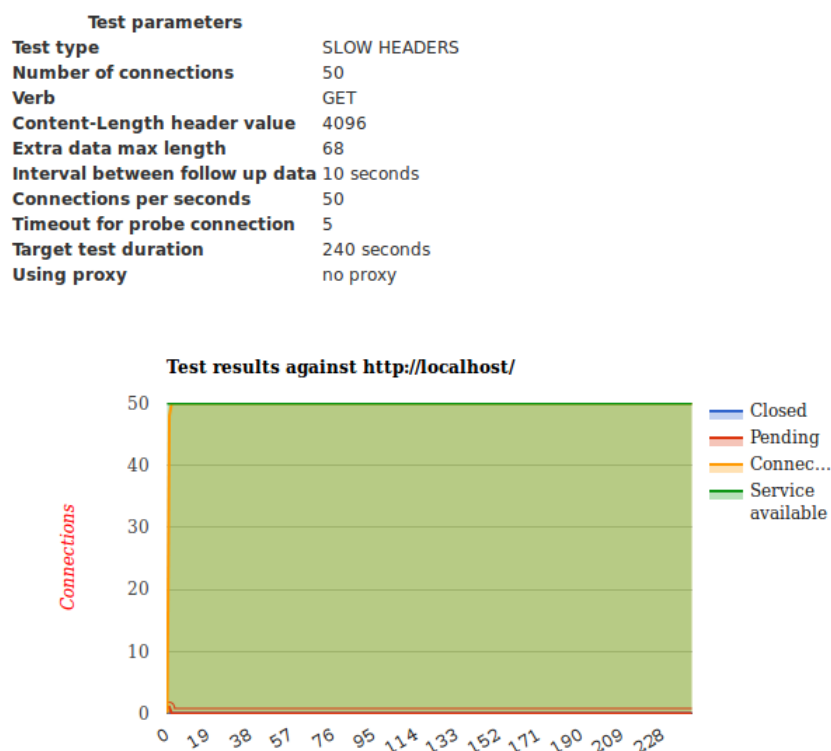
Wed Jan  4 13:58:40 2017:
  slowhttptest version 1.6
- https://code.google.com/p/slowhttptest/ -
test type:                SLOW HEADERS
number of connections:    50
URL:                      http://localhost/
verb:                     GET
Content-Length header value: 4096
follow up data max size:  68
interval between follow up data: 10 seconds
connections per seconds:  50
probe connection timeout: 5 seconds
test duration:            240 seconds
using proxy:              no proxy

Wed Jan  4 13:58:40 2017:
slow HTTP test status on 5th second:

initializing:             0
pending:                  0
connected:                50
error:                    0
closed:                   0
service available:        YES
  
```

Il·lustració 134 - Lanzando ataque slowhttptest

Y esta es la página que nos deja como estadística del ataque:



Il·lustració 135 - Estadística sobre master

```

URL: http://172.16.1.2/
verb: GET
Content-Length header value: 4096
follow up data max size: 68
interval between follow up data: 10 seconds
connections per seconds: 50
probe connection timeout: 5 seconds
test duration: 240 seconds
using proxy: no proxy

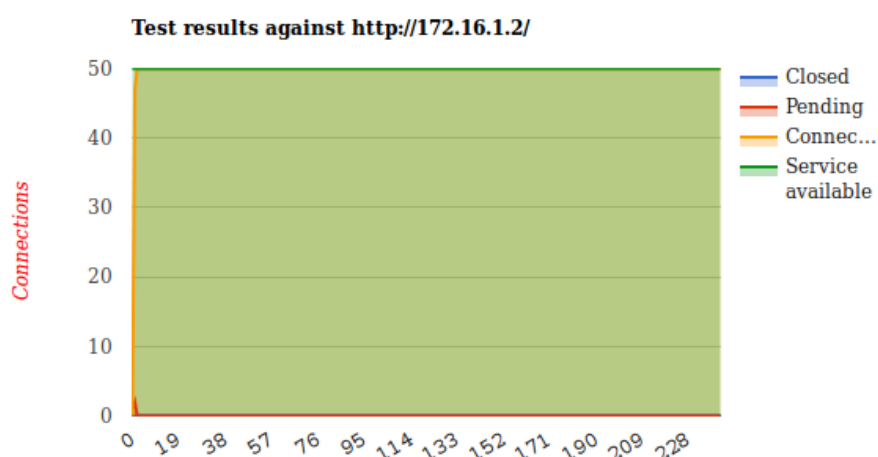
Wed Jan 4 14:26:59 2017:
slow HTTP test status on 240th second:

initializing: 0
pending: 0
connected: 50
error: 0
closed: 0
service available: YES

```

Il·lustració 136 - Lanzamos el ataque sin peticiones para obtener la página de estadística sobre slave1

Test parameters	
Test type	SLOW HEADERS
Number of connections	50
Verb	GET
Content-Length header value	4096
Extra data max length	68
Interval between follow up data	10 seconds
Connections per seconds	50
Timeout for probe connection	5
Target test duration	240 seconds
Using proxy	no proxy



Il·lustració 137 - Estadísticas ataque sin peticiones sobre slave1

Ahora realizamos el ataque sobre el apache2 de slave1.

```
# /root/slow/slowhttptest -g -c 10000 -u http://172.16.1.2 -l 1000
```

```
Wed Jan 4 14:32:13 2017:
    slowhttptest version 1.6
- https://code.google.com/p/slowhttptest/ -
test type:                SLOW HEADERS
number of connections:    10000
URL:                      http://172.16.1.2/
verb:                     GET
Content-Length header value: 4096
follow up data max size:  68
interval between follow up data: 10 seconds
connections per seconds:  50
probe connection timeout:  5 seconds
test duration:            1000 seconds
using proxy:              no proxy

Wed Jan 4 14:32:13 2017:
slow HTTP test status on 10th second:

initializing:              0
pending:                   173
connected:                 280
error:                     0
closed:                    0
service available:         NO
```

Ilustración 138 - Ataque con 10000 conexiones sobre slave1

Estadística después del ataque con 10000 conexiones a slave1:

Test parameters	
Test type	SLOW HEADERS
Number of connections	10000
Verb	GET
Content-Length header value	4096
Extra data max length	68
Interval between follow up data	10 seconds
Connections per seconds	50
Timeout for probe connection	5
Target test duration	1000 seconds
Using proxy	no proxy

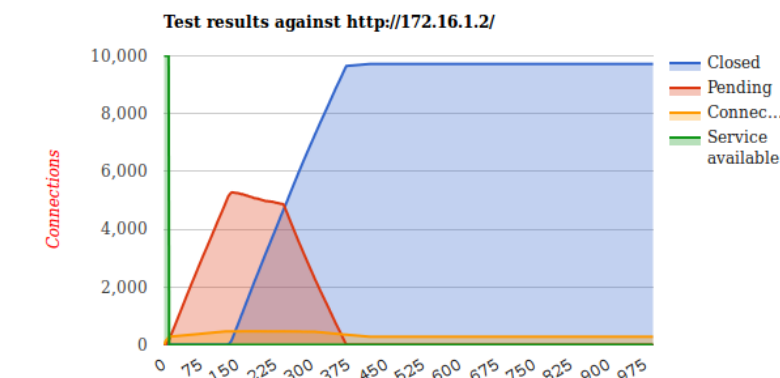


Ilustración 139 - Estadística ataque 10k conexiones a slave1

Si hacemos un *tcpdump* sobre slave1 mientras se está produciendo el ataque, podemos ver la procedencia de dicho ataque y tomar cartas en el asunto, como bloquear esa IP o bloquear ese tipo de tráfico.

Tal vez una estrategia más óptima sería aplicar un límite de conexiones en un tiempo determinado.

En este caso vemos el resultado del *tcpdump*:

```
14:56:18.618873 IP master.gm.org.45185 > slavela.gm.org.http: Flags [S], seq 1016250614, win 29200, options [mss 1460,sackOK,TS val 11721164 ecr 0,nop,wscale 7], length 0
14:56:18.627357 IP master.gm.org.45358 > slavela.gm.org.http: Flags [S], seq 3962709458, win 29200, options [mss 1460,sackOK,TS val 11721166 ecr 0,nop,wscale 7], length 0
14:56:18.633641 IP master.gm.org.45488 > slavela.gm.org.http: Flags [S], seq 74246474, win 29200, options [mss 1460,sackOK,TS val 11721167 ecr 0,nop,wscale 7], length 0
14:56:18.634473 IP master.gm.org.45445 > slavela.gm.org.http: Flags [S], seq 3758753355, win 29200, options [mss 1460,sackOK,TS val 11721168 ecr 0,nop,wscale 7], length 0
14:56:18.634509 IP master.gm.org.42797 > slavela.gm.org.http: Flags [S], seq 846867326, win 29200, options [mss 1460,sackOK,TS val 11721168 ecr 0,nop,wscale 7], length 0
```

Ilustración 140 - *Tcpdump* en slave1 durante el ataque de master

A continuación, también se muestra el *netstat*, para ver qué peticiones o qué conexiones se encuentran abiertas sobre nuestra máquina slave1.

```
tcp6      0      0 slavela.gm.org:http  master.gm.org:42197  ESTABLISHED
tcp6      0      0 slavela.gm.org:http  master.gm.org:42130  ESTABLISHED
tcp6    1408      0 slavela.gm.org:http  master.gm.org:42356  ESTABLISHED
tcp6      0      0 slavela.gm.org:http  master.gm.org:42160  ESTABLISHED
tcp6    1368      0 slavela.gm.org:http  master.gm.org:42271  ESTABLISHED
tcp6      0      0 slavela.gm.org:http  master.gm.org:42177  ESTABLISHED
tcp6      0      0 slavela.gm.org:http  master.gm.org:42180  ESTABLISHED
tcp6    1448      0 slavela.gm.org:http  master.gm.org:42378  ESTABLISHED
tcp6      0      0 slavela.gm.org:http  master.gm.org:42173  ESTABLISHED
```

Ilustración 141 - *Netstat* sobre slave1 durante el ataque de master

Con Ganglia vemos la crecida de paquetes recibidos por segundo durante los dos ataques de prueba que se han llevado a cabo sobre slave1.

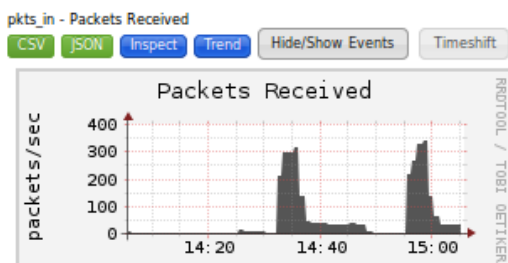


Ilustración 142 - Gráfico de red de slave1 durante los ataques.

Si no conociésemos el origen de dicha IP, podríamos indagar un poco más sobre esa, de ese modo podríamos determinar la procedencia y ver qué opciones son las más viables para mitigar el ataque.

4.5 Ejercicio 4

Se nos pide la instalación de medusa:

```
# apt-get install medusa
```

```
root@master:~# apt-get install medusa
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libpq5 libserf-1-1 libsvn1
The following NEW packages will be installed:
  libpq5 libserf-1-1 libsvn1 medusa
0 upgraded, 4 newly installed, 0 to remove and 92 not upgraded.
Need to get 1,399 kB of archives.
After this operation, 4,662 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 143 - Instalación de Medusa

Ahora debemos descargar el diccionario de contraseñas para poder ejecutar el ataque.

```
# wget http://downloads.skullsecurity.org/passwords/500-worst-
passwords.txt.bz2
```

```
root@master:/tmp# wget http://downloads.skullsecurity.org/passwords/500-worst-passw
--2017-01-04 13:11:00-- http://downloads.skullsecurity.org/passwords/500-worst-pas
Resolving downloads.skullsecurity.org (downloads.skullsecurity.org)... 192.155.81.8
c:91ff:fec8:b832
Connecting to downloads.skullsecurity.org (downloads.skullsecurity.org)|192.155.81.
ed.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://downloads.skullsecurity.org/passwords/500-worst-passwords.txt.bz2
--2017-01-04 13:11:01-- https://downloads.skullsecurity.org/passwords/500-worst-pa
Connecting to downloads.skullsecurity.org (downloads.skullsecurity.org)|192.155.81.
ted.
HTTP request sent, awaiting response... 200 OK
Length: 1868 (1.8K) [application/octet-stream]
Saving to: '500-worst-passwords.txt.bz2'

500-worst-passwords.txt. 100%[=====] 1.82K --.-K
2017-01-04 13:11:04 (25.0 MB/s) - '500-worst-passwords.txt.bz2' saved [1868/1868]
```

Ilustración 144 - Descargando diccionario de contraseñas

Nos movemos al Slave1 (B) y creamos un usuario *victim* con contraseña *hola*

```
# useradd victim -p hola
```

```
root@slave1:~# useradd victim -p hola
root@slave1:~# _
```

Ilustración 145 - Creación del usuario

Ahora ya podemos realizar el ataque de fuerza bruta.

```
# medusa -h 172.16.1.2 -P /tmp/500-worst-passwords.txt -u victim -M ssh
```

```
root@master:/tmp# medusa -h 172.16.1.2 -P /tmp/500-worst-passwords.txt -u victim -M ssh
Medusa v2.1.1 [http://www.fooofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: 123456 (1 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: password (2 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: 12345678 (3 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: 1234 (4 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: pussy (5 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: 12345 (6 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: dragon (7 of 500 complete)
```

Ilustración 146 - Realizando ataque

Con el objetivo de prevenir los ataques de medusa, se instala el software *fail2ban* y se realiza la configuración pertinente.

```
# apt-get install fail2ban
```

```
root@master:/tmp# apt-get install fail2ban
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  python-pyinotify
Suggested packages:
  python-gamin python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python-pyinotify
0 upgraded, 2 newly installed, 0 to remove and 92 not upgraded.
Need to get 192 kB of archives.
After this operation, 713 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 147 - Instalación fail2ban

A continuación, debemos configurar el archivo generado en */etc/fail2ban/jail.conf*. Desde ahí, podremos especificar los parámetros de la restricción de acceso.

Nos interesa modificar la configuración relativa a ssh para evitar el ataque con medusa:

```
# Optionally you may override any other parameter (e.g. banaction,
# action, port, logpath, etc) in that section within jail.local

[ssh]

enabled = true
port    = ssh
filter  = sshd
logpath = /var/log/auth.log
maxretry = 6

[ssh-blocklist]

enabled = true
filter  = sshd
action  = iptables[name=SSH, port=ssh, protocol=tcp]
        sendmail-whois[name=SSH, dest="%(destemail)s", sender="%(sender)s", s
endername="%(sendername)s"]
        blocklist_de[email="%(sender)s", apikey="xxxxxx", service="%(filter)s
"]
logpath = /var/log/sshd.log
maxretry = 10

[ssh-ddos]

enabled = true
port    = ssh
filter  = sshd-ddos
logpath = /var/log/auth.log
maxretry = 6

# Here we use blackhole routes for not requiring any additional kernel support
# to store large volumes of banned IPs

[ssh-route]

enabled = true
filter  = sshd
action  = route
logpath = /var/log/sshd.log
maxretry = 6
```

Ilustración 148 - Modificaciones fail2ban (jail.conf)

Lo habilitamos y guardamos. Por último, reiniciamos el servicio y ejecutamos el ataque de nuevo.

Mostramos el contenido de iptables de slave1 antes de volver a lanzar el ataque de medusa.

```
# iptables -L
```

```
root@slave1:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination            tcp dpt:ssh
fail2ban-SSH tcp -- anywhere             anywhere              multiport dports ssh
fail2ban-ssh-ddos tcp -- anywhere             anywhere              multiport dports ssh
fail2ban-dropbear tcp -- anywhere             anywhere              multiport dports ssh
fail2ban-ssh tcp -- anywhere             anywhere              multiport dports ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain fail2ban-SSH (1 references)
target     prot opt source                destination
RETURN     all -- anywhere             anywhere

Chain fail2ban-dropbear (1 references)
target     prot opt source                destination
RETURN     all -- anywhere             anywhere

Chain fail2ban-ssh (1 references)
target     prot opt source                destination
RETURN     all -- anywhere             anywhere

Chain fail2ban-ssh-ddos (1 references)
target     prot opt source                destination
RETURN     all -- anywhere             anywhere
```

Il·lustració 149 - Estado iptables antes del ataque

Relanzamos el ataque medusa sobre slave1 desde master.

```
# medusa -h 172.16.1.2 -P /tmp/500-worst-passwords.txt -u victim -M ssh
ssh
```

```
root@master:~# medusa -h 172.16.1.2 -P /tmp/500-worst-passwords.txt -u victim -M ssh
Medusa v2.1.1 [http://www.fooofus.net] (C) JoMo-Kun / Foofus Networks <jmk@fooofus.net>

ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: 123456 (1 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: password (2 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: 12345678 (3 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: 1234 (4 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: pussy (5 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: 12345 (6 of 500 complete)
ACCOUNT CHECK: [ssh] Host: 172.16.1.2 (1 of 1, 0 complete) User: victim (1 of 1, 0 complete) Password: dragon (7 of 500 complete)
NOTICE: ssh.mod: failed to connect, port 22 was not open on 172.16.1.2
```

Il·lustració 150 - Ataque Medusa

Vemos que la conexión se interrumpe.

Ahora volvemos a verificar las cadenas de iptables de slave1.

```
# iptables -L
```

```
root@slave1:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination          tcp dpt:ssh
fail2ban-SSH tcp -- anywhere             anywhere             multiport dports ssh
fail2ban-ssh-ddos tcp -- anywhere             anywhere             multiport dports ssh
fail2ban-dropbear tcp -- anywhere             anywhere             multiport dports ssh
fail2ban-ssh tcp -- anywhere             anywhere             multiport dports ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain fail2ban-SSH (1 references)
target     prot opt source                destination
RETURN     all -- anywhere             anywhere

Chain fail2ban-dropbear (1 references)
target     prot opt source                destination
RETURN     all -- anywhere             anywhere

Chain fail2ban-ssh (1 references)
target     prot opt source                destination
REJECT     all -- master.gm.org      anywhere             reject-with icmp-port-unreach
able
RETURN     all -- anywhere             anywhere

Chain fail2ban-ssh-ddos (1 references)
target     prot opt source                destination
RETURN     all -- anywhere             anywhere
```

Ilustración 151 - Master bloqueado

Aquí ya apreciamos que *fail2ban* ha añadido una regla de para denegar el login a master, que era el que había lanzado el ataque medusa.

Nota:

Las configuraciones de seguridad han sido seteadas sobre el master, por si se quisiera realizar alguna prueba.

5. Quinto Informe – Monitorización

5.1 Introducción

5.2 Nagios

5.2.1 Instalación sobre *master*

Para instalar correctamente nagios y todos sus correspondientes paquetes, hay una serie de dependencias que deben ser resueltas.

Por lo tanto, antes de instalar los paquetes de Nagios Core, se procede a la instalación de dichas dependencias.

Para ello usamos el siguiente snippet:

```
# apt-get install build-essential
```

```
root@master:~# hostname
master.gm.org
root@master:~# apt-get install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  dpkg-dev g++ g++-4.9 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libdpkg-perl libfile-fcntllock-perl
  libstdc++-4.9-dev
Suggested packages:
  debian-keyring g++-multilib g++-4.9-multilib gcc-4.9-doc
  libstdc++6-4.9-dbg libstdc++-4.9-doc
The following NEW packages will be installed:
  build-essential dpkg-dev g++ g++-4.9 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libdpkg-perl
  libfile-fcntllock-perl libstdc++-4.9-dev
0 upgraded, 10 newly installed, 0 to remove and 85 not upgraded.
Need to get 25.2 MB of archives.
After this operation, 50.4 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 152 - Instalación de dependencias

```
# apt-get -y install libapache2-mod-php5
```

```
root@master:~# apt-get install libapache2-mod-php5
```

Ilustración 153 - Instalación de dependencias

```
# apt-get -y install libgd2-xpm-dev
```

```
root@master:~# apt-get install libgd2-xpm-dev
```

Ilustración 154 - Instalación de dependencias

Ahora que ya disponemos de todas las dependencias de Nagios, se procede a crear el usuario Nagios.

```
# useradd nagios
```

```
root@master:~# useradd nagios
root@master:~#
```

Ilustración 155 - creación usuario nagios

Y se le asigna una contraseña.

```
# passwd nagios
```

```
root@master:~# passwd nagios
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@master:~# █
```

Ilustración 156 - Asignación contraseña usuario nagios

Ahora añadimos el usuario Nagios al grupo creado específicamente para nagios.

```
# usermod -G nagios nagios
```

```
root@master:~# usermod -G nagios nagios
root@master:~# █
```

Ilustración 157 - Añadiendo usuario nagios al grupo nagios

Por último, creamos el grupo para alojar la ejecución de los comandos de la interfaz web.

```
# groupadd nagcmd
```

```
root@master:~# groupadd nagcmd
root@master:~# █
```

Ilustración 158 - Creación grupo nagcmd

Añadimos el usuario en el grupo nagios.

```
# usermod -a -G nagcmd nagios
```

```
root@master:~# usermod -a -G nagcmd nagios
root@master:~# █
```

Ilustración 159 - Asignación nagcmd a nagios

Y también en el grupo apache.

```
# usermod -a -G nagcmd www-data
```

```
root@master:~# usermod -a -G nagcmd www-data
root@master:~# █
```

Ilustración 160 - Asignación nagcmd a www-data

Una vez tenemos todo el sistema preparado para la instalación de Nagios Core, empezamos a descargar el software y los plugins del mismo.

Dicha descarga la realizamos en el directorio */tmp*.

Para la descarga, ejecutamos el siguiente comando:

```
# wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.2.4.tar.gz#_ga=1.13317879.1719724961.1481384046
```

```
root@master:/tmp# wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.2.4.tar.gz#_ga=1.13317879.1719724961.1481384046
--2016-12-10 15:43:02-- https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.2.4.tar.gz
Resolving assets.nagios.com (assets.nagios.com)... 72.14.181.71, 2600:3c00::f03c:91ff:fedf:b821
Connecting to assets.nagios.com (assets.nagios.com)|72.14.181.71|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11088206 (11M) [application/x-gzip]
Saving to: 'nagios-4.2.4.tar.gz'

nagios-4.2.4.tar.g 100%[=====>] 10.57M 1.58MB/s in 7.3s

2016-12-10 15:43:11 (1.44 MB/s) - 'nagios-4.2.4.tar.gz' saved [11088206/11088206]
```

Ilustración 161 - Descarga Nagios

De la misma manera, descargamos los plugins:

```
# wget https://nagios-plugins.org/download/nagios-plugins-2.1.4.tar.gz
```

```
root@master:/tmp# wget https://nagios-plugins.org/download/nagios-plugins-2.1.4.tar.gz
--2016-12-10 15:45:08-- https://nagios-plugins.org/download/nagios-plugins-2.1.4.tar.gz
Resolving nagios-plugins.org (nagios-plugins.org)... 72.14.186.43
Connecting to nagios-plugins.org (nagios-plugins.org)|72.14.186.43|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2721216 (2.6M) [application/x-gzip]
Saving to: 'nagios-plugins-2.1.4.tar.gz'

nagios-plugins-2.1 100%[=====>] 2.59M 1.09MB/s in 2.4s

2016-12-10 15:45:11 (1.09 MB/s) - 'nagios-plugins-2.1.4.tar.gz' saved [2721216/2721216]

root@master:/tmp#
```

Ilustración 162 - Descarga Plugins Nagios

Centrándonos en los paquetes de Nagios, empezamos descomprimiendo sus archivos y los compilamos.

```
# tar -xzf nagios-4.2.4.tar.gz
```

```
root@master:/tmp# tar -xzf nagios-4.2.4.tar.gz
root@master:/tmp# ll
total 14M
drwxrwxr-x 18 root root 4.0K Dec  7 16:31 nagios-4.2.4
-rw-r--r--  1 root root 11M Dec  7 16:34 nagios-4.2.4.tar.gz
-rw-r--r--  1 root root 2.6M Nov 17 17:25 nagios-plugins-2.1.4.tar.gz
root@master:/tmp#
```

Ilustración 163 - Descompresión y listado de Nagios

Antes de realizar la compilación, pasamos el script de configuración.

```
# ./configure --with-command-group=nagcmd
```

Cuyo output es el siguiente:

```
*** Configuration summary for nagios 4.2.4 12-07-2016 ***:

General Options:
-----
Nagios executable: nagios
Nagios user/group: nagios,nagios
Command user/group: nagios,nagcmd
Event Broker: yes
Install ${prefix}: /usr/local/nagios
Install ${includedir}: /usr/local/nagios/include/nagios
Lock file: ${prefix}/var/nagios.lock
Check result directory: ${prefix}/var/spool/checkresults
Init directory: /etc/init.d
Apache conf.d directory: /etc/apache2/sites-available
Mail program: /bin/mail
Host OS: linux-gnu
IOBroker Method: epoll

Web Interface Options:
-----
HTML URL: http://localhost/nagios/
CGI URL: http://localhost/nagios/cgi-bin/
Traceroute (used by WAP): /usr/sbin/traceroute

Review the options above for accuracy. If they look okay,
type 'make all' to compile the main program and CGIs.
```

Ilustración 164 - Output compilación Nagios

Ahora ya estamos listos para proceder a la compilación de Nagios.

```
# make all
```

```
root@master:/tmp/nagios-4.2.4# make all
cd ./base && make
make[1]: Entering directory '/tmp/nagios-4.2.4/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmodes.o nebmodes.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o ../c
ommon/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nard.o nard.c
```

Ilustración 165 - Proceso de compilación...

```
*** Compile finished ***

If the main program and CGIs compiled without any errors, you
can continue with installing Nagios as follows (type 'make'
without any arguments for a list of all possible options):
```

Il·lustració 166 - Compilació finalizada

Una vez tenemos Nagios compilado, para verificar su correcto funcionamiento, instalamos los scripts de inicio y de ejemplo.

```
# make install
```

```
root@master:/tmp/nagios-4.2.4# make install
```

Il·lustració 167 - Make install

```
# make install-init
```

```
root@master:/tmp/nagios-4.2.4# make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/init.d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/init.d/nagios
*** Init script installed ***
```

Il·lustració 168 - Make install-init

```
# make install-config
```

```
root@master:/tmp/nagios-4.2.4# make install-config
```

Il·lustració 169 - Make install-config

```
# make install-commandmode
```

```
root@master:/tmp/nagios-4.2.4# make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/rw
chmod g+s /usr/local/nagios/var/rw
*** External command directory configured ***
```

Il·lustració 170 - Make install-commandmode

Por último, con el objetivo de hacer visible Nagios a través del navegador, ejecutamos el siguiente comando que instalará el panel frontal web.

```
# make install-webconf
```

```
root@master:/tmp/nagios-4.2.4# make install-webconf
/usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-avail
able/nagios.conf
if [ 1 -eq 1 ]; then \
    ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-en
abled/nagios.conf; \
fi
*** Nagios/Apache conf file installed ***
root@master:/tmp/nagios-4.2.4#
```

Il·lustració 171 - Instalando panel web Nagios

Y asignamos un usuario para que pueda acceder al panel web de nagios. Usamos autenticación de archivo de apache2.

```
# htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

```
root@master:/tmp/nagios-4.2.4# htpasswd -c /usr/local/nagios/etc/htpasswd.us
ers nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
root@master:/tmp/nagios-4.2.4#
```

Ilustración 172 - Añadimos password a nagiosadmin

```
root@master:/tmp/nagios-4.2.4# cp /etc/init.d/skeleton /etc/init.d/nagios
root@master:/tmp/nagios-4.2.4# vi /etc/init.d/nagios
```

Ilustración 173 - Copiando esqueleto Unit

```
DESC="Nagios"
NAME=nagios
DAEMON=/usr/local/nagios/bin/$NAME
DAEMON_ARGS="-d /usr/local/nagios/etc/nagios.cfg"
PIDFILE=/usr/local/nagios/var/$NA
```

Ilustración 174 - Contenido Unit Nagios

Y accedemos al panel web.

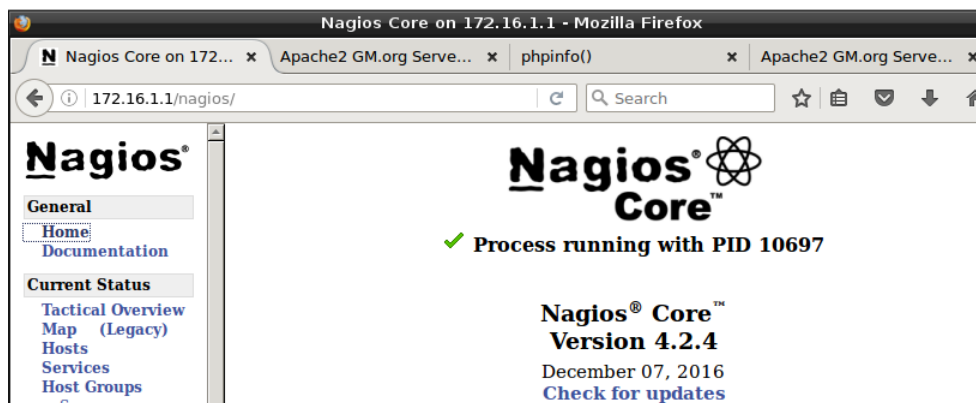


Ilustración 175 - Panel web Nagios Core

Una vez tenemos Nagios instalado y operativo, se procede a la instalación de los plugins y comandos para poder monitorizar los distintos servicios de los nodos seleccionados.

Para ello, nos dirigimos al directorio donde descargamos los plugins de nagios (/tmp) y descomprimos el paquete.

```
# tar -xzf nagios-plugins-2.1.4.tar.gz
```

```
root@master:/tmp# tar -xzf nagios-plugins-2.1.4.tar.gz
root@master:/tmp# ll
total 14M
drwxrwxr-x 18 root root 4.0K Dec 10 15:50 nagios-4.2.4
-rw-r--r--  1 root root 11M Dec  7 16:34 nagios-4.2.4.tar.gz
drwxr-xr-x 15 root root 4.0K Nov 17 17:25 nagios-plugins-2.1.4
-rw-r--r--  1 root root 2.6M Nov 17 17:25 nagios-plugins-2.1.4.tar.gz
root@master:/tmp#
```

Il·lustració 176 - Descompresió nagios plugins y muestra del contenido de /tmp

Accedemos al directorio recién descomprimido y ejecutamos los comandos de configuración e instalación de los binarios.

```
# ./configure --with-nagios-user=nagios --with-nagios-group=nagios
```

```
config.status: config.h is unchanged
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
root@master:/tmp/nagios-plugins-2.1.4# hostname
master.gm.org
root@master:/tmp/nagios-plugins-2.1.4#
```

Il·lustració 177 - Output configuració + hostname identificando autoría

Una vez configurado, ejecutamos los scripts de instalación.

```
# make
```

```
Making all in po
make[2]: Entering directory '/tmp/nagios-plugins-2.1.4/po'
make[2]: Nothing to be done for 'all'.
make[2]: Leaving directory '/tmp/nagios-plugins-2.1.4/po'
make[2]: Entering directory '/tmp/nagios-plugins-2.1.4'
make[2]: Leaving directory '/tmp/nagios-plugins-2.1.4'
make[1]: Leaving directory '/tmp/nagios-plugins-2.1.4'
root@master:/tmp/nagios-plugins-2.1.4#
```

Il·lustració 178 - Compilación con nuestra configuración

```
# make install
```

```
make[1]: Leaving directory '/tmp/nagios-plugins-2.1.4/po'
make[1]: Entering directory '/tmp/nagios-plugins-2.1.4'
make[2]: Entering directory '/tmp/nagios-plugins-2.1.4'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/tmp/nagios-plugins-2.1.4'
make[1]: Leaving directory '/tmp/nagios-plugins-2.1.4'
root@master:/tmp/nagios-plugins-2.1.4#
```

Il·lustració 179 - Proceso de instalación

Una vez alcanzado este punto, ya tenemos monitorizando nuestro propio servidor nagios (localhost), por lo que si nos dirigimos al navegador, podremos ver los resultados de la monitorización que viene configurada por defecto.

The screenshot shows the Nagios Core web interface at 172.16.1.1/nagios/. The interface includes a sidebar with navigation links like General, Current Status, Host Groups, and Problems. The main content area displays the 'Current Network Status' and 'Host Status Totals'. Below this, the 'Service Status Totals' are shown. The 'Service Status Details For Host 'localhost'' table lists various services and their status.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	12-11-2016 10:44:18	0d 0h 1m 20s	1/4	OK - load average: 0.32, 0.19, 0.07
	Current Users	OK	12-11-2016 10:41:49	0d 0h 3m 45s	1/4	USERS OK - 3 users currently logged in
	HTTP	OK	12-11-2016 10:42:27	0d 0h 3m 7s	1/4	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.000 second response time
	PING	OK	12-11-2016 10:43:42	0d 0h 1m 52s	1/4	PING OK - Packet loss = 0%, RTA = 0.07 ms
	Root Partition	OK	12-11-2016 10:43:42	0d 0h 1m 52s	1/4	DISK OK - free space: / 2009 MB (56% inode=72%);
	SSH	OK	12-11-2016 10:44:58	0d 0h 0m 36s	1/4	SSH OK - OpenSSH 6.7p1 Debian-5+deb8u3 (protocol 2.0)

Ilustración 180 - Monitor localhost

Por último, se efectúa la instalación de los paquetes NRPE (Nagios Remote Plugin Executor), gracias a estos binarios, se podrá monitorizar cualquier detalle del nodo objetivo. Básicamente permite la ejecución remota de los plugins de nagios.

Volvemos al directorio `/tmp` y descargamos los paquetes de NRPE.

```
# wget
https://github.com/NagiosEnterprises/nrpe/archive/3.0.1.tar.gz
```

```
root@master:~# cd /tmp/
root@master:/tmp# wget https://github.com/NagiosEnterprises/nrpe/archive/3.0.1.tar.gz
--2016-12-11 10:53:43-- https://github.com/NagiosEnterprises/nrpe/archive/3.0.1.tar.gz
Resolving github.com (github.com)... 192.30.253.113, 192.30.253.112
Connecting to github.com (github.com)|192.30.253.113|:443... ^[0F^[3-connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/NagiosEnterprises/nrpe/tar.gz/3.0.1 [following]
--2016-12-11 10:53:44-- https://codeload.github.com/NagiosEnterprises/nrpe/tar.gz/3.0.1
Resolving codeload.github.com (codeload.github.com)... 192.30.253.121, 192.30.253.120
Connecting to codeload.github.com (codeload.github.com)|192.30.253.121|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 514097 (502K) [application/x-gzip]
Saving to: '3.0.1.tar.gz'

3.0.1.tar.gz      100%[=====>] 502.05K   550KB/s   in 0.9s
2016-12-11 10:53:45 (550 KB/s) - '3.0.1.tar.gz' saved [514097/514097]
```

Ilustración 181 - Descarga NRPE

Descomprimos el paquete recién descargado.

```
# tar -xzf 3.0.1.tar.gz
```

```
root@master:/tmp# tar -xzf 3.0.1.tar.gz
```

Ilustración 182 - Descomprimos NRPE

```
root@master:/tmp# ll
total 508K
-rw-r--r--  1 root root 503K Dec 11 10:53 3.0.1.tar.gz
drwxrwxr-x 10 root root 4.0K Sep  8 17:18 nrpe-3.0.1
```

Ilustración 183 - Muestra del contenido

Antes de continuar, es necesario que instalemos una dependencia del paquete NRPE. Se trata de *libssl-dev*, que nos será requerida durante el proceso de configuración de NRPE.

Por lo tanto, con el comando `apt-get` procedemos a su instalación.

```
# apt-get install libssl-dev
```

```
root@master:/tmp/nrpe-3.0.1# apt-get install libssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libssl-doc libssl1.0.0 zlib1g-dev
The following NEW packages will be installed:
  libssl-dev libssl-doc zlib1g-dev
The following packages will be upgraded:
  libssl1.0.0
1 upgraded, 3 newly installed, 0 to remove and 84 not upgraded.
Need to get 3,704 kB of archives.
After this operation, 8,807 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 184 - Instalación libssl-dev

Una vez se ha realizado la instalación de la librería *libssl-dev*, volvemos con la configuración del plugin de nagios NRPE. En este caso, se deben asignar una serie de parámetros al script *configure* para que éste coja las opciones requeridas.

Por lo tanto, accedemos al directorio y ejecutamos el script de configuración.

```
# ./configure --with-ssl=/usr/bin/openssl --with-ssl-
lib=/usr/lib/x86_64-linux-gnu
```

```
*** Configuration summary for nrpe 3.0.1 09-08-2016 ***:

General Options:
-----
NRPE port:      5666
NRPE user:      nagios
NRPE group:     nagios
Nagios user:    nagios
Nagios group:   nagios

Review the options above for accuracy.  If they look okay,
type 'make all' to compile the NRPE daemon and client
or type 'make' to get a list of make options.

root@master:/tmp/nrpe-3.0.1#
```

Ilustración 185 - Resultado configuración

Una vez configurado compilamos todos los paquetes y, finalmente, ejecutamos el script de instalación.

```
# make all
```

```
make[1]: Leaving directory '/tmp/nrpe-3.0.1/src'

*** Compile finished ***

You can now continue with the installation or upgrade process.

Read the PDF documentation (NRPE.pdf) for information on the next
steps you should take to complete the installation or upgrade.

root@master:/tmp/nrpe-3.0.1#
```

Ilustración 186 - Proceso de compilación

```
# make      install-plugin
```

```
root@master:/tmp/nrpe-3.0.1# make install-plugin
cd ./src/; make install-plugin
make[1]: Entering directory '/tmp/nrpe-3.0.1/src'
/usr/bin/install -c -m 755 -d /usr/local/nagios/bin
/usr/bin/install -c -m 755 ../uninstall /usr/local/nagios/bin/nrpe-uninstall
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios check_nrpe /usr/local/nagios/
libexec
make[1]: Leaving directory '/tmp/nrpe-3.0.1/src'
root@master:/tmp/nrpe-3.0.1#
```

Ilustración 187 - Instalación NRPE finalizada

5.2.2 Instalación sobre los clientes

Llegados a este punto, tan sólo debemos repetir dos veces (una para cada host que queramos monitorizar) el procedimiento de instalación de los plugins de nagios y el plugin NRPE.

Para ahorrarnos tiempo de configuración ejecutando varios comandos y realizando las mismas operaciones dos veces, usaremos un paquete llamado *PSSH* que nos va a permitir ejecutar el mismo comando una serie de hosts seleccionados. *Se omite el proceso de instalación y también la explicación de la sintaxis de PSSH.*

Procedemos a crear los usuarios necesarios y les añadimos la contraseña.

```
# pssh -h nhosts.txt -l root -i "useradd nagios -p nagios"
```

```
root@master:/tmp# pssh -h nhosts.txt -l root -i "useradd nagios -p nagios"
[1] 11:22:09 [SUCCESS] 172.16.1.2:22
[2] 11:22:09 [SUCCESS] 172.16.2.3:22
root@master:/tmp#
```

Ilustración 188 - Creación de usuarios clientes

```
# pssh -h nhosts.txt -l root -i "cd /tmp | wget https://nagios-  
plugins.org/download/nagios-plugins-2.1.4.tar.gz"
```

```
2016-12-11 11:24:58 (646 KB/s) - 'nagios-plugins-2.1.4.tar.gz' saved [272121  
6/2721216]
```

Ilustración 189 - Output descarga

Descomprimos el fichero recién descargado.

```
# pssh -h nhosts.txt -l root -i "tar xzf nagios-plugins-  
2.1.4.tar.gz"
```

```
root@master:/tmp# pssh -h nhosts.txt -l root -i "tar xzf nagios-plugins-2.1.  
4.tar.gz"
[1] 11:27:36 [SUCCESS] 172.16.1.2:22
[2] 11:27:36 [SUCCESS] 172.16.2.3:22
root@master:/tmp#
```

Para no desarrollar ningún script y perder tiempo en esa tarea, accedemos, dejamos de lado un momento el comando *pssh* y realizamos las parametrizaciones pertinentes.

Añadimos la variable de entorno a ambos.

```
# export LD_FLAGS=-ldl
```

```
root@slave1:~# export LD_FLAGS=-ldl
```

Ilustración 190 - Adición variable entorno a ambos slaves

Ejecutamos el comando de configuración sobre ambos slaves.

```
# ./configure --with-nagios-user=nagios --with-nagios-group=nagios  
--enable-redhat-pthread-workaround
```

```
config.status: executing libtool commands  
config.status: executing po-directories commands  
config.status: creating po/POTFILES  
config.status: creating po/Makefile  
root@slave2:~/nagios-plugins-2.1.4#
```

Ilustración 191 - Configuración Slave2

```
config.status: executing libtool commands  
config.status: executing po-directories commands  
config.status: creating po/POTFILES  
config.status: creating po/Makefile  
root@slave1:~/nagios-plugins-2.1.4#
```

Ilustración 192 - Configuración slave1

Compilamos.

```
# make
```

```
make[2]: Leaving directory '/root/nagios-plugins-2.1.4/po'  
make[2]: Entering directory '/root/nagios-plugins-2.1.4'  
make[2]: Leaving directory '/root/nagios-plugins-2.1.4'  
make[1]: Leaving directory '/root/nagios-plugins-2.1.4'  
root@slave2:~/nagios-plugins-2.1.4#
```

Ilustración 193 - Output Make Slave2

```
make[2]: Nothing to be done for 'all'.  
make[2]: Leaving directory '/root/nagios-plugins-2.1.4/po'  
make[2]: Entering directory '/root/nagios-plugins-2.1.4'  
make[2]: Leaving directory '/root/nagios-plugins-2.1.4'  
make[1]: Leaving directory '/root/nagios-plugins-2.1.4'  
root@slave1:~/nagios-plugins-2.1.4#
```

Ilustración 194 - Output make slave1

E instalamos.

```
# make install
```

```
make[2]: Nothing to be done for 'install-exec-am'.  
make[2]: Nothing to be done for 'install-data-am'.  
make[2]: Leaving directory '/root/nagios-plugins-2.1.4'  
make[1]: Leaving directory '/root/nagios-plugins-2.1.4'  
root@slave2:~/nagios-plugins-2.1.4#
```

Ilustración 195 - Instalación slave2

```
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/root/nagios-plugins-2.1.4'
make[1]: Leaving directory '/root/nagios-plugins-2.1.4'
root@slave1:~/nagios-plugins-2.1.4#
```

Il·lustració 196 - Instal·lació slave1

Con esto ya tenemos los plugins de nagios instalados en nuestros slaves.

A continuació, debemos cambiar los permisos de los directorios sobre los que trabaja nagios en los clientes.

Para ello, volvemos a *pssh* y ejecutamos lo siguiente:

```
# pssh -h nhosts.txt -l root -i "chown nagios.nagios
/usr/local/nagios"
```

```
root@master:/tmp# pssh -h nhosts.txt -l root -i "chown nagios.nagios /usr/local/nagios"
[1] 11:48:03 [SUCCESS] 172.16.1.2:22
[2] 11:48:03 [SUCCESS] 172.16.2.3:22
root@master:/tmp#
```

Il·lustració 197 - Cambio de grupo

Verificamos el cambio.

```
root@master:/tmp# pssh -h nhosts.txt -l root -i "ls -lh /usr/local"
[1] 11:50:51 [SUCCESS] 172.16.1.2:22
total 36K
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 bin
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 etc
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 games
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 include
drwxrwsr-x 3 root  staff  4.0K Jun 21 13:57 lib
lrwxrwxrwx 1 root  staff    9 Jun 21 13:50 man -> share/man
drwxr-sr-x 5 nagios nagios 4.0K Dec 11 11:44 nagios
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 sbin
drwxrwsr-x 8 root  staff  4.0K Jun 21 13:58 share
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 src
[2] 11:50:51 [SUCCESS] 172.16.2.3:22
total 36K
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 bin
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 etc
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 games
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 include
drwxrwsr-x 3 root  staff  4.0K Jun 21 13:57 lib
lrwxrwxrwx 1 root  staff    9 Jun 21 13:50 man -> share/man
drwxr-sr-x 5 nagios nagios 4.0K Dec 11 11:42 nagios
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 sbin
drwxrwsr-x 8 root  staff  4.0K Jun 21 13:58 share
drwxrwsr-x 2 root  staff  4.0K Jun 21 13:50 src
root@master:/tmp#
```

Il·lustració 198 - Verificació grupu canbiado

Y, por último, aplicamos otro cambio de grupo sobre el directorio libexec y todos sus hijos.

```
# pssh -h nhosts.txt -l root -i "chown -R nagios.nagios /usr/local/nagios/libexec/"
```

```
root@master:/tmp# pssh -h nhosts.txt -l root -i "chown -R nagios.nagios /usr/local/nagios/libexec/"
[1] 11:53:17 [SUCCESS] 172.16.2.3:22
[2] 11:53:17 [SUCCESS] 172.16.1.2:22
root@master:/tmp#
```

Ilustración 199 - Cambio de grupo

Ahora que ya disponemos de todos los comandos relativos a nagios, debemos habilitar el acceso a las peticiones del servidor nagios (master). Para ello necesitamos el plugin NRPE.

Procedemos a descargarlo en ambos equipos.

```
# pssh -h nhosts.txt -l root -i "wget https://github.com/NagiosEnterprises/nrpe/archive/3.0.1.tar.gz"
```

```
Saving to: '3.0.1.tar.gz'
 0K ..... 183K
 50K ..... 402K
100K ..... 372K
150K ..... 630K
200K ..... 305K
250K ..... 388K
300K ..... 451K
350K ..... 533K
400K ..... 920K
450K ..... 446K
500K .. 26.2M=1.3
s
2016-12-11 13:56:50 (394 KB/s) - '3.0.1.tar.gz' saved [514097]
root@master:/tmp#
```

Ilustración 200 - Descarga NRPE en ambos slaves

Descomprimos los paquetes.

```
# pssh -h nhosts.txt -l root -i "tar -xzf 3.0.1.tar.gz"
```

```
root@master:/tmp# pssh -h nhosts.txt -l root -i "tar -xzf 3.0.1.tar.gz"
[1] 14:04:56 [SUCCESS] 172.16.2.3:22
[2] 14:04:56 [SUCCESS] 172.16.1.2:22
root@master:/tmp#
```

Ilustración 201 - Descompresión de NRPE en Slaves

Con el objetivo de reducir el tamaño del documento y evitar contenido redundante, sólo se muestra el proceso de instalación y configuración de NRPE en *Slave2*. El procedimiento es exactamente el mismo que se ha seguido en el caso de *Slave1*.

Para empezar con la configuración de NRPE debemos instalar las dependencias tal y como hicimos anteriormente con el servidor de nagios.

Instalamos los paquetes de libssl-dev.

```
# apt-get install libssl-dev
```

```
root@slave2:~/nrpe-3.0.1# apt-get install libssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libssl-doc libssl1.0.0 zlib1g-dev
The following NEW packages will be installed:
  libssl-dev libssl-doc zlib1g-dev
The following packages will be upgraded:
  libssl1.0.0
1 upgraded, 3 newly installed, 0 to remove and 78 not upgraded.
Need to get 3,704 kB of archives.
After this operation, 8,807 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 202 - Instalación libssl-dev

A continuación, debemos instalar xinetd, para hacerlo usamos el siguiente snippet:

```
# apt-get install xinetd
```

```
root@slave2:~/nrpe-3.0.1# apt-get install xinetd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  xinetd
0 upgraded, 1 newly installed, 0 to remove and 78 not upgraded.
Need to get 130 kB of archives.
After this operation, 338 kB of additional disk space will be used.
Get:1 http://ftp.uk.debian.org/debian/ jessie/main xinetd amd64 1:2.3.15-3 [130 kB]
Fetched 130 kB in 0s (281 kB/s)
Selecting previously unselected package xinetd.
(Reading database ... 65064 files and directories currently installed.)
Preparing to unpack .../xinetd_1%3a2.3.15-3_amd64.deb ...
Unpacking xinetd (1:2.3.15-3) ...
Processing triggers for man-db (2.7.0.2-5) ...
Processing triggers for systemd (215-17+deb8u4) ...
Setting up xinetd (1:2.3.15-3) ...
Processing triggers for systemd (215-17+deb8u4) ...
root@slave2:~/nrpe-3.0.1#
```

Ilustración 203 - Instalación xinetd

Ahora que ya tenemos las dependencias instaladas, podemos empezar con la configuración, compilación e instalación del plugin NRPE.

El primer paso que debemos hacer es ejecutar el script de configuración, para ello lanzamos el siguiente comando (dentro del directorio nrpe recientemente descomprimido):

```
# ./configure
```

```
*** Configuration summary for nrpe 3.0.1 09-08-2016 ***:

General Options:
-----
NRPE port:      5666
NRPE user:      nagios
NRPE group:     nagios
Nagios user:    nagios
Nagios group:   nagios

Review the options above for accuracy.  If they look okay,
type 'make all' to compile the NRPE daemon and client
or type 'make' to get a list of make options.

root@slave2:~/nrpe-3.0.1#
```

Ilustración 204 - Configuración NRPE

Compilamos los paquetes.

```
# make all
```

```
*** Compile finished ***

You can now continue with the installation or upgrade process.

Read the PDF documentation (NRPE.pdf) for information on the next
steps you should take to complete the installation or upgrade.

root@slave2:~/nrpe-3.0.1#
```

Ilustración 205 - Proceso de compilación

Lanzamos el comando de instalación del plugin.

```
# make install-plugin
```

```
root@slave2:~/nrpe-3.0.1# make install-plugin
cd ./src/; make install-plugin
make[1]: Entering directory '/root/nrpe-3.0.1/src'
/usr/bin/install -c -m 755 -d /usr/local/nagios/bin
/usr/bin/install -c -m 755 ../uninstall /usr/local/nagios/bin/nrpe-uninstall
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios check_nrpe /usr/local/nagios/libexec
make[1]: Leaving directory '/root/nrpe-3.0.1/src'
root@slave2:~/nrpe-3.0.1#
```

Ilustración 206 - Instalación del plugin

Lanzamos el comando de instalación del daemon.

```
# make install-daemon
```

```
root@slave2:~/nrpe-3.0.1# make install-daemon
cd ./src/; make install-daemon
make[1]: Entering directory '/root/nrpe-3.0.1/src'
/usr/bin/install -c -m 755 -d /usr/local/nagios/bin
/usr/bin/install -c -m 755 ../uninstall /usr/local/nagios/bin/nrpe-uninstall
/usr/bin/install -c -m 755 nrpe /usr/local/nagios/bin
/usr/bin/install -c -m 755 -o nagios -g nagios -d /usr/local/nagios/var
/usr/bin/install -c -m 755 -d /usr/lib/tmpfiles.d
/usr/bin/install -c -m 644 ../startup/tmpfile.conf /usr/lib/tmpfiles.d/nrpe.conf
make[1]: Leaving directory '/root/nrpe-3.0.1/src'
root@slave2:~/nrpe-3.0.1#
```

Ilustración 207 - Instalación daemon

Seguidamente lanzamos el comando que se encargará de crear el archivo de configuración.

```
# make install-config
```

```
root@slave2:~/nrpe-3.0.1# make install-config
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/etc
/usr/bin/install -c -m 644 -o nagios -g nagios sample-config/nrpe.cfg /usr/local/nagios/etc
root@slave2:~/nrpe-3.0.1#
```

Ilustración 208 - Instalación de los ficheros de configuración

Y por último, lanzamos el de systemd para insertar nrpe como servicio.

```
# make install-init
```

```
root@slave2:~/nrpe-3.0.1# make install-init
/usr/bin/install -c -m 644 startup/default-service /lib/systemd/system/nrpe.service
root@slave2:~/nrpe-3.0.1#
```

Ilustración 209 - Instalación como servicio nrpe

Con todo esto ya tenemos instalado el plugin NRPE, ahora queda realizar la configuración del daemon NRPE.

Se deben especificar los orígenes admitidos para realizar peticiones al puerto NRPE (5666) del equipo en cuestión. Para ello, modificamos el archivo localizado en */etc/xinetd.d/nrpe* añadiendo la IP de los equipos que tienen autorización para realizar peticiones.

```
# vi /etc/xinetd.d/nrpe
```

```
only_from = 127.0.0.1 172.16.1.1
```

Ilustración 210 - Equipos autorizados

Por último y para finalizar con la configuración del daemon, el servicio NRPE junto a su puerto, debe ser añadido al final del fichero `/etc/services`.

```
# vi /etc/services
```

```
# Local services
nrpe          5666/tcp          # NRPE
```

Ilustración 211 - Registro NRPE en el fichero `services`

5.2.3 Pruebas de monitorización

5.2.3.1 Client health

Ya desde la línea de comandos del master, procedemos a realizar la verificación del estado de los slaves, empezamos con el comando `check_ping`, el más básico. Se ejecuta sobre ambos clientes y vemos la respuesta satisfactoria.

```
root@master:~# hostname
master.gm.org
root@master:~# /usr/local/nagios/libexec/check_ping -H 172.16.2.3 -w 10,50% -c 2
0,100%
PING OK - Packet loss = 0%, RTA = 0.83 ms|rta=0.834000ms;10.000000;20.000000;0.0
00000 p1=0%;50;100;0
root@master:~# /usr/local/nagios/libexec/check_ping -H 172.16.1.2 -w 10,50% -c 2
0,100%
PING OK - Packet loss = 0%, RTA = 0.64 ms|rta=0.645000ms;10.000000;20.000000;0.0
00000 p1=0%;50;100;0
root@master:~# _
```

Ilustración 212 - `Check_ping` desde master a ambos slaves

5.2.3.2 Servicios SSH y Apache

```
root@master:~# /usr/local/nagios/libexec/check_ssh -p 22 172.16.2.3
SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0) | time=0.012174s;;;0.00000
0;10.000000
root@master:~# /usr/local/nagios/libexec/check_ssh -p 22 172.16.1.2
SSH OK - OpenSSH_6.7p1 Debian-5+deb8u3 (protocol 2.0) | time=0.010217s;;;0.00000
0;10.000000
root@master:~# _
```

Ilustración 213 - `Check_ssh` para ambos slaves

5.2.3.3 Servicios DHCP y NFS

Para el caso del servicio NFS, se debe descargar el fichero que se encargará de hacer las comprobaciones pertinentes del servicio. Una vez descargado sobre el master, se mueve el fichero a la ruta donde se encuentran todos los snippets de nagios (`/usr/local/nagios/libexec`).

```
root@master:/tmp/nagios# wget http://www.gbl-software.de/nagiosbinaries/check_nfs/check_nfs-src.tgz
--2017-01-02 10:44:02-- http://www.gbl-software.de/nagiosbinaries/check_nfs/check_nfs-src.tgz
Resolving www.gbl-software.de (www.gbl-software.de)... 85.25.44.187
Connecting to www.gbl-software.de (www.gbl-software.de)|85.25.44.187|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 59594 (58K) [application/x-gzip]
Saving to: 'check_nfs-src.tgz'

check_nfs-src.tgz      100%[=====>]  58.20K  --.-KB/s  in 0.1s

2017-01-02 10:44:03 (470 KB/s) - 'check_nfs-src.tgz' saved [59594/59594]

root@master:/tmp/nagios# ll
total 60K
-rw-r--r-- 1 root root 59K Jan 22  2013 check_nfs-src.tgz
root@master:/tmp/nagios# tar -xvf check_nfs-src.tgz
```

Ilustración 214 - Descarga y extracción de `check_nfs`

El otro método es utilizar NRPE para monitorizar el estado del servicio y es así como se explicará en la guía.

Lo primero que se debe modificar es el archivo local NRPE del cliente que se va a monitorizar. Dicho archivo se encuentra en `/usr/local/nagios/etc/nrpe.cfg`

Lo abrimos y añadimos las líneas que se especifican.

Líneas a añadir al final del archivo para *nfs-common* y para *dhcp*.

```
command[check_nfs]=/usr/local/nagios/libexec/check_procs -c 1:30 -C nfs-common
```

```
command[check_dhcp]=/usr/local/nagios/libexec/check_procs -c 1:30 -C isc-dhcp-server
```

```
# vi /usr/local/nagios/etc/nrpe.cfg
```

```
# Hardcoded commands, by hector@gm.org
command[check_nfs]=/usr/local/nagios/libexec/check_procs -c 1:30 -C nfs-common
```

Ilustración 215 - Check_nfs via NRPE

```
# Hardcoded snippets by hector@gm.org
command[check_dhcp]=/usr/local/nagios/libexec/check_procs -c 1:30 -C isc-dhcp-server
```

Ilustración 216 - Check_dhcp vía NRPE

Para poder hacer uso de este comando desde la web o desde la línea de comandos, es necesaria una modificación del archivo `commands` (`/usr/local/nagios/etc/objects/commands.cfg`) ya que por defecto no contiene ningún snippet relativo al NRPE, por lo tanto, accedemos al archivo y aplicamos las adiciones pertinentes.

```
# vi /usr/local/nagios/etc/objects/commands.cfg
```

```
#####
##          NRPE COMMANDS          ##
#####

define command{
    command_name check_nrpe
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

Ilustración 217 - Adición comando NRPE sobre master

Con esa adición no es suficiente, debemos especificar la definición de los servicios para que nagios cuando quiera ejecutar el comando, sepa donde debe ir a por el script.

Con la finalidad de tener todos los archivos algo más organizados, se procede a crear un archivo con el nombre *services.cfg* dentro de la ruta de nagios (*/usr/local/nagios/etc/objects/*) y se define el o los servicios a monitorizar.

```
# vi /usr/local/nagios/etc/objects/services.cfg
```

Añadimos lo siguiente:

```
define service{
    use                generic-service
    host_name          slave1
    service_description NFS Service
    check_command       check_nrpe!check_nfs
    check_period        24x7
    max_check_attempts 3
    is_volatile         0
    check_period        24x7
}

define service{
    use                generic-service
    host_name          master
    service_description DHCP Service
    check_command       check_nrpe!check_dhcp
    check_period        24x7
    max_check_attempts 3
    is_volatile         0
    check_period        24x7
}
```

Ilustración 218 - Definición de servicios

Listamos para verificar la existencia de *services.cfg*

```
root@master:/usr/local/nagios/etc/objects# ll
total 52K
-rw-rw-r-- 1 nagios nagios 7.8K Jan  2 16:23 commands.cfg
-rw-rw-r-- 1 nagios nagios 2.1K Dec 10 16:00 contacts.cfg
-rw-rw-r-- 1 nagios nagios 5.3K Dec 10 16:00 localhost.cfg
-rw-rw-r-- 1 nagios nagios 3.1K Dec 10 16:00 printer.cfg
-rw-r--r-- 1 root  nagios  538 Jan  2 16:31 services.cfg
-rw-rw-r-- 1 nagios nagios 3.2K Dec 10 16:00 switch.cfg
-rw-rw-r-- 1 nagios nagios 11K Dec 10 16:00 templates.cfg
-rw-rw-r-- 1 nagios nagios 3.2K Dec 10 16:00 timeperiods.cfg
-rw-rw-r-- 1 nagios nagios 3.9K Dec 10 16:00 windows.cfg
```

Ilustración 219 - Listado objetos directorio

Añadimos tal y como se dijo la ruta del fichero en *nagios.cfg*.

```
# vi /usr/local/nagios/etc/nagios.cfg
```

```
# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_file=/usr/local/nagios/etc/objects/services.cfg
```

Ilustración 220 - Especificación archivo *services.cfg* en *nagios.cfg*

Si reiniciamos el servicio de nagios después de todo lo que se realizado, nos va a fallar. Es necesario especificar los hosts para que sepa sobre qué IPs se deben ejecutar los comandos.

Para la definición de los nodos, será necesario establecer un hostgroup. De esta forma, nagios agrupará los equipos según hostgroup, cosa que nos facilitará el trabajo para localizar rápidamente los nodos en cuestión.

Para crear un hostgroup crearemos un nuevo archivo (*hostgroups.cfg*) dentro de objects.

```
# vi /usr/local/nagios/etc/objects/hostgroups.cfg
```

Y le añadimos el contenido mostrado:

```
define hostgroup {
    hostgroup_name linux
    alias          Srv-Linux
}
```

Ilustración 221 - Definición hostgroup Linux

Ahora añadimos este archivo a l fichero de configuración de nagios (*nagios.cfg*).

```
# vi /usr/local/nagios/etc/nagios.cfg
```

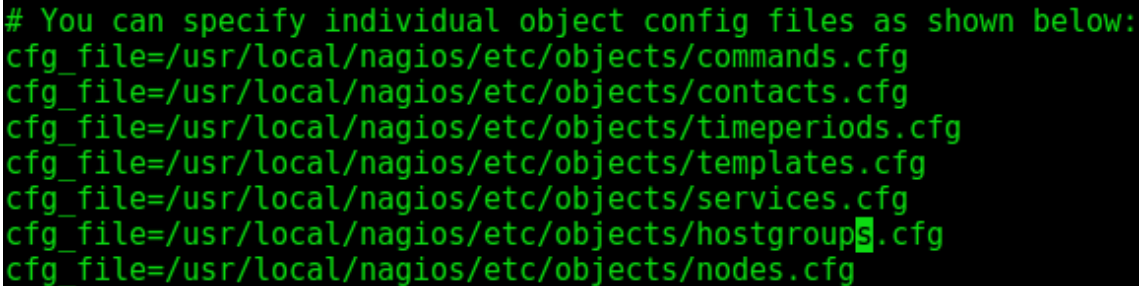
```
# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_file=/usr/local/nagios/etc/objects/services.cfg
cfg_file=/usr/local/nagios/etc/objects/hostgroups.cfg
```

Ilustración 222 - Adición *hostgroups.cfg* en *nagios.cfg*

De la misma forma que añadimos en nagios.cfg el archivo services.cfg, haremos lo mismo para los nodos.

```
# vi /usr/local/nagios/etc/nagios.cfg
```

Quedará de la siguiente forma:



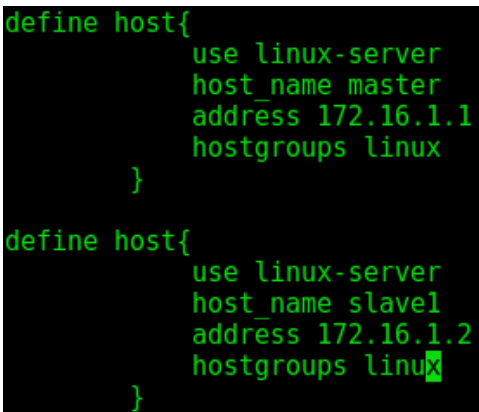
```
# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg
cfg_file=/usr/local/nagios/etc/objects/templates.cfg
cfg_file=/usr/local/nagios/etc/objects/services.cfg
cfg_file=/usr/local/nagios/etc/objects/hostgroups.cfg
cfg_file=/usr/local/nagios/etc/objects/nodes.cfg
```

Ilustración 223 - Adición nodes.cfg en nagios.cfg

Para ello modificamos el archivo . y añadimos los hosts que van a ser monitorizados.

```
# vi /usr/local/nagios/etc/objects/nodes.cfg
```

Y añadimos el contenido que se muestra en la imagen:



```
define host{
    use linux-server
    host_name master
    address 172.16.1.1
    hostgroups linux
}

define host{
    use linux-server
    host_name slave1
    address 172.16.1.2
    hostgroups linux
}
```

Ilustración 224 - Nodes.cfg

Debemos reiniciar el servicio nagios para que los cambios tengan efecto.

Nagios listo, funcionando y monitorizando los servicios.

```
# service nagios restart // systemctl restart nagios.service
```

Nos desplazamos al navegador y veremos que ya se han añadido los hosts

The screenshot shows the Nagios Core web interface. The top navigation bar includes 'General', 'Current Status', and 'Problems'. The 'Current Status' section displays 'Current Network Status' with a last update time of 'Wed Jan 4 15:21:48 GMT 2017'. Below this, there are 'Host Status Totals' and 'Service Status Totals' tables. The 'Host Status Totals' table shows 1 Up, 2 Down, 0 Unreachable, and 0 Pending. The 'Service Status Totals' table shows 7 OK, 0 Warning, and 0 Unknown. The 'Service Overview For All Host Groups' section shows a table with columns for Host, Status, Services, and Actions. The table lists two host groups: 'Srv-Linux (linux)' and 'Linux Servers (linux-servers)'. The 'Srv-Linux (linux)' group has two hosts: 'master' (DOWN) and 'slave1' (UP). The 'Linux Servers (linux-servers)' group has one host: 'localhost' (DOWN). The 'master' host has 1 PENDING service. The 'localhost' host has 7 OK and 1 CRITICAL service.

Ilustración 225 - Web panel de Nagios

Nota Importante:

En la práctica de seguridad se seteo que el servidor Master no aceptase ningún tipo de ping. Nagios por defecto, utiliza el comando ping para determinar si un host está activo o no, es por eso que sale como *down* pero realmente está UP.

5.3 Ganglia

5.3.1 Instalación sobre master y configuración del servidor

La instalación de los paquetes es exactamente igual que lo que hemos ido viendo hasta ahora. Usamos el snippet apt-get para realizar la instalación de los paquetes.

```
# apt-get install ganglia-monitor rrdtool gmetad ganglia-webfrontend
```

```
root@slave2:~# apt-get install ganglia-monitor rrdtool gmetad ganglia-webfrontend
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php5 libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libconfuse-common libconfuse0 libdbi1 libganglia1
  liblua5.1-0 libonig2 libqdbm14 librrd4 php5 php5-cli php5-common php5-gd php5-json php5-readline
  ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom php-pear php5-user-cache librrds-perl
  openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils ganglia-monitor ganglia-webfrontend gmetad
  libapache2-mod-php5 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libconfuse-common libconfuse0 libdbi1 libganglia1 liblua5.1-0 libonig2 libqdbm14 librrd4 php5
  php5-cli php5-common php5-gd php5-json php5-readline rrdtool ssl-cert
0 upgraded, 28 newly installed, 0 to remove and 78 not upgraded.
Need to get 11.7 MB of archives.
After this operation, 42.9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 226 - Instalación Ganglia sobre Master

Esto en cuanto a la instalación.

Procedemos a la configuración del nodo máster que será el que se encargará de recibir todos los datos de los distintos clientes conectados.

Inmediatamente después de finalizar la instalación, copiaremos la configuración de ganglia para apache y lo haremos sobre el directorio que corresponde.

```
# cp /etc/ganglia-webfrontend/apache.conf /etc/apache2/sites-enabled/ganglia.conf
```

```
root@master:~# cp /etc/ganglia-webfrontend/apache.conf /etc/apache2/sites-enabled/ganglia.conf
```

Seguidamente modificamos el archivo de configuración relativo al daemon meta de ganglia. Esta modificación nos permitirá nombrar nuestro clúster, especificar la frecuencia de actualización y qué host será el encargado de la recolección de datos.

Seteamos *my_cluster* como nombre del clúster, con una tasa de refresco de 60 segundo y que localhost sea el encargado de captar y procesar toda la información recibida, ya que es el master.

```
# vi /etc/ganglia/gmetad.conf
```

```
data_source "my_cluster" 60 localhost
```

Ilustración 227 - Data source ganglia

El siguiente paso consiste en modificar el archivo de configuración del daemon que envía información. Con el objetivo de obtener información del master, también se va a aplicar la configuración de cliente.

Especificamos el host al que se envían los datos, que en este caso se trata del mismo localhost, y se deshabilitan las opciones de multicast, ya que el servidor no envía datos a nadie.

```
#vi /etc/ganglia/gmond.conf
```

```
/* Feel free to specify as many udp_send_channels as you like. Gmond
   used to only support having a single channel */
udp_send_channel {
    #mcast_join = 239.2.11.71
    host = localhost
    port = 8649
    ttl = 1
}
```

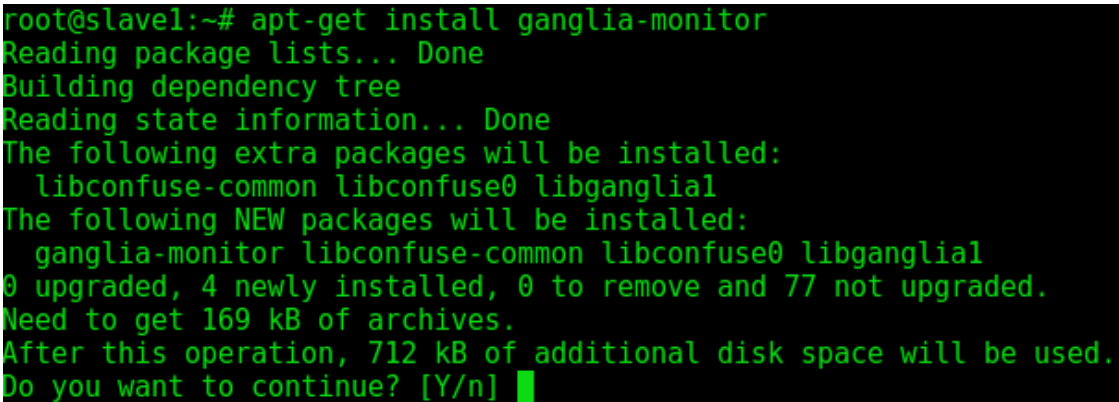
Ilustración 228 - Configuración gmond.conf en master

5.3.2 Configuración clientes

La configuración de los clientes es exactamente idéntica por lo que se muestra el procedimiento realizado sobre slave1.

Lo primero que se hace es instalar los paquetes del monitor ganglia.

```
# apt-get install ganglia-monitor
```



```
root@slave1:~# apt-get install ganglia-monitor
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libconfuse-common libconfuse0 libganglia1
The following NEW packages will be installed:
  ganglia-monitor libconfuse-common libconfuse0 libganglia1
0 upgraded, 4 newly installed, 0 to remove and 77 not upgraded.
Need to get 169 kB of archives.
After this operation, 712 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ilustración 229 - Instalación de ganglia-monitor en slaves

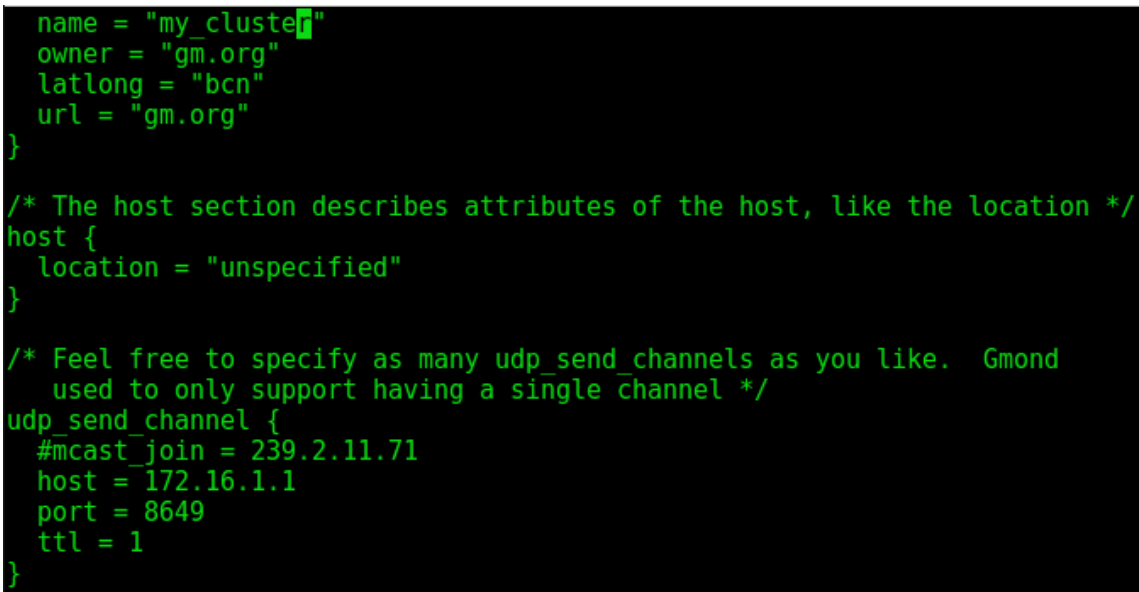
Una vez lo tenemos instalado, debemos modificar los ficheros del demonio de ganglia del host monitorizado.

Aquí se especificará el clúster al que pertenece el host (y al que reporta), además de la configuración relativa a cómo y a quién se envía la información.

En nuestro caso debemos deshabilitar la opción de envío multicast, ya que sólo se debe reportar al master y no a todos los nodos del clúster.

Por último, también se comentan las líneas de configuración relativas a la recepción de datos; el cliente no recibe de nadie, sólo envía.

```
# vi /etc/ganglia/gmond.conf
```



```
name = "my_cluster"
owner = "gm.org"
latlong = "bcn"
url = "gm.org"
}

/* The host section describes attributes of the host, like the location */
host {
  location = "unspecified"
}

/* Feel free to specify as many udp_send_channels as you like.  Gmond
   used to only support having a single channel */
udp_send_channel {
  #mcast_join = 239.2.11.71
  host = 172.16.1.1
  port = 8649
  ttl = 1
}
```

Ilustración 230 - Configuración gmond.conf en slaves (1)

```
/* You can specify as many udp_recv_channels as you like as well. */
#udp_recv_channel {
    #mcast_join = 239.2.11.71
    # port = 8649
    #bind = 239.2.11.71
#}
```

Il·lustració 231 - Configuració gmond.conf en slaves (2)

Una vez hecho este procedimiento en ambos slaves, reiniciamos los servicios implicados desde el master, también el servicio de monitorización del cliente.

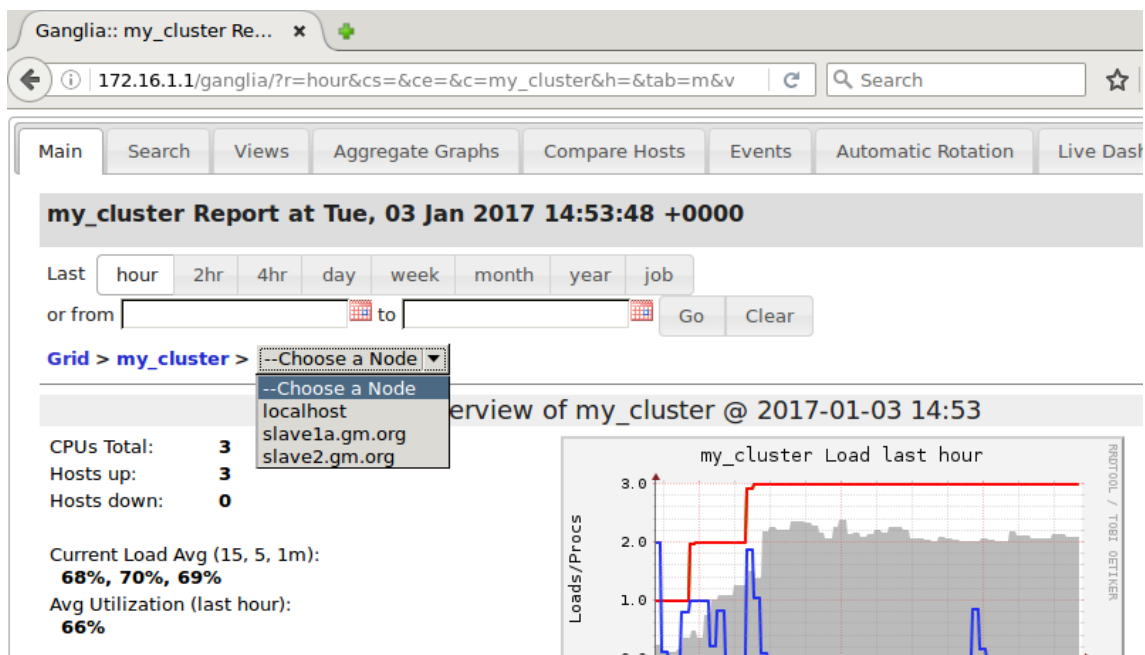
Clientes:

```
# service ganglia-monitor restart
```

Servidor:

```
# service ganglia-monitor restart && sudo service gmetad restart &&
sudo service apache2 restart
```

Con los servicios reiniciados, tan sólo nos quedará desplazarnos al navegador, introducir la IP del master seguido de `/ganglia` y accederemos al panel web con todos los datos relativos a la monitorización del clúster.



Il·lustració 232 - Web panel ganglia desde master. Listado de hosts disponibles

5.4 Comparativa y conclusiones

La principal diferencia entre ambos sistemas de monitorización es el objetivo o el tipo de máquina que va a ser monitorizada.

Generalmente, Ganglia se utiliza para monitorizar grids de cómputo, como puede ser clúster (o varios) de servidores web.

Por otro lado, Nagios se utiliza para monitorizar cualquier sistema y cualquier tipo de servicio o parámetro que sea necesario. Además, este sistema es capaz de enviarnos alertas personalizadas en función del estado de los sistemas/servicios.

Como conclusión se extrae que, de necesitar todos los nodos para realizar una misma operación, el sistema monitor que usaría para datos generales sería Ganglia, sin olvidar Nagios si quisiera ir en más detalle sobre alguno de esos nodos del clúster o grid.

Si de lo contrario todos los nodos realizan funciones distintas, iría directamente a Nagios.

