

Clonado de aulas con multicasting usando DRBL y Clonezilla Server

Índice de contenido

Prólogo.....	3
DRBL y Clonezilla Server.....	4
Escenario.....	5
Funcionamiento.....	5
Tareas Realizadas.....	5
Forma de trabajo.....	6
ANEXO I.....	8
INSTALACION DE DRBL-Clonezilla SE.....	8
INSTALACION SIN DHCP.....	9
ANEXO II.....	10
Creación de imágenes y Clonado de W7 y Linux.....	10
ANEXO III.....	11
Scripts De Tareas.....	11
ANEXO IV.....	21
Script de Creación de Reglas.....	21
ANEXO V.....	26
Ejecutando Scripts después de Clonar.....	26
Scripts para Parsear Sysprep.....	27

Copyright © 2010 David Fuertes Ortiz

Algunos derechos reservados. Se concede permiso para copiar, distribuir y/o modificar este documento bajo las condiciones de la licencia Reconocimiento-CompartirIgual 3.0 o, a su elección, cualquier versión posterior publicada por Creative Commons.

Puede consultar una copia de dicha licencia en
<http://creativecommons.org/licenses/by-sa/3.0/es/legalcode.es>

Prólogo

En este documento se intenta explicar de forma clara como utilizar DRBL/Clonezilla para herramienta de mantenimiento y despliegue de aulas de informática.

Con el fin de intentar que el documento sea lo mas claro posible, no se ha profundizado en el uso e instalación de DRBL/Clonezilla, ya que existe una amplia documentación al respecto, sino que he intentado centrarme en **como utilizar** esta herramienta para nuestros fines.

La idea es utilizar el fichero de configuración del servidor dhcp con los datos de nuestra red ya en funcionamiento y conseguir que el servidor DRBL/Clonezilla trabaje con el.

Por otro lado, Clonezilla permite trabajar en modo comando con listas de IP's por lo que se han creado scripts en Perl que ejecutan esos comandos pasando las listas de IP's en función del nombre del aula introducido.

Los detalles “técnicos” se pueden ver fácilmente en los scripts presentados en los anexos.

Espero que este trabajo sirva de ayuda para que cualquiera pueda adaptar DRBL/Clonezilla SE a las necesidades de su entorno.

Por último puesto que seguro, este trabajo es mejorable (sobre todo mis scripts en Perl) y no exento de errores, animo a todos aquellos que lo deseen a que lo completen y modifiquen a su antojo.

DRBL y Clonezilla Server

Se opta por el uso de DRBL-Clonezilla Server porque es gratuito y con licencia GPL. Implementa multicasting y permite clonar particiones por separado. Cosa que otras herramientas como FOG no permiten.

Los requerimientos mínimos en el cliente, según las pruebas efectuadas, es de 512MB de RAM.

DRBL (Diskless Remote Boot in Linux) Ver. 1.9.6-35:

Proyecto desarrollado por el NCHC's Free Software Lab (National Center for High-Performance Computing localizado en Hsinchu, Taiwan).

DRBL es un solución Open Source para el despliegue y gestión de sistemas operativos GNU/Linux sobre múltiples equipos.

DRBL va a proporcionar los servicios necesarios para que los clientes arranquen por red, utilizando el protocolo PXE. Mediante este protocolo se pasa al cliente el cargador de arranque desde el servidor DRBL.

Este cargador de arranque se crea de forma automatizada por DRBL en función de las tareas que se quieran realizar en los clientes.

Básicamente, el cargador de arranque lanzará un sistema Linux mínimo, que usará mediante NFS los sistemas de archivos `/tftpboot/node_root`, `/usr`, `/opt`, `/home` y `/var/spool/mail` del servidor DRBL (ver Anexo IV).

Clonezilla Server Ver. 2.3.7-41:

Utilidad Open Source para clonado y particionado de discos incluida en el proyecto DRBL. Será la herramienta utilizada para clonar.

Escenario

Se necesita clonar PC's en aulas de informática de hasta 27 equipos iguales. Cada equipo tendrá instalado un sistema Windows 7 y otro Linux.

Hasta la fecha se utilizaba la herramienta rembo toolkit, pero a pasado a ser deprecated y no soporta Windows 7 ni el sistema de archivos ext3 (sí el ext2 con journaling activado).

Se pretende aprovechar el DHCP ya configurado con la asignación IP-MAC ya realizada.

Funcionamiento

Se tendrá un servidor DRBL que permitirá a los clientes arrancar mediante PXE y lanzar un sistema Linux mínimo que utilizará Clonezilla para clonar particiones.

Los clientes montarán parte de su sistema de archivos, mediante NFS, desde el sistema de archivos del servidor DRBL. Así, el cliente tendrá acceso a las imágenes a utilizar y a los scripts propios que deseemos que ejecute dicho cliente (ver Anexo V).

Tareas Realizadas

Con el fin de poder usar DRBL para nuestros fines, se han realizado las siguientes tareas:

1. Se instala DRBL/Clonezilla y se configura en modo Clonezilla BOX.
2. Se desinstala el servidor DHCP.
3. Se crea un fichero ejecutable (scripts_sh) que llamará a nuestros futuros scripts de postclonado (con el fin de parsear sysprep.xml).
4. Se crea un fichero IP_HOST_TABLE a partir del fichero dhcpd.conf que utiliza clonezilla y los scripts creados.
5. Se introducen los equipos del dhcp en los ficheros de configuración del DRBL, NIS y NAS.
6. Se colocan todos los scripts así como el fichero de respuesta sysprep.xml dentro del directorio /home/partimag/ accesible por los clientes mediante NFS

Forma de trabajo

Existirán los siguientes componentes:

Servidor DRBL: Máquina con S.O Linux. En nuestro caso Debian lenny con DRBL/Clonezilla instalado (ver ANEXO I).

Servidor DHCP: Máquina con S.O Linux. En nuestro caso Debian lenny con dhcp3.

Clientes: PC's con arranque por red y protocolo PXE activado.

La forma de trabajo habitual consistirá en introducir las tareas en el servidor DRBL y arrancar el cliente.

Las tareas en el servidor se introducen por medio de unos scripts creados en perl (ver ANEXO III) que permiten trabajar con aulas (mediante grupos de IP's) y equipos individuales (mediante la IP correspondiente).

Estas tareas serán:

Alta o baja de equipos en las aulas.

Creación de imágenes nuevas a partir de equipo cliente.

Recarga de imágenes en equipos dañados.

Recarga de imágenes en aulas.

Arranque normal desde disco de los equipos.

Alta o baja de equipos en las aulas:

Copiar fichero dhcpd.conf del servidor DHCP al DRBL.

Ejecutar:

./CrearReglasEquipos.sh

Creación de imágenes nuevas a partir de equipo cliente:

./CrearImagenDisco.pl IP_Equipo_origen Aula

Recarga de imágenes en equipos dañados:

./RecargaDiscoIP.pl IP_Equipo Aula

./RecargaW7IP.pl IP_Equipo Aula

./RecargaLinuxIP.pl IP_Equipo Aula

Recarga de imágenes en aulas:

./RecargaDiscoSala.pl Aula

./RecargaW7Sala.pl Aula

./RecargaLinuxSala.pl Aula

Arranque normal desde disco de los equipos:

No es una tarea habitual ya que una vez que el cliente termina de realizar su tarea arrancara de forma normal desde disco. Sin embargo, es posible arrancar clonezilla en modo interactivo y decirle que se quiere que todos los equipos arranquen en local: `./dcs -i`

ANEXO I

INSTALACION DE DRBL-Clonezilla SE

1. Se añade al sources.list deb <http://drbl.sourceforge.net/drbl-core> drbl stable
2. Se instala mediante `/opt/drbl/sbin/drblsrv -i` (descarga de red todos los paquetes necesarios), con las siguientes opciones:
 - a) No instalar imagenes de inico.
 - b) No usar salida de consola.
 - c) Usar el mismo nivel de cpu que el servidor.**

Nota: En nuestro caso previamente hemos instalado un kernel (desde los backports de debian) con soporte para las tarjetas de red de nuestros equipos clientes. Así, es posible seleccionarlo durante el proceso de instalación.
 - d) No actualizar SO.
3. Se copia el scripts (scripts_sh, ver ANEXO V) que se ejecutara después de clonar en `/opt/drbl/share/ocs/posrun` . Se hace en este paso porque al configurar el drbl se introduce este script en las imágenes de arranque de los clientes.
4. Se configura mediante `/opt/drbl/sbin/drblpush -i` con las siguientes opciones:
 - a) DNS: el nuestro.
 - b) Dominio NIS: el nuestro.
 - c) Prefijo hostname cliente: **es indiferente ya que no lo usaremos.**
 - d) Puerto ethernet para acceso a INET: el nuestro.
 - e) No recompilar MAC de los clientes.**
 - f) Desea que el servicio DHCP ofrezca.... misma IP..... : No**
 - g) Número inicial a usar en IP's: 1 (**es indiferente ya que no lo usaremos**).
 - h) **Número de clientes DRBL: 1** (no lo usaremos y borraremos los datos creados durante esta configuración).
 - i) Servicio linux disless: 2 == NO**
 - j) Modo Clonezilla: 1 == Clonezilla BOX**
 - k) Directorio de imagenes por defecto: /home/partimag**
 - l) Configurar contraseña pxelinux NO
 - m) prompt de inicio Si
 - n) 10 ms
 - o) Modo texto
 - p) usar NAT NO**

5. Se modifica el dhcpd.conf cambiándolo por el que existe en nuestro servidor DHCP.
6. Se borran /etc/exports, /etc/drbl/IP_HOST_TABLE. Se quita el cliente añadido en /etc/ypserv.securenets.
7. Se introducen los clientes del dhcp.conf en /etc/exports, /etc/drbl/IP_HOST_TABLE y /etc/ypserv.securenets.
8. Se reinician servicios NFS, NIS y DHCP.

Los pasos 6, 7 y 8 se pueden realizar parseando el dhcp mediante un script que hemos llamado **CrearReglasEquipos.pl** (ver anexo IV).

INSTALACION SIN DHCP

En el servidor drbl:

1. Se instala drbl-Clonezilla como en los pasos anteriores 1 al 6.
2. Se desinstala el dhcp: apt-get remove dhcp3-server
Nota: no borrar el fichero /etc/dhcp3/dhcpd.conf por que Clonezilla utiliza la información IP-MAC para generar los ficheros PXE de arranque del cliente.
3. Se configuran los gestores de arranque PXE de los clientes para que no utilicen el parámetro server-name del servidor dhcp (útil para no tener que tocar la configuración del dhcp):
/opt/drbl/sbin/mknic-nbi -c n

En el servidor dhcp:

Configurar los parametros:

```
next-server    IP-servidorDRBL
filename       pxelinux.0
```

NOTA: Si no se ejecuta: /opt/drbl/sbin/mknic-nbi -c n, en el servidor DRBL, será necesario introducir el parámetro:

```
server-name    nombredadoalDRBL
```

El servidor dhcp deberá tener la opcion:

```
option host-name nombrecliente
```

Con esto se consigue que le cliente Linux tome el nombre del equipo del dhcp.

ANEXO II

Creación de imágenes y Clonado de W7 y Linux.

1. Preparación del equipo para nuestro entorno de trabajo:

En nuestras salas tendremos equipos con sistemas Windows7 y Linux.

El arranque dual se llevara a cargo por medio de grub (concretamente grub2).

Aunque no es la única posibilidad, se recomienda instalar W7 y luego Linux, para que realice la instalación del grub automáticamente.

2. Creación de imagen del equipo origen:

Habrà que crear una imagen de todo el disco (seleccionando todas las particiones) en el servidor para poder trabajar después con particiones independientes.

Windows7:

W7 crea una partición de 100 MB aprox. oculta y la partición de sistema, luego en nuestro caso W7 estará en las particiones /dev/sda1 (oculta) y /dev/sda2 ([c:](#)).

Una vez instalado el sistema o las aplicaciones deseadas se ejecuta el sysprep para que actúe sobre el fichero respuesta sysprep.xml (que sera parseado con la información del dhcp después de clonar).

```
sysprep.exe /generalize /oobe /shutdown /quiet /unattend:sysprep.xml
```

Luego la imagen generada queda lista para ejecutar sysprep en el primer arranque utilizando los valores del sysprep.xml parseado después de clonar.

Nota: Es necesario borrar usuario alumno antes de crear la imagen en servidor. Ya que durante el primer arranque se crea un usuario (proceso obligatorio) con dicho nombre.

Linux:

Se instalara la distribución deseada en el sistema de origen. En nuestro caso se usarán las particiones /dev/sda3 (para el sistema raiz) y /dev/sda4 (para swap).

Serà necesario modificar el /boot/grub/grub.cfg para que arranque por defecto W7 así como quitar el modo rescate para Linux.

El sistema Linux lo configuraremos de manera que los parámetros de red IP y hostname los pase el dhcp.

```
/etc/network/interfaces => para dhcp
```

```
/etc/hostname => vacio.
```

Hay que poner atención a si el kernel del cliente nombra los discos como hda o sda, puesto que por defecto clonezilla los nombra hda.

ANEXO III

Scripts De Tareas

CrearImagenDisco.pl

```
#!/usr/bin/perl

# Script para generar una nueva imagen del equipo dado en el servidor"

$IP_HOST_TABLE = "/etc/drbl/IP_HOST_TABLE";

if ( $ARGV[0] eq "--help " || $#ARGV == -1 )
{
    print "Uso: $0 IP_origen aula\n";
    print "Crea una nueva imagen, llamada aula-img, de la IP origen en el servidor DRBL\n";
    exit;
}

$equipo=$ARGV[0];
$aula=$ARGV[1];

open (IP_HOST, "<$IP_HOST_TABLE");

$existe="no";
while (<IP_HOST>)
{
    if (/ $equipo\s/) {
        $existe="si";
    }
}

close(IP_HOST);
```

```

if ( $existe eq "no" ) {
    print "El equipo especificado no es un cliente del DRBL\n";
    exit;
} else {
    system("/opt/drbl/sbin/drbl-ocs -b -q2 -j2 -nogui -p poweroff -z1 -i 1000000 -h \"$equipo\" -
l es_ES.UTF-8 startparts save $aula-img sda1 sda2 sda3");
}

```

RecargaDiscoIP.pl

```
#!/usr/bin/perl
```

```
# Script para clonar una imagen del disco completo en un equipo.
```

```
$IP_HOST_TABLE = "/etc/drbl/IP_HOST_TABLE";
```

```
$Dir_Img="/home/partimag/";
```

```
if ( $ARGV[0] eq "--help " || $#ARGV == -1 )
```

```

{
    print "Uso: $0 IP_Equipo Aula\n";
    print "Realiza una recarga completa del disco, creando las particiones necesarias, en el
equipo indicado.\n";
    exit;
}

```

```
$equipo=$ARGV[0];
```

```
$aula=$ARGV[1];
```

```

open (IP_HOST, "<$IP_HOST_TABLE");

$existe="no";
while (<IP_HOST>)
{
    if (/Sequipo\s/) {
        $existe="si";
    }

}

close(IP_HOST);
if ( $existe eq "no" ) {
    print "El equipo especificado no es un cliente del DRBL\n";
    exit;
} else {
    if (-d "$Dir_Img$aula-img" ) {
        system("/opt/drbl/sbin/drbl-ocs -b -g auto -e1 auto -e2 -x -nogui -r -j2 -o1 -p reboot --
clients-to-wait 1 --max-time-to-wait 300 -h \"$Sequipo\" -l es_ES.UTF-8 startparts multicast_restore
$aula-img sda1 sda2 sda3");
    } else {
        print "No existe imagen $Dir_Img$aula-img\n";
        exit;
    }
}
}

```

RecargaDiscoSala.pl

```
#!/usr/bin/perl
```

```
# Script para clonar una nueva imagen del disco completo en una sala
```

```

$IP_HOST_TABLE = "/etc/drbl/IP_HOST_TABLE";

if ( $ARGV[0] eq "--help " || $#ARGV == -1 )
{
    print "Uso: $0 aula\n";
    print "Realiza una recarga completa del disco. Creando las particiones necesarias, en el aula
indicada.\n";
    exit;
}
$aula=$ARGV[0];

# Vamos a crear la lista de IP`S de las sala

open (IP_HOST, "<$IP_HOST_TABLE");

while (<IP_HOST>)
{
    if (/^$aula/) {
        @linea = split(/\t/, $_);
        $IP = @linea[0];
        # print "$IP\n";
        push(@ListaIPs, $IP);
    }
}

close(IP_HOST);

if ( $#ListaIPs != -1 )
{
    system("/opt/drbl/sbin/drbl-ocs -b -g auto -e1 auto -e2 -x -nogui -r -j2 -o1 -p reboot --clients-
to-wait $#ListaIPs --max-time-to-wait 300 -h \"@ListaIPs\" -l es_ES.UTF-8 startparts
multicast_restore $aula-img sda1 sda2 sda3");
} else {
    print "El aula $aula, no pertenece a este servidor DRBL\n";
}

```

```
        exit;
    }

```

RecargaLinuxIP.pl

```
#!/usr/bin/perl
```

```
# Script para clonar una imagen del Linux en un equipo.
```

```
$IP_HOST_TABLE = "/etc/dtbl/IP_HOST_TABLE";
```

```
$Dir_Img="/home/partimag/";
```

```
if ( $ARGV[0] eq "--help " || $#ARGV == -1 )
```

```
{
    print "Uso: $0 IP_Equipo Aula\n";
    print "Realiza una recarga de la partición Linux en el equipo indicado.\n";
    exit;
}
```

```
$equipo=$ARGV[0];
```

```
$aula=$ARGV[1];
```

```
open (IP_HOST, "<$IP_HOST_TABLE");
```

```
$existe="no";
```

```
while (<IP_HOST>)
```

```
{
```

```

        if (/ $equipo\s/) {
            $existe="si";
        }

    }

close(IP_HOST);

if ( $existe eq "no" ) {
    print "El equipo especificado $equipo no es un cliente del DRBL\n";
    exit;
} else {
    if (-d "$Dir_Img$aula-img" ) {
        system("/opt/drbl/sbin/drbl-ocs -b -g auto -e1 auto -e2 -x -nogui -r -j2 -k -p
reboot --clients-to-wait 1 --max-time-to-wait 300 -h \"$equipo\" -l es_ES.UTF-8 startparts
multicast_restore $aula-img sda3");
    } else {
        print "No existe imagen $Dir_Img$aula-img\n";
        exit;
    }
}
}

```

RecargaLinuxSala.pl

```

#!/usr/bin/perl

# Script para clonar una imagen de Linux en una sala

$IP_HOST_TABLE = "/etc/drbl/IP_HOST_TABLE";

if ( $ARGV[0] eq "--help " || $#ARGV == -1 )
{

```



```

    print "Uso: $0 aula\n";
    print "Realiza una recarga de la partición Linux en el aula indicada.\n";
    exit;
}
$aula=$ARGV[0];

# Vamos a crear la lista de IP`S de las sala

open (IP_HOST, "<$IP_HOST_TABLE");

while (<IP_HOST>)
{
    if (/^$aula/) {
        @linea = split(/\t/, $_);
        $IP = @linea[0];
        # print "$IP\n";
        push(@ListaIPs, $IP);
    }
}
close(IP_HOST);

if ( $#ListaIPs != -1 )
{
    system("/opt/drbl/sbin/drbl-ocs -b -g auto -e1 auto -e2 -x -nogui -r -j2 -k -p reboot --clients-
to-wait $#ListaIPs --max-time-to-wait 300 -h \"@ListaIPs\" -l es_ES.UTF-8 startparts
multicast_restore $aula-img sda3");
} else {
    print "El aula $aula, no pertenece a este servidor DRBL\n";
    exit;
}

```

RecargaXPIP.pl

```
#!/usr/bin/perl
```

```
# Script para clonar una imagen del XP en un equipo.
```

```
$IP_HOST_TABLE = "/etc/dnbl/IP_HOST_TABLE";
```

```
$Dir_Img="/home/partimag/";
```

```
if ( $ARGV[0] eq "--help " || $#ARGV == -1 )
```

```
{
```

```
    print "Uso: $0 IP_Equipo Aula\n";
```

```
    print "Realiza una recarga de la partición XP del disco en el equipo indicado.\n";
```

```
    exit;
```

```
}
```

```
$equipo=$ARGV[0];
```

```
$aula=$ARGV[1];
```

```
open (IP_HOST, "<$IP_HOST_TABLE");
```

```
$existe="no";
```

```
while (<IP_HOST>)
```

```
{
```

```
    if (/ $equipo\s/) {
```

```

        $existe="si";
    }

}

close(IP_HOST);

if ( $existe eq "no" ) {
    print "El equipo especificado $equipo no es un cliente del DRBL\n";
    exit;
} else {
    if (-d "$Dir_Img$aula-img" ) {
        system("/opt/drbl/sbin/drbl-ocs -b -g auto -e1 auto -e2 -x -nogui -r -j2 -o1 -k -p
reboot --clients-to-wait 1 --max-time-to-wait 300 -h \"$equipo\" -l es_ES.UTF-8 startparts
multicast_restore $aula-img sda1 sda2");
    } else {
        print "No existe imagen $Dir_Img$aula-img\n";
        exit;
    }
}
}

```

RecargaXPSala.pl

```

#!/usr/bin/perl

# Script para clonar una imagen de XP en una sala

$IP_HOST_TABLE = "/etc/drbl/IP_HOST_TABLE";

if ( $ARGV[0] eq "--help " || $#ARGV == -1 )
{
    print "Uso: $0 aula\n";
    print "Realiza una recarga completa del disco en el aula indicada.\n";
}

```

```

        exit;
    }
$aula=$ARGV[0];

# Vamos a crear la lista de IP`S de las sala

open (IP_HOST, "<$IP_HOST_TABLE");

while (<IP_HOST>)
{
    if (/^s$aula/) {
        @linea = split(/\t/, $_);
        $IP = @linea[0];
        # print "$IP\n";
        push(@ListaIPs, $IP);
    }
}

close(IP_HOST);

if ( $#ListaIPs != -1 )
{
    system("/opt/drbl/sbin/drbl-ocs -b -g auto -e1 auto -e2 -x -nogui -r -j2 -o1 -k -p reboot
--clients-to-wait $#ListaIPs --max-time-to-wait 300 -h \"@ListaIPs\" -l es_ES.UTF-8 startparts
multicast_restore $aula-img sda1 sda2");
} else {
    print "El aula $aula, no pertenece a este servidor DRBL\n";
    exit;
}

```

ANEXO IV

Script de Creación de Reglas

CrearReglasEquipos.pl

```
#!/usr/bin/perl

#####
# Script a ejecutar despues de configurar clonezilla mediante drblpush -i #
# para crear reglas NIS, NFS a los clientes DHCP de laboratorios. ###
# Tambien crea lista IP_HOST_TABLE #####
#####

##### Ficheros a modificar

$dhcpFile = "/etc/dhcp3/dhcpd.conf";
$IP_HOST_TABLE = "/etc/drbl/IP_HOST_TABLE";
$IP_HOST_TABLE_CP = "/home/partimag/scripts/IP_HOST_TABLE";
$exports = "/etc/exports";
$ypservsecurenets = "/etc/ypserv\securenets";

#### Comprobación de parametros de entrada

if ( $ARGV[0] eq "--help " || $#ARGV == -1 )
{
    print "Crear reglas NFS, NIS y fichero IP_HOST_TABLE a partir del dhcpd.conf";
    exit;
}
```

Funciones

```
sub CrearIP_HOST_TABLE
```

```
{  
    $ip = "@_[0]";  
    $host = "@_[1]";  
    print IP_HOST "$ip\t$host\n";  
}
```

```
sub CrearExports
```

```
{  
    $ip = "@_[0]";  
    print  exports"/tftpboot/node_root $ip(ro,async,no_root_squash,no_subtree_check)\n";  
    print  exports"/usr $ip(ro,async,no_root_squash,no_subtree_check)\n";  
    print  exports"/opt $ip(ro,async,no_root_squash,no_subtree_check)\n";  
    print  exports"/home $ip(rw,sync,no_root_squash,no_subtree_check)\n";  
    print  exports"/var/spool/mail $ip(rw,sync,root_squash,no_subtree_check)\n\n";  
}
```

```
sub CrearYPSecurenets
```

```
{  
    $ip = "@_[0]";  
    print yp "255.255.255.255 $ip\n";  
}
```

PROGRAMA PRINCIPAL

```
# Como es crear reglas por primera vez abrimos los ficheros para que sobreescriban  
# los creados por DRBL.
```

```
open (dhcp,"<$dhcpFile");
open (IP_HOST, ">$IP_HOST_TABLE");
print IP_HOST "\#Fichero creado parseando dhcp.conf\n\n";
```

```
open (exports, ">$exports");
print exports "\#Clientes añadidos del dhcp\n\n";
```

```
open (yp, ">$ypservsecurenets");
```

```
print yp "\#Datos del servidor DRBL\n";
```

```
print yp "255.0.0.0    127.0.0.0\n";
```

```
@ifconfig= `ifconfig`;
$EvalSiguiente="no";
foreach $linea(@ifconfig)
{
    @campos = split(/\s+/, $linea);
    if ( $EvalSiguiente eq "si" )
    {
        @addr=split(/:/, @campos[2]);
        # print "IP: @addr[1]\n";
        $EvalSiguiente="no";
        print yp "255.255.255@addr[1]\n";
    }
    if ( @campos[0] =~ eth )
    {
        $EvalSiguiente="si";
        # print "Dev: @campos[0]\n";
    }
}

}
```

```
print yp "\#Clientes añadidos del dhcp\n\n";
```

```

$siguiente="no";

while (<dhcp>)
{
    if ($siguiente eq "si" && !/^#/ && /(\d+\.\d+\.\d+\.\d+)/ )
    {
        $siguiente="no";
        $ip = $&;
        &CrearIP_HOST_TABLE("$ip", "$equipo");
        &CrearExports("$ip");
        &CrearYPSecurenets("$ip");
    }
    if (/.*host\s.*/) {
        $siguiente="si";
        @linea = split(/ /,$&);
        $equipo = @linea[1];
    }
}

close(yp);
close(exports);
close(IP_HOST);
close(dhcp);

# Borro plantillas para clientes que nos se usaran

system("rm -fr /tftboot/nodes/*");
# Hago copia de la tabla de IP's para usarla al parchear el sysprep

system("cp $IP_HOST_TABLE $IP_HOST_TABLE_CP");

```


Reinicar los servicios NAS y NIS

```
system("/etc/init.d/nfs-kernel-server restart");
```

```
system("/etc/init.d/nfs-common restart");
```

```
system("/etc/init.d/nis restart");
```

ANEXO V

Ejecutando Scripts después de Clonar

Es posible indicarle al servidor Clonezilla que ejecute un scripts cuando termine de clonar (parámetro `-o1` en línea de comandos).

Los scripts a ejecutar se introducen en el directorio `/opt/drbl/share/ocs/postrun/`.

El servidor ejecutará todos los scripts **ejecutables** que se encuentren dentro de ese directorio, por medio de la herramienta **run-parts** de Linux.

Nota: run-parts solo ejecuta scripts cuyos nombres contengan mayúsculas, minúsculas, dígitos, guiones bajos y medios.

En nuestro caso hemos creado un script ejecutable llamado `scripts_sh` que llamará a otro script que se encuentra en el `/home/partimag/scripts/` llamado `scripts.sh`.

scripts_sh

#!/bin/bash

/home/partimag/scripts/scripts.sh

scripts.sh

#!/bin/bash

echo "Parseando DHCP"

/home/partimag/scripts/ParsearSysprep.sh

echo "Hecho"

La idea es poder modificar las tareas a realizar después de clonar sin necesidad de tener que reconfigurar el servidor Clonezilla. Con este sistema basta modificar el contenido del script `scripts.sh`.

Scripts para Parsear Sysprep

Estos Script copian el fichero sysprep.xml ubicado en el servidor (en */home/partimag/sysprep/*) en la partición de Windows 7 y lo parsean con la información correspondiente al cliente que se esta clonando.

ParsearSysprep.sh

```
#!/bin/sh
ntfs-3g /dev/sda2 /mnt/ &&
perl /home/partimag/scripts/ParsearSysprep.pl &&
umount /mnt
```

ParsearSysprep.pl

```
#!/usr/bin/perl

$IP_HOST_TABLE = "/home/partimag/scripts/IP_HOST_TABLE";
$SYSPREP = "/home/partimag/sysprep/sysprep.xml";
$SYSPREP_PARCH = "/mnt/Windows/System32/sysprep/sysprep.xml";
```

```
$HOST_IP = `hostname`;
$HOST_IP =~ s/-/./g;
$HOST_IP =~ s/\n//g;
```

```
open(tabla_ip, "<$IP_HOST_TABLE");
while (<tabla_ip>) {
    if (/ $HOST_IP/) {
        $linea=$_;
        $linea =~ s/\s+/ /g;
```

```
        @linea = split(/ /,$linea);
        $hostname = @linea[1];
    }
}
close(tabla_ip);

open(origen,"<$SYSPREP");
open(destino,">$SYSPREP_PARCH");

while(<origen>)
{
    s/informaticaX-XX/$hostname/g;
    print destino $_;

}

close(origen);
close(destino);
```
