

Configuración de un servidor de impresión para uso doméstico fuera de cualquier hardware antiguo con Slackware

Después de actualizar recientemente la red wifi de mi casa para usar wpa2, mi antiguo servidor de impresión wifi Edimax dejó de funcionar debido a la incompatibilidad con wpa2. No quería tener que dejar la computadora encendida todo el tiempo ni tampoco quería conectar directamente la PC desde la que necesito imprimir desde la impresora.

La solución más elegante sería configurar algún dispositivo ARM de bajo consumo (que se ejecute naturalmente Slackware ARM) para que se convierta en un servidor de impresión, pero cualquier hardware antiguo que pueda ejecutarse con Slackware Linux estará bien y si está dispuesto a seguir con las distribuciones alternativas, suponga Cualquier hardware que pueda ejecutar Linux lo hará.

Sé que no es una cosa difícil, pero desde la última vez que hice algo como esto muchas cosas han cambiado, pensé que otras personas podrían beneficiarse de un inicio rápido para ponerse en marcha muy rápido ... así que aquí vamos: (los pasos Supongo que la red ya se ha configurado correctamente con todos los equipos)

Configuración de la impresora en el servidor de impresión

Lo primero que puede que le resulte útil saber es que no es obligatorio tener el filtro de impresión correcto para la impresora en el servidor de impresión, el cliente tiene la obligación. El servidor de impresión solo necesita tener la impresora configurada como impresora sin formato.

El servidor de la impresora necesitará los paquetes cups, cyrus-sasl y openssl (openssl solo es obligatorio si desea administrar tazas de forma remota). Si su impresora tiene una interfaz usb, como la mayoría actualmente, será necesario tener también estos paquetes para ayudar al módulo del kernel usblp: libusb, libusb-compat y usbutils. Una vez que haya instalado los paquetes, si luego conecta la impresora, udev debería cargar automáticamente el módulo del kernel usblp y lsusb debería mostrar, entre otras cosas, su impresora. Este es el tipo de salida que necesita ver desde lsusb para confirmar que se detectó la impresora:

```
root@printserver:~# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 05e3:0608 Genesys Logic, Inc. USB-2.0 4-Port HUB
Bus 001 Device 003: ID 0bda:0119 Realtek Semiconductor Corp. Storage Device
(SD card reader)
Bus 001 Device 004: ID 0ace:1211 ZyDAS ZD1211 802.11g
Bus 001 Device 005: ID 04e8:3260 Samsung Electronics Co., Ltd CLP-510 Color
Laser Printer
root@printserver:~#
```

Si puede ver algo similar a esto, donde su impresora es detectada por lsusb, puede proceder de lo contrario, debe depurar las razones que impiden que su impresora sea detectada.

Una vez que haya ordenado los paquetes y la impresora, puede dar permisos de ejecución a `/etc/rc.d/rc.cups` y luego iniciarlo. Si pretende administrarlo de manera remota, debería ejecutar este comando para permitirlo:

```
cupsctl --remote-admin
```

Entonces abre tu navegador y accede

```
http://<your print server ip>:631  
o  
http://localhost:631 en el propio servidor de impresión
```

En la mayoría de los casos, puede usar enlaces en un terminal de texto en el mismo servidor de impresión. Las únicas cosas que no funcionan correctamente en los enlaces son algunas selecciones de cuadros combinados, por eso permití la administración remota. Continúe y configure la impresora local como una impresora sin formato. Asegúrese de marcar la casilla de una impresora compartida. Puede llamar a la impresora como quiera, pero para mayor comodidad en esta guía, la llamaremos "prueba". Su `/etc/cups/printers.conf` debería tener este aspecto:

```
<Printer test>  
UUID urn:uuid:8d60a6be-4d86-3abf-5b8d-d5a03f10a753  
Info test printer  
Location test location  
DeviceURI usb:///Samsung/CLP-510?serial=xxxxxxxxxxxxx. #your setup will  
depend on the printer brand on how it is connected to the print server  
State Idle  
StateTime 16147  
Type 4  
Accepting Yes  
Shared Yes  
JobSheets none none  
QuotaPeriod 0  
PageLimit 0  
KLimit 0  
OpPolicy default  
ErrorPolicy abort-job  
</Printer>
```

Si tiene los controladores para su impresora, puede configurarlo opcionalmente con el filtro de impresión correcto, pero no es obligatorio. Si los tiene, puede ser una buena idea configurarlo correctamente solo para verificar que el servidor de impresión pueda usar la impresora correctamente.

Una vez que la impresora está configurada, puede indicar a las tazas que acepten las solicitudes de impresión remota:

```
cupsctl --share-printers --remote-any --remote-admin
```

Si no está interesado en compartir la impresora fuera de su red de área local, simplemente elimine la opción “-remote-any” y, por defecto, Cups solo servirá para imprimir solicitudes desde su LAN.

Algunas plataformas de hardware no le dejan ninguna manera de hacer un cierre limpio sin interactuar con el sistema operativo que se ejecuta en él (como el antiguo AT PC o seagate dockstar). Puede solucionar este problema haciendo una regla de udev que ejecute un apagado cuando se retira la impresora (desenchufada o apagada) para que cuando termine de imprimir, apague la impresora, el servidor de impresión también se apagaría sin fallar.

Configuración de la impresora en los clientes de impresión

Ahora vaya a la PC donde desea acceder a la impresora y configure una impresora remota ipp con el controlador correcto para la impresora. El /etc/cups/printers.conf debe tener una entrada similar a esta:

```
<Printer test>
UUID urn:uuid:6abca077-c999-3d8a-5ce0-41b7bd3c2ddf
AuthInfoRequired none
Info test
Location study room
MakeModel Samsung CLP-510, 2.0.0 #in this case I setup splix driver for the
samsung printer but this entirely depends on my setup
DeviceURI ipp://<print server ip address>:631/printers/test
State Idle
StateTime 1387373858
Type 8400972
Accepting Yes
Shared No
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
OpPolicy default
ErrorPolicy abort-job
</Printer>
```

Ahora puede intentar imprimir una página de prueba en la PC cliente.

Hacer que el servidor de impresión se bloquee a prueba

Una vez que tenga su servidor de impresión funcionando correctamente, probablemente querrá evitar hacerle nada a menos que sea realmente necesario. Cuidar de que se apague correctamente y resolver los problemas de fsck en el reinicio no debería ser algo de lo que preocuparse, pero los sistemas de archivos registrados están diseñados para hacer frente a las cortes de energía accidentales, no realmente para que bloquee el sistema cada vez que lo apague. Tener el sistema de archivos raíz montado solo para lectura, con solo las partes necesarias para escribir vinculadas en tmpfs, nos liberaría de tener que preocuparnos por apagar el servidor de impresión.

Así es como modifiqué rc.S y rc.M para el trabajo:

```
rc.S.org 2013-12-28 14:25:08.569250342 +0100
rc.S     2013-12-28 14:15:50.594483496 +0100
@@ -18,12 +18,31 @@
     fi
     fi
     fi
+echo "timer" > /sys/class/leds/dockstar:green:health/trigger
+echo "125" > /sys/class/leds/dockstar:green:health/delay_on
+echo "250" > /sys/class/leds/dockstar:green:health/delay_off

+#
if [ -d /run ]; then
    if ! grep -wq "tmpfs /run tmpfs" /proc/mounts ; then
        /sbin/mount -v -n -t tmpfs tmpfs /run -o mode=0755
    fi
fi
+( cd /run
+ /usr/bin/mkdir -p log/{cups,nfsd,samba} spool/{cups/tmp,mail}
run/{cups/certs,dbus,dhccpd/ntp.conf,dhccpd/resolv.conf} tmp
+ /usr/bin/chmod a+rwx tmp lock/subsys
+ for DIR in packages removed_packages removed_scripts scripts setup
+ do
+ /usr/bin/ln -s /var/static/$DIR log/$DIR
+ done
+ /usr/bin/ln -s /var/static/cron spool/cron
+ /usr/bin/touch /run/ld.so.cache
+)

# Load the loop device kernel module:
if [ -x /etc/rc.d/rc.loop ]; then
@@ -219,32 +238,32 @@
    reboot -f
    fi
    # Remount the root filesystem in read-write mode
- echo "Remounting root device with read-write enabled."
- /sbin/mount -w -v -n -o remount /
- if [ $? -gt 0 ] ; then
- echo
- echo "Attempt to remount root device as read-write failed! This is
going to"
- echo "cause serious problems."
- echo
- echo "If you're using the UMSDOS filesystem, you MUST mount the
root partition"
- echo "read-write! You can make sure the root filesystem is getting
```

```
mounted "
-   echo "read-write with the 'rw' flag to Loadlin:"
-   echo
-   echo "loadlin vmlinuz root=/dev/hda1 rw   (replace /dev/hda1 with your
root device)"
-   echo
-   echo "Normal bootdisks can be made to mount a system read-write with
the rdev command:"
-   echo
-   -   echo "rdev -R /dev/fd0 0"
-   echo
-   echo "You can also get into your system by using a boot disk with a
command like this"
-   echo "on the LILO prompt line: (change the root partition name as
needed)"
-   echo
-   echo "LILO: mount root=/dev/hda1 rw"
-   echo
-   echo "Please press ENTER to continue, then reboot and use one of the
above methods to"
-   echo -n "get into your machine and start looking for the problem. "
-   read junk;
- fi
+# echo "Remounting root device with read-write enabled."
+# /sbin/mount -w -v -n -o remount /
+# if [ $? -gt 0 ] ; then
+#   echo
+#   echo "Attempt to remount root device as read-write failed! This is
going to"
+#   echo "cause serious problems."
+#   echo
+#   echo "If you're using the UMSDOS filesystem, you **MUST** mount the
root partition"
+#   echo "read-write! You can make sure the root filesystem is getting
mounted "
+#   echo "read-write with the 'rw' flag to Loadlin:"
+#   echo
+#   echo "loadlin vmlinuz root=/dev/hda1 rw   (replace /dev/hda1 with your
root device)"
+#   echo
+#   echo "Normal bootdisks can be made to mount a system read-write with
the rdev command:"
+#   echo
+#   echo "rdev -R /dev/fd0 0"
+#   echo
+#   echo "You can also get into your system by using a boot disk with a
command like this"
+#   echo "on the LILO prompt line: (change the root partition name as
needed)"
+#   echo
```

```
+# echo "LILO: mount root=/dev/hda1 rw"
+# echo
+# echo "Please press ENTER to continue, then reboot and use one of the
above methods to"
+# echo -n "get into your machine and start looking for the problem. "
+# read junk;
+# fi
else
echo "Testing root filesystem status: read-write filesystem"
echo
--- rc.M.org 2013-12-28 14:24:12.088970256 +0100
+++ rc.M 2013-12-28 14:15:56.890514695 +0100
@@ -17,7 +17,7 @@
# Update all the shared library links:
if [ -x /sbin/ldconfig ]; then
echo "Updating shared library links: /sbin/ldconfig &"
- /sbin/ldconfig &
+ /sbin/ldconfig -C /run/ld.so.cache &
fi

# Screen blanks after 15 minutes idle time, and powers down in one hour
```

Un poco de retoques con enlaces en var y en algunos otros lugares y ya está.

```
root@printserver:~# ls -l /var/
total 0
lrwxrwxrwx 1 root root 3 Nov 4 2013 adm -> log/
drwxr-xr-x 5 root root 368 Jan 1 1970 cache/
drwxr-xr-x 2 root root 160 Oct 18 2013 empty/
drwxr-xr-x 7 root root 488 Jan 1 1970 lib/
lrwxrwxrwx 1 root root 9 Jan 1 01:29 lock -> /run/lock/
lrwxrwxrwx 1 root root 10 Jan 1 1970 log -> ../run/log/
lrwxrwxrwx 1 root root 10 Nov 4 2013 mail -> spool/mail/
drwxr-xr-x 12 root root 800 Nov 25 1993 man/
lrwxrwxrwx 1 root root 10 Jan 1 1970 run -> ../run/run/
lrwxrwxrwx 1 root root 15 Nov 4 2013 rwho -> /var/spool/rwho
lrwxrwxrwx 1 root root 12 Jan 1 1970 spool -> ../run/spool/
drwxr-xr-x 3 root root 232 Jul 15 2013 state/
drwxr-xr-x 8 root root 576 Jan 1 1970 static/
lrwxrwxrwx 1 root root 10 Jan 1 1970 tmp -> ../run/tmp/
root@printserver:~# ls -l /etc/ld.so.cache
lrwxrwxrwx 1 root root 16 Jan 1 01:06 /etc/ld.so.cache -> /run/ld.so.cache
root@printserver:~# ls -l /run/log
total 40
-rw-r--r-- 1 root root 0 Jan 1 01:00 cron
drwxr-xr-x 2 root root 100 Jan 1 01:01 cups/
-rw-r--r-- 1 root root 1280 Jan 1 01:01 debug
-rw-r--r-- 1 root root 11512 Jan 1 01:00 dmesg
```

```

-rw-r--r-- 1 root root    0 Jan  1 01:00 maillog
-rw-r--r-- 1 root root 19290 Jan  1 01:06 messages
drwxr-xr-x 2 root root   40 Jan  1 01:00 nfsd/
lrwxrwxrwx 1 root root   20 Jan  1 01:00 packages -> /var/static/packages/
lrwxrwxrwx 1 root root   28 Jan  1 01:00 removed_packages ->
/var/static/removed_packages/
lrwxrwxrwx 1 root root   27 Jan  1 01:00 removed_scripts ->
/var/static/removed_scripts/
drwxr-xr-x 2 root root   40 Jan  1 01:00 samba/
lrwxrwxrwx 1 root root   19 Jan  1 01:00 scripts -> /var/static/scripts/
-rw-r--r-- 1 root root    0 Jan  1 01:00 secure
lrwxrwxrwx 1 root root   17 Jan  1 01:00 setup -> /var/static/setup/
-rw-r--r-- 1 root root    0 Jan  1 01:00 spooler
-rw-r--r-- 1 root root 2083 Jan  1 01:01 syslog
root@printserver:~# ls -l /run/spool/
total 0
lrwxrwxrwx 1 root root 16 Jan  1 01:00 cron -> /var/static/cron/
drwx--x--- 3 root lp   60 Jan  1 01:00 cups/
drwxr-xr-x 2 root root 40 Jan  1 01:00 mail/
root@printserver:~# ls -l /tmp
lrwxrwxrwx 1 root root 7 Jan  1 01:03 /tmp -> run/tmp/
root@printserver:~# mount
proc on /proc type proc (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
tmpfs on /run type tmpfs (rw,relatime,mode=755)
devtmpfs on /dev type devtmpfs
(rw,relatime,size=60180k,nr_inodes=15045,mode=755)
/dev/ubi0_0 on / type ubifs (ro,relatime)
devpts on /dev/pts type devpts (rw,relatime,gid=5,mode=620,ptmxmode=000)
cgroup on /sys/fs/cgroup type cgroup
(rw,relatime,net_cls,freezer,devices,cpuacct)
root@printserver:~#

```

Con estos enlaces, la distribución de Slackware funcionará de manera bastante normal y podrá reanudar la operación normal simplemente volviendo a montar la lectura/escritura de la raíz “mount -o remount, rw /” para permitir cierto mantenimiento (como la gestión de paquetes) y cuando se hace, simplemente vuelva a montarlo “solo lectura” con un simple “mount -o remount,ro /”.

Aquí está mi servidor de impresión wifi

Aquí está uno de mis Dockstars en una carcasa casera que ejecuta el servidor de impresión descrito anteriormente desde el flash interno. (la memoria USB en la parte superior es una tarjeta wifi)



Fuentes

- Escrito originalmente por [louigi600](#)
- Traducido por: [Victor](#) 2019/02/17 14:01 (UTC)

[howtos](#), [print](#), [server](#)

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
https://docs.slackware.com/es:howtos:network_services:slackware_print_server

Last update: **2019/02/17 14:02 (UTC)**

