Setting up a print server for home use out of any old Slackware box

After recently upgrading my home wifi network to use wpa2 my old edimax wifi print server stopped working due to incompatibility with wpa2. I did not want to have to leave a computer on all the time neither did I want to directly connect whatever PC I need to print from to the printer itself.

The most elegant solution would be to setup some low power ARM device (naturally running Slackware ARM) to become a print server, but any old hardware that can run Slackware linux will be fine and if you're willing to stick up with alternative distributions I suppose any hardware that can run linux will do.

I know it's not a difficult thing but since the last time I did something like this a lot of things have changed, I thought that other people could benefit from a quickstart to get going really fast ... so here we go: (the steps blow assume that networking has already been setup correctly on all participants)

Setting up the printer on the print server

First thing you might find handy to know is that it is not mandatory to have the correct print filter for the printer on the print server, the client is mandated with that burden. The print server needs only have the printer configured as a raw printer.

The printer server will need cups, cyrus-sasl and openssl packages (openssl is only mandatory if you wish to remotely administer cups). If your printer has a usb interface, like most currently, it will be necessary to also have these packages to assist the usblp kernel module: libusb, libusb-compat and usbutils. Once the packages are installed if you then plug in the printer udev should automatically load usblp kernel module and lsusb should list, possibly amongst other things, your printer. This is the sort of output you need to see from lsusb to confirm that the printer is detected:

```
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:~#
```

If you can see something similar to this, where your printer is detected by lsusb, you can proceed otherwise you need to debug the reasons that are preventing your printer from being detected.

Once you have the packages and printer sorted out you can then just give execute permissions to /etc/rc.d/rc.cups and then start it. If you intend to remotely administer it you might like to execute this command to allow it:

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

then fire up your browser and access

```
http://<your print server ip>:631
or
http://localhost:631 on the printserver itself
```

For the most part you can use links on a text terminal on the print-server itself the only things that not working right on links are some combo box selections, that's why I then allowed for remote administration. Go ahead and configure the local printer as a raw printer. Be sure to check the box for a shared printer.You can call the printer whatever you like but for convenience in this guide we will call it “test”. Your /etc/cups/printers.conf should look like this:

```
<Printer test>
UUID urn:uuid:8d60a6be-4d86-3abf-5b8d-d5a03f10a753
Info test printer
Location test location
DeviceURI usb://Samsung/CLP-510?serial=xxxxxxxxxxxx. #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>
```

If you have the drivers for your printer you can optionally configure it with the correct print filter but it's not mandatory. If you do have them it might be a good idea to configure it right just to check that the printserver can correctly use the printer.

Once the printer is configured you can then instruct cups to accept remote printing requests:

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

If you're not interested to share the printer to outside your Local Area Net just remove the “–remote-any” option and by default cups will only serve printing requests from your LAN.

Some hardware platforms leave you no way to do a clean shutdown without interacting to the os running on it (like old AT PC or seagate dockstar). You can work around this problem by making a udev rule that executes a shutdown when the printer is removed (unplugged or turned off) so that when you're done printing, you turn off the printer, the print-server would shutdown along too without crashing.

Setting up the printer on the print clients

Now go on the PC's where you want to access the printer and configure an ipp remote printer with the correct driver for the printer. The /etc/cups/printers.conf should have an entry similar to this:

```
<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>
```

You may now try printing a test page on the client PC.

Making the printserver crash proof

Once you have your print server working right you will probably want avoid doing anything to it unless it's really necessary. Looking after it's shutting down right and resolving fsck issues on reboot should not be something to worry about but journaled filesystems are meant to deal with accidental power rages not really for you to crash the system every time you switch it off.

Having the root filesystem mounted read only, with only the parts that are necessary for writing linked into tmpfs, would free us form having to worry about powering down the print server.

This is how I modified rc.S and rc.M for the job:

```
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,dhcpcd/ntp.conf,dhcpcd/resolv.conf} tmp
+  /usr/bin/chmod a+rwxt 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
```

A bit of tinkering with links in var and a few other places and you're done.

```
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:~#
```

With these links the Slackware distribution will operate pretty much normally and can resume to ordinary operation by just remounting root read/write “mount -o remount,rw /” to allow some maintenance (like package management) and when that's done just remount it ro with a simple “mount -o remount,ro /”.

Here's my wifi print server

Here's one of my Dockstars in a homebrew casing running the above described printserver from internal flash. (the usb stick on top is a wifi stick)

Sources

- Originally written by louigi600

howtos, print, server