

# Open your office to the outside world

# GET CONNECTED

## Networking in mind

Linux has been built with networking in mind, making it the ideal platform to share information on. Your office generates information, and the benefits of being able to share that information are increasing daily. It is possible to not only share this information over a local network, but via the Internet too, if necessary.

Email is not only very common, but also valuable to your business. You can take full control over your email by running a few servers on a permanent connection. This minimises the impact should your ISP let you down. The ability to seamlessly share files across a network – be it across the office or across the world – enables you to get the maximum out of your data. This is where NFS can help you, and if you are looking to share files with Windows systems, then Samba gives you that option. Sharing printing resources saves time and money, and we also cover that to in this article.

## Mail services

With the best will in the world, a detailed examination of running email services on your network could not be done justice in just a few pages, so we won't even attempt to try. What we will do is highlight all of the important aspects of email and offer you further pointers to them.

## To read and write

The handling of email takes on three main processes. At the very outset or endpoint, an email needs to be written or read. This task falls to the *Mail User Agent*. There are MUAs to suit a range of people and needs from the most basic – *mail* – through programs like *pine* and *elm* – which are terminal based, to MUAs that have all the bells, whistles and graphics, to run on desktops.

When receiving mail, the more basic MUAs expect to find files to work on locally – they do not have any means at their disposal for collecting email from other sources – either locally, or externally. For this they need to call upon the services of a *Delivery Agent*. The

DA in turn calls on the services of the *Mail Transport Agent* (MTA).

As root type *mail* in to a terminal to get an example of what email was like 20 years ago. Enter a message number to view a message, enter *q* to come out of the message and *q* again to come out of *mail*. Examples of MUAs with bells and whistles include KMail, Sylpheed and Netscape Mail to name just three. These three, and others of their type, are unusual in that they also deal with the actions of the remaining two processes (those covered by the DA and MTA). Because of this, they really fall outside of what we are going to continue to describe, and so won't be mentioned again.

This modularity allows for complete control over the system, which is why, in a multi-user networked system you would choose to administer your system this way.

Which MUA you wish to use is largely a question of

Office communication is of the highest importance and networking is what Linux does best. Colin Murphy looks at how Linux could help you in a SOHO environment

*elm* is a small MUA, which uses an external text editor for its part in handling email. By default, this is usually *vi*, though this can be easily changed if that thought sends shivers down your spine. When run for the first time *elm* will create its own runtime configuration file in the user's home directory `#!/elm/elmrc`. In this file you will find the editor option, which you can change to suit your needs.

**pine** is a much more feature rich MUA than *elm*, as a browse through its configuration file (though the program does provide its own configuration utility) `~/pinerc` will show. *pine* supports lots of useful features like multiple email folders and automatic backing up and archiving of old messages – both sent and received.

personal choice. When making this decision the administrator will want to be conscious of size, speed and the availability of a global configuration file to administer the MUA, as well keeping in mind the demands of users for more and more features.

### Moving on

The Delivery Agent can be called upon to deliver mail to a user or to send control messages to other programs. It places mail in mail stores, once it has been handed over by the MTA. There are only a few DAs – */bin/mail* being the basic one found on most distributions by default – but a lot of people put their faith in *procmail*, which enables you to create a range of rules to define where certain messages get stored. A large part of *procmail*'s success lies in the fact that you can make rules that will automatically dispose of spam.

### Transports of delight

The MTA lies at the heart of the worldwide email system. There are three main players when it comes to transport agents: *sendmail*, *exim* and *qmail*. It is said that as much as 60 per cent of the world's email travels courtesy of *sendmail*, compared to as little as four per cent for Microsoft Exchange Server.

*sendmail* has a complicated configuration file that no-one in their right mind attempts to configure by hand these days. For those that prefer the quiet tranquillity of their terminal, this elevates the configuration process to something akin to a scripting language, as opposed to plugging away in machine code.

Configuration is also possible, and practical in some circumstances, via graphical tools, Figure 1 shows part of this procedure in the YaST tool, as used in the SuSE distribution.

Configuration is also possible through the use of



Fig 1: The YaST tool can help with the setting up sendmail



Fig 2: Further uses of WebMin in helping to configure network services over the network. Here it is with *sendmail*

WebMin, which enables you to use any browser to configure, remotely if needs be, many types of services including *sendmail*, an example of which can be seen in Figure 2.

### The POP3 connection

If you do not have a permanent Internet connection, then there is one more stage that needs to be considered and configured. Without a permanent connection, email coming to you needs to be stored on a POP3 server, or possibly an IMAP server, while you do not have a connection. As part of your own local mail system you will need to run a service that will regularly collect mail from this store. *fetchmail* is the most often used in distributions like Mandrake and Red Hat, while SuSE will provide you with *popper*. Hopefully these highlights will give you the opportunity to further investigate the power and possibilities to be had when running your own email services.

### Why use NFS?

NFS means we can keep just one copy of data and access it from wherever we choose. When the data is updated all users can see the information at the same time and so are in sync. By having the data centralised we can also have a simpler backup policy and thus keep our jobs. NFS is also access transparent. This means we can be machine and operating system independent, which in a modern network is very important. It is also location transparent. The client adds the filesystem on the remote server to its name space. The server exports the filesystems and the client mounts these before access is enabled.

### Drawbacks

- Write performance is slow.
- Giving access to files on a system inevitably introduces security flaws that should be considered. Spoofing can be a problem, as the

**Stateless** All of the file access operations are independent. Each client call to the server has all the necessary information to carry out the task. If the server crashes the client just keeps on resending requests until the server is rebooted. If the client crashes the server does not care and once the client is rebooted the program can be restarted.

## NFS (Network File System)

The NFS started in 1985 and was developed by Sun Microsystems. It lets you share a filesystem amongst computers on your network. It is a Server-Client system where one computer (the client) accesses the files on another (the server). For simple crash recovery the system is **stateless**.

server trusts that each client is who they say they are, based on permissions in the */etc/exports* file.

- NFS runs on top of the RPC (Remote Procedure Call) protocol either on UDP or TCP. UDP has no headers so can be thought of as streaming. The advantage is speed but the disadvantage is a lack of failover protection.

To prevent anyone with root access from running rampant over the files we can use root squashing. This is where root access is intercepted and modified to act only as a user. Care must be taken with the *hosts.allow* file, as a missing space in the wrong place can make all the difference:

```
/Public client(rw,no_root_squash)
/Public client (rw,no_root_squash)
```

The first line will grant rewrite access without squashing root privileges.

## Setting up the server

We need to configure the three access files: */etc/exports*, */etc/hosts.allow* and */etc/hosts.deny*. The first contains a list of directories that we wish to share and to whom, thus:

```
/server moon(ro)
/Public (rw,root_squash, all_squash)
/usr/local 192.168.0.0/255.255.255.0(ro)
```

This gives the client machine ("moon") read only access to the */server* directory and all machines read-write access to the */Public* directory. We have also given read only access to */usr/local* to a range of IP addresses.

*/etc/hosts.allow* contains a list of machines to allow, and */etc/hosts.deny* contains a list of machines we want to deny access. If a machine is not found in either then access is allowed. So in */etc/hosts.deny* we have the lines:

```
portmap:ALL
lockd:ALL
mountd:ALL
rquotad:ALL
statd:ALL
```

The first line restricts the *portmapper* daemon and so makes it a little harder for an unauthorised user to find where the NFS services are residing. If we were being ultra cautious we would just use the line *ALL:ALL* and then rely on specific allows in the *hosts.allow* file.

For our *etc/hosts.allow* we have the following lines:

```
portmap: 198.168.0.1, 189.168.0.2
lockd: 198.168.0.1, 189.168.0.2
```

```
mountd: 198.168.0.1, 189.168.0.2
rquotad: 198.168.0.1, 189.168.0.2
statd: 198.168.0.1, 189.168.0.2
```

We now have access for two specific clients. We can now usually just reboot and the NFS daemon will read in the configuration files. We can also start the daemons by hand if required:

```
rpcinfo -p
```

will tell us if *portmapper*, *rstatd*, *mountd* and NFS are running.

```
rpc.portmap
rpc.mountd
rpc.nfsd
rpc.statd, rpc.lockd (if necessary),
rpc.rquotad
```

Now for the client machine. Create a directory that we will use as the mount point:

```
mkdir /share1
```

Now mount the NFS filesystem:

```
mount -t nfs earth:/server /share1
```

This now lets us see the contents of the */server* directory on the computer "earth" in our directory */share1*.

If we want this each time we boot we must include the following line in our */etc/fstab*:

```
earth:/server /share1 nfs rw 0 0
```

That is all that is required. You can now access all those MP3 files from the server wherever on the network you happen to be sitting.

## samba

The main configuration file for **samba**, *smb.conf*, can be found either in */etc/samba/* or in */usr/local/samba/bin/* depending on how you installed the package. Opening the conf file in a text editor we can see two main sections. The first section is for Global settings.

The first line lets us set the Windows workgroup or NT domain:

```
workgroup=MYGROUP
```

The second line allows us to define the server name that will appear when browsing in the Windows network neighbourhood:

```
server string = SambaBox
```

**samba** *samba* is a useful tool that allows files to be shared amongst Windows and Linux users. It allows hard drives, directories and printers to be used as shared resources.

Next we will include the two lines:

```
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd
```

This will allow connection from Windows 98, Me, NT and 2000, which use encrypted passwords. 95 just uses PlainText.

The second part of the `smb.conf` file is for the Share definitions. The default settings within the `conf` file let users with the same username on a windows machine access the user's home directories on the Linux machines.

```
[homes]
comment = Home Directories
browseable = yes
writable = yes
```

We can now save the config file. We now need to create a *samba* user password. As root use the command:

```
smbpasswd -a username
```

where *username* is the name which you sign on to the windows machine with. You are then asked for the password, which should be the same as that on the Windows machine. Remember the password is case sensitive.

Now we can start the daemon as root with the command:

```
smb start
```

If we were lucky enough to have installed from source we should start two daemons:

```
smbd -D
nmbd -D
```

The latter daemon is for SMB or CIFS clients and allows NetBIOS over IP name service requests, while the former gives print and file services to SMB clients.

At this stage we can login to the Windows machine and see the *samba* box in the network neighbourhood. To map the network drive, in Windows Explorer use Tools/Map Network Drive. Choose a drive letter as a mount point (E:, F:, G: etc.) and enter the path to share:

```
\\hostname\username
```

This will then attach the share to the drive letter you chose.

To test for errors you can always run *testparm* as root and then check the two *samba* log files, `log.smbd` and `log.nmbd` as well as the usual `/var/log/messages`.

The `smb.conf` file does have more options such as

setting up printer shares and security levels but you can easily add this with the help of the HOWTOs. For those of you who do not like the idea of using a text editor to change the configuration file there are graphical front-ends.

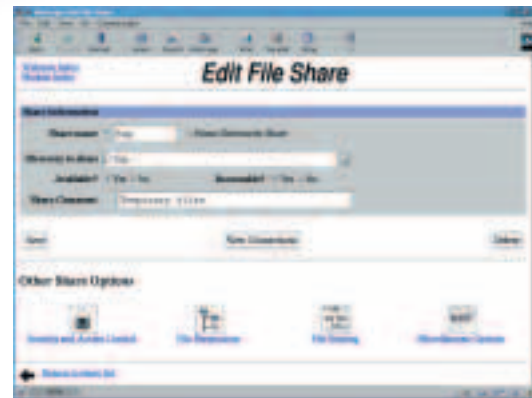
## Ksamba

Ksamba acts as a GUI to aid in the configuration of the `smb.conf` file. With the wizard are some set templates that enable quick set up.

## Webmin

If you have not previously come across Webmin, now is a good time to install. Webmin is a Web-based interface for system administration – using any browser that supports tables and forms, you can set up *samba* and a host of other programs and daemons. It uses modules, which allows you to add more CGI scripts to control more tasks. The *samba* module is *samba.wbm*

The Server Message Block (SMB) is a network protocol for sharing files, printers, serial ports and communications abstractions such as named pipes and mail slots between computers.



Webmin's samba interface

## Printing and print servers

The paperless office isn't with us quite yet, unfortunately, and we still need to get information on paper. Printing in an office environment is one of those core functions that can't afford to go wrong. By spending a little time and effort, maybe even a little money, you can set yourself up with a rock solid system and have one less thing to worry about.

Linux has an advantage in this arena, as it is made with networking in mind. There are many advantages to wanting to have your printing resources on a network. Decent printers don't come cheap, so being able to share their use out over a number of users reduces the burden of capital expense a little. Wastage is minimised, as are the costs on consumables, which, now you are using more of the same type, can be bought in larger quantities.

**PostScript** PostScript is a programming language used to describe the way an image is supposed to appear on the page. It is a printing standard which is widely acknowledged as the standard. A PostScript compatible printer actually decodes this page description itself to render the final image to the page. Because of this, PostScript printers need to have some processing power on board, which, in turn, makes them more expensive. However, the inclusion of PostScript means you can rely on the printer running on Linux and will cut down on the number of unpleasant surprises when in use.

## Cost implications

Having a shared printer does not mean you will have to tie up a computer to serve it. The resources put on a print server can be quite small if all the page rendering is done on the originating machine.

## Installation

All of the modern commercial Linux distributions will come with their own suite of printer configuration tools, usually graphical. For these to work, you will need to have a printer that is supported under Linux, and not all are. The ease with which you can configure a printer depends heavily on the amount of support Linux offers it.

**PostScript** is the feature to lookout for when deciding on a laser printer. If your printing needs don't run as far as the extra expense incurred through buying a PostScript printer, then you should check out the suitability of a printer beforehand. There are some good resources available on the Internet, one of the best being <http://www.linuxprinting.org/>.

## Spools

Because of the way that Linux has grown it still relies very much on some old toolsets, which have embedded themselves very deeply into the heart of the operating system. LPD (the Line Printer Daemon), came about in the days when heavy line printers where the only things available. As technology has changed to include more modern printing systems, like laser printers and inkjet printers, so LPD has had elements added to it to help it cope.

LPD revolves around the configuration file `/etc/printcap`, which contains details of all the printers known to the system. Each entry is referred to as a spool.

Printers in this situation doesn't necessarily refer just to the hardware device, it is customary to set up different spools which refer to the same physical printer but have different options attached, like print quality or which paper tray to use. By doing this, it is simple to access the range of features your printers may have from the command line. There are also utilities available that let you select these options graphically, but it is still easier to select a pre-configured spool with a single click than going through the myriad of options each time.

LPD, or some development of it like LPRng which has some enhanced printer spool facilities, will be provided by default on all Linux systems.

## CUPS

CUPS (the Common Unix Printing System) is another way to bring printing services to your system. CUPS is a modern departure from the old school, and calls upon a new printing protocol – IPP. While LPD can be used for network printing, it can be a little clunky in

## Network printers

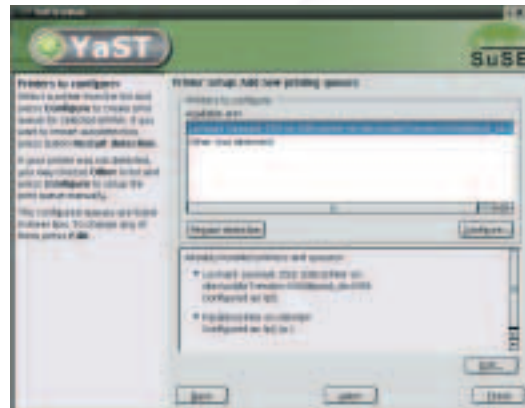
To simplify the set up of printing on anything more complicated than a standalone workstation for one, you have the option of spending a little extra on your printer by making sure it is network compatible

In effect, you are getting a printer that connects directly to your network, which can then be configured so that every one else on that network can see and use it. This is especially true for those that are going to have a mixture of different clients on their network or if you are planning on using *samba*.

use, especially when you want to monitor the state of your network printer (i.e. is it online, has it finished printing that last job, has it now run out of paper?).

With CUPS you have a more natural way of investigating printers on a network, especially if your distribution ships with Webmin, as Mandrake and Red Hat do. Webmin allows you to access the CUPS server control front-end via a HTTP connection, through a Web browser.

All of the distributions will come with tools allowing you to set up networked printer sharing without too much hair pulling, but this can only be relied upon if the underlying network has been configured.



SuSE sets up and configures printers with its YaST configuration tool

## Info

Webmin homepage  
The Linux email administrator HOWTO

Elm user's guide  
Pine information centre  
Sendmail homepage  
Samba homepage  
Webmin  
Ksamba

The Printing HOWTO  
The CUPS Web page  
An example of Webmin in action

<http://www.webmin.com/>  
<http://www.tldp.org/HOWTO/Mail-Administrator-HOWTO.html>  
<http://www.tldp.org/HOWTO/Mail-User-HOWTO/>  
<http://www.instinct.org/elm/>  
<http://www.washington.edu/pine/>  
<http://www.sendmail.org/>  
<http://www.samba.org>  
<http://www.webmin.com/webmin/>  
<http://www.kneschke.de/projekte/ksamba/>  
<http://www.linuxprinting.org/howto/>  
<http://www.cups.org/support.html>  
[http://www.webmin.com/screens/edit\\_printer.html](http://www.webmin.com/screens/edit_printer.html)