Windows–LAN–Server–HOWTO

Table of Contents

Windows LAN server HOW−TO

by Ryan Cartwright, `crimperman@enterprise.net`

v0.1, 21 September 2000

This document is intended to assist those who wish to consider Linux as a server within an office environment which has PC's primarily running Microsoft Windows 9x.

1.Dedication

2.Introduction

3.The Linux Option

4.Installation

5.Is that it?

6.Conclusion

7.References

1.Dedication

This document is dedicated firstly to Jesus Christ, my Lord and Saviour, thanks to Him I have the ability to do this. It is secondarily dedicated to the authors of the various utilities and documents referred to here. Thanks to them I have the tools to do it.

2.Introduction

Linux is gaining increasing popularity within the workplace. Primarily it is deployed within the Internet marketplace at server level but it is beginning to make in roads into other areas such as internal network servers and desktop workstations. With this in mind and for reasons given below, my company decided to deploy a Linux based LAN server into our Windows9x based network. I started this project with basic knowledge of Linux and some knowledge of Unix. During the course of the project it ocurred to me that some sort of document describing the tasks involved would be helpful. I could not find such a document and hence wrote this one.

What you will not find here is a repeat of installation and configuration documentation for the various tools and utilites used. I see no reason to repeat that but have instead opted to include problems encountered whilst installtin or configuring these and workarounds/soultions for those situations.

2.1 The Scenario

It will probably be helpful to give a short background of the environment in which the new server will be deployed.

Some 35 PC's are linked in an Ethernet LAN across a sprawling site. Like many offices this one started with a single PC and grew bit by bit into the current environment. For reasons of speed, convenience and cost a peer−to−peer network was employed. Users share directories and printers across the network using share level access.

One of the PC's became designated as a "server" (*from here on I shall refer to this as the "serverPC"*). Peer−to−peer networks have no server as such and thus this PC was identical to the others except that it had no consistent user. It was used to store common files (templates, small database files etc.) for use by all users and also contained the Microsoft Mail postoffice directories for the internal mail system. Networking faxing was also routed through this PC by means of Microsoft Fax and more recently, internet e−mail distribution was added by means of a mailserver utility which connected periodically to an external mailserver and redistributes the mail accordingly. It also shares a printer for use by the majority of users in the vicinity. The client side of the mail and fax systems was handled by Microsoft Outlook.

Increase of traffic through the serverPC especially internet mail increased to the point where file access slowed and users could not always log onto the internet mailserver. At first the internet mailserver program was suspected but further tests proved this to be untrue. Users were becoming increasingly frustrated and subsequently handing these emotions onto the IT support people.

There was also a secondary issue to consider. Having a designated serverPC meant from the management viewpoint a perfectly good PC was "doing nothing" because nobody was sitting at it. A decision was made to allow occasional users to use this PC as a workstation. The PC would sometimes lock during these occasional uses as a workstation meaning the loss of access to important files to all users while it was rebooted and subsequently database and file locks would need to be cleared to allow users to get back to their data.

2.2 The Options

The situation called for some kind of remedy. At the most basic level the options were simply "repair or replace" and as is often the case, there was funding limitation.

Repair

Repair is at first glance the quickest and cheapest option but it is rarely easy, especially if you are unsure of the exact cause of the problem. As a workstation there was nothing "wrong" with this PC but as a server it often seemed overwhelmed. The situation could have been partially solved by installation of a network switch to speed the network traffic but could have possibly resulted in creating a bottleneck at the serverPC as it struggled to keep up with traffic demand. The PC was running Windows98 which as a desktop environment is perfectly adequate but as a server starts to struggle. At best it was considered that this option would only postpone the problem for a while especially if network growth continued.

Replace

Replacing the serverPC with a dedicated server and establishing a client−server relationship would allow for the expected increase in network size and traffic. Traditionally a dedicated server would involve some considerable outlay as the options here were either WindowsNT or NetWare. Recently Linux has come very much into the spotlight and it provided an alternative replacement strategy.

3.The Linux Option

Linux is a Unix clone and as such the incorporates the excellent netwroking abilities of the latter. It is this trait (among others) which has lead to its' increasing deployment in the Internet server market. It could provide a low cost replacement strategy for the problem at hand and yet allow for the expected network growth at little or no extra cost.

That Linux was an effective and cost−effective server solution was not in question but we need to know whether it could provide a specific solution in this case. Could Linux fulfill all the roles provided by the current serverPC, including file−serving, internal mail, network faxing and Internet mail redistribution. Initial enquiries showed that it could and so the question became less of "can Linux do it?" and more "can I make Linux do it?".

3.1 Research is the key

Before presenting any argument for deployment to management it seemed prudent to research said argument. This would serve the additional purpose of educating myself in the finer details of Linux Administration. My Linux experience stemmed from a few months use at home and as Linux was not in use within the company I was to all intents and purposes the Linux expert.

I started my research lurking at newsgroups, particularly uk.comp.os.linux (u.c.o.l.).Although lurking can be frowned upon in some circles it is something I recommend in early stages of a project like this. Reading other peoples questions and answers gives valuable insight into your approach to future projects you may encounter. They say it is a fool who does not learn from others mistakes. In addition I had a copy of the book "Learning RedHat Linux" published by O'Reilly (http://www.ora.com). This book was used when installing my home version of Linux and is excellent for this purpose. It also contains a very significant chapter on Samba – a networking application which allows Linux to act as a fileserver for Windows9x PCs. I also made extensive use of the Linux Documentation Project (LDP – http://www.linuxdoc.org) especially the Linux Users Guide, the System Administrators Guide and the Network Administrators Guide.

The importance of further reading

I cannot stress enough the importance of the research to the outcome of the overall project. There are many phrases and anecdotes which accurately summarise this, including "forewarned is forearmed" and the five P's (proper preparation prevents poor performance).

3.2 The tools

My initial research revealed the direction I should go and what specific programs I should learn more about. These included:−

- Samba (for file and printer serving),
- qmail (for mail delivery – MTA)
- fetchmail (for collecting Internet mail from our ISP mailboxes)
- mgetty+sendfax or HylaFAX (for faxserving)

Although there were alternatives, these were the most recommended and a quick question to u.c.o.l. confirmed these as good choices. I was aware that network fax serving could be done and that tools were available – articles in Linux Journal helped as did advice from u.c.o.l. users.

3.3 Convincing the boss

This proved to be one of the most anxious tasks of the early stages. It was one thing to bring myself to the realisation that Linux provided the best solution and quite another to consider guiding my boss(es) to the same conclusion.

Although there was virtually no outlay cost involved (always a good stumbling block to remove) there was the matter of time. The project would involve certain amounts of time for me to learn as I went and this in turn would involve a longer overall timescale before the solution was in place.

The temptation was to point out the faults of the existing solution and then present the Linux proposal as an all conquering hero. This was unlikely to work as it could have been interpreted as me pushing a solution simply because I liked the idea. In addition had I presented this argument any delay (or pecieved one) in deploying the Linux server would be harder to explain. I had to present my argument as a benefit for the company. To this end I could use the existing problems but I had to be careful to avoid a "Linux for Linux sake" point of view.

As it happened all my concern was for nothing − during a conversation about the existing server the IT manager suggested the very solution I was about to argue for! However he did require some reassurances which were all along the lines I have discussed here. Your situation will of course be different but in any case it must surely be beneficial to present as objective an argument as possible.

3.4 Which distribution?

I chose to use RedHat 6.0 for this project. This was down to a very simple reason − I already had a copy and could therefore get started quicker. Also I was used to it as I had been using it at home. I can see no real reason why in this case one distribution should be used over another except for personal preference. There are some server editions of several distributions and again use of these is in the realms of personal preference. I have limited experience of various distributions and thus feel inadequately qualified to make a recommendations, my advice would be that you may want to eliminate as many unknowns as possible and thus learning the nuances of a different distribution may cause further hindrances.

4.Installation

4.1 RedHat

I assembled a PC from bits lying around the IT stores − a fun exercise in itself − and ended up with a test system of P133, 32MB Ram and 540MB HD. I was planning on replacing the HD with a much larger one but wanted to test the installation on the rest of the system first.

Having installed RH6 before a few times I *knew* this would be a breeze...I believe "famous last words" is the phrase I am looking for here! Installation seemed fine but on 1st boot (and subsequent ones) I encountered "invalid compressed format" errors as the system tried to Uncompress Linux. This evolved into a system that hung at boot with a "LI" prompt and a few questions on UCOL highlighted this problem as being a drive geometry problem. The system could boot from an MSDos bootdisk launching Linux from LOADLIN but this was far from acceptable. A 1GB hard disk was used instead.

A secondary problem was the NIC. The one I first used was a Realtek8019 ISA card, this is an NE2000 compatible card and thus *should* use the ne2000 driver. After much trying and even a kernel recompile the card refused to work with said driver, so I swapped it with a D−Link DT−530 PCI card from another PC. This card was reported to work with the 'tulip' driver. However the RedHat install procedure could not detect it. A quick look on the D−Link website pointed to the latest via−rhine driver as a solution. This was downloaded and compiled and installed along with the pci−scan driver file from the same site (http://www.scyld.com/network/via−rhine.html). This site also contains excellent installation notes. With the new drivers in place the machine was up and running and a few ping tests proved the NIC was running fine.

4.2 Samba

Version 2.0.3 was installed as part of the RedHat installation and because this was a trial run I saw no reason to download the latest (at time writing this is 2.0.7). smbclient was not installed as there would be no reason for the Linux box to access shares on the Windows PC's. Configuration was a breeze thanks to the SWAT utility which is accessed by pointing a web browser at port 901 (ie: http://localhost:901). I was even able to access and configure this from one of the Windows boxes across the network (http://<ip address>:901).

Samba configuration

For some reason our users have a habit of not exploring the network beyond their workgroup − even though they often can. To avoid confusion on their part and to keep accidents to a minimum the server was put in it's own workgroup. There are much excellent documentation on setup and configuration of Samba and thus I refer the reader to those rather than repeat the information here.

As our PC's all have static IP addresses and users are primarily seated in front of the same PC every day I opted for the share security option in Samba. This has the danger of leaving resources open for anyone browsing the network so I also employed the hosts−allow feature in the globals section. This was restricted to those on our network using a partial IP address. Shares were enabled pointing to various directories all under a new /resources directory.

Microsoft Word templates

All the shares worked fine except when it came to templates for MS Word97. Word has a feature where you can set a Workgroup Templates location in its options. The problem was that if that pointed to a Samba share, the share could not be at top level (ie: //SERVER/template). When you clicked File|New, MSWord would report that it could not open the templates in the location selected yet you could open a template from that location through File|Open. This was further confused because you could navigate to said top level share in Explorer and double click on the template file and Word would create a new document based on your selected template. The workaround was found to be as simple as sharing the parent of the template directory and setting Word to look through that path (ie: //SERVER/resource/template). Despite much amending of file permissions and usernames it seems this is the only way to get this to work. I remain unsure as to which end causes the problem, Word seems a likely culprit (because everything else can use the files okay) but Samba can also be pointed to (because a Windows top level share will work in Word).

4.3 E−mail

qmail was chosen as the Mail Transport Agent (MTA) over sendmail which was supplied with RedHat. This is primarily because the former has a reputation for easier configuration and better security than the latter.

qmail

The latest source files were downloaded via a mirror of http://www.qmail.org and compiled and installed. There is plenty of documentation supplied with qmail but I chose to also use Life With Qmail (http://web.infoave.net/ dsill/lwq.html). This document is similar to a hoot and was probably the most useful document for our purposes.

Qmail installed easily enough but I encountered a few minor problems with using it. I configured it, for performance and reliability reasons to use Maildir as the default delivery. The good old standard mail program does not recognise this type of delivery and thus it took me a while to figure out why my mail was being sent but I could not see it. The solution was to use mutt (http://www.mutt.org) which does support Maildir. Of course this was a minor problem as the users would not be using the Linux box to read their mail but rather get it through a pop client (MS Outlook) on their Windows workstations.

fetchmail

Fetchmail was used as a collection agent and installation and setup of this was a breeze, especially when I found out about fetchmailconf :o). We do not require mail collection at all times but prefer to collect at set intervals. To facilitate this fetchmail is called using the −d switch by a cron job everyday and stopped by another one.

We collect our mail from ten mailboxes on our webhosts server, one of these is a bulk redirect where anything addressed to our hostname but not to one of the other nine specific addresses is deposited. fetchmails multidrop facility was employed to allow us to download all the mail from this mailbox and then smtp it to qmail using the intended recipients address. One problem we encountered was sending mail from our new qmail server to our salespeople. They collect their mail direct from the webhosts yet their domain is the same as everyone else's. This meant that every time a local user tried to send a message to one of the salespeople, qmail tried to find a local username to pass the message to and, upon finding no matching user, bounced it to the postmaster. The solution was to use a secondary e−mail address for the salespeople. Our webhosts do not provide dial−up services so our salespeople each have their own free ISP account to get access to the web. This account provides them with an address on a different domain and so qmail was able to forward all mail for them to this address using the alias files.

Note: To make life easier for the salespeople our webhosts redirected all mail coming to their mailbox for these people to the free ISP mail address − this meant the salespeople weren't 'confused' by having to juggle multiple accounts and addresses on their notebooks − bless 'em :o).

4.4 Faxing

The old serverPC used Microsoft's network faxing to share a fax service across the network. Users then used MS Word templates (which had VBA macros) to create and send faxes automatically, errors were mailed to the user. To provide and equal if not better service on the new server I chose mgetty+sendfax to provide the local faxing service. This installed easily and I was soon able to spool faxes from the Linux server. Spooling from Windows clients was to prove a much tougher nut to crack and resulted in a change from the original choice.

Faxing from Windows clients

The previous arrangement shared a fax modem from the serverPC using Microsoft Fax under MS Outlook to provide fax services to all Windows clients. Further to this we used a standard Word97 template which had a macro attached for automatic sending of faxes. Utilising the Sendfax VBA command, this macro meant users had only to fill in the template and hit the "fax now" button on their Word toolbar to send a fax. They didn't have to deal with any third party programs which asked them to repeat everything they had just typed into the template. This arrangement thus provided seamless faxing to the user and it was one I was keen to continue.

Ideally what I wanted to do was have some way of passing the intended document, the username and the fax number to faxspool on the Linux box from the Windows client applications. The traditional way to provide fax services to any Windows app is to setup a "printer" which points to the fax modem.

HylaFAX

Originally I installed mgetty+sendfax to use as a fax server. This was primarily because it is supplied with RedHat 6 and so was readily available. Unfortunately it proved to be unsuitable for our particular use as we required some way of sending faxes to the faxserver using Microsoft Word macros. There are some excellent Windows clients for mgetty+sendfax but alas they all require the user to enter the fax number etc. each time a fax is sent. I wanted a solution to match our current one where the user fills in a Word template, hits a button and the macro reads the fax number from the document and uses the VBA Sendfax command to send the fax via MS Fax.

After much deliberating and searching I was pointed toward HylaFAX (http://www.hylafax.org) which has a windows client WHFC (http://www.transcom.de/whfc/). This client allows for communication through VBA macros which was exactly what I wanted. Hylafax installed okay although I had some rather annoying client access problems. These were solved by ensuring the client IP addresses were correctly added to not only /var/spool/fax/etc/hosts (as indicated in the man pages and FAQ) but to /var/spool/fax/etc/hosts.hfaxd. Once this was done I was up and running in no time. WHFC installed very easily and was set−up in seconds.

Word macros

As mentioned, our users are accustomed to being able to hit one button to send a fax document from within MS Word97, it was important to keep this feature available with the new server. WHFC has OLE capabilities and thus we were able to write a new macro which allowed the user to send a fax from within Word without having to enter the fax details into a secondary popup box. The macro does two things − first it prints the current document to a file, then it uses WHFC's SendFax OLE function to send the printed file to HylaFAX. The printer driver we use is the Apple Laserwriter 16/600(ps) one as recommended in the WHFC setup notes.

Here is the macro code we use ...

Sub Spool_fax() *' Spool_fax Macro ' Macro created 09/08/00 by Ryan Cartwright*

Dim givenfax, realnum As String Dim whfc As Object Dim OLE_Return As Long Dim Box_Return As Integer

' First we print the document to a local file − note that Background must be false ' otherwise the function will return before the file is entirely written ' and thus HylaFAX will be unable to convert it properly. Application.PrintOut FileName:="", Range:=wdPrintAllDocument, Item:= _ wdPrintDocumentContent, Copies:=1, Pages:="", PageType:=wdPrintAllPages, _ Collate:=True, Background:=False, PrintToFile:=True, _ OutputFileName:="c:\faxtemp\printout.ps", Append:=False

' In our template the user is asked by an Autonew macro for the faxnumber etc. ' the field number for the faxnumber is 8, we need to retrieve this ' and add a 9 to the front of it Set givenfax = ActiveDocument.Fields(8).Result realnum = "9" + givenfax

' Now we create the OLE object and call the Sendfax() routine Set whfc = CreateObject("WHFC.OleSrv") OLE_Return = whfc.SendFax("c:\faxtemp\faxoutput.ps", realnum, False)

' Finally we test the returned value and report accordingly If OLE_Return `&`<= 0 Then Box_Return = MsgBox("Error sending file", 16, "FAX Not Spooled") Else Box_Return = MsgBox("Fax Job ID:" `&amp` OLE_Return `&amp` Chr(13) `&amp` "You will be notified by email if it was successfully sent", 0, "Fax spooled") End If Set whfc = Nothing End Sub

5.Is that it?

That pretty much covers the installation and configuration of all the tools and utilities required to get our new server up and running. Having said that there is more to a good server than just the tools to do the job required. I advise you read the afore−mentioned Linux System Administrators Guide especially chapter 10 – backups!

6.Conclusion

The Linux server was in place some two months after starting this project. I am sure this could have been days if I had known what I was doing but I would recommend to anyone considering a similar project to allow themselves a good time period. This is especially applicable if like me they have cut their support teeth in Windows. Linux is not difficult to use, just different and the transition from Windows takes time. Read the excellent documentation around before you start and also you may find it more beneficial to try this step by step on a secondary machine whilst your old one is still running elsewhere. Migrating is a better approach than a straight swap.

7.References

- Linux Documentation Project – http://www.linuxdoc.org
- Freshmeat – http://www.freshmeat.net
- qmail – http://www.qmail.org
- HylaFAX – http://www.hylafax.org
- Samba – http://www.samba.org