Better e-mail management

Mail Servers

If you value your email, you will want to have full and total control over it. Relying, totally, on the email provisions of your ISP may not be enough. Here are the details that will put you on the right road to setting up your own e-mail servers. BY COLIN MURPHY

What's wrong with having clients taking mail? Mail handling should not be the responsibility of a desktop application. Letting Kmail or other client you use take your mail down from an ISP is missing the point of running a Linux system.

If you are working with other users on a network, or even a single workstation with multiple login's mail can get lost or misplaced into the wrong account.

It is even more impractical to use only desktop clients if you are looking to collect and distribute mail for a group of people, and more so again should you be thinking of doing this through your own domain name and not that of your ISP.

You will find advantages also if you get lots of daily emails, because you can use better filtering facilities to control how your email is presented and prioritized, and you will have more control over how you can maintain backups and mergers. You will be able to control spam better, using applications like **SpamAssassin**.

If you are using a stand alone system as a workstation you won't need to worry about taking on the responsibility of running your own email server, but if you are responsible, you may very well want to.

There is nothing more annoying and frustrating than finding that you are being denied access to the most fundamental internet services, your email, just because your ISP hasn't made enough provision when their email server fails you. If you run your own mail servers then you have access to an alternative system.

Responsibility may not be your only issue, you may also find yourself wishing for better features, not present in your email client. It may not have the filtering support you need, for instance.

In this article we will give you some highlights and guidance to how you might improve the email provision on your machine.

How email gets from A to B

If you are going to take a more hands on approach to how email is handled on your machine, you will need to start out with an understanding of the processes that it goes through to get to and from your machine. Unfortunately, for simplicity's sake, there is no one chain of processes to follow, it depends very much on how much control you are willing to pay for and who is providing you with your email addresses.

At the beginning of the process if you are sending an email, or at the end if you are receiving, is the mail user agent (MTA). This is the client side application you use to read and write your emails with. See the Box: E-mail Clients, for a brief list.

The important thing about the MUA, from the mail servers perspective is not to handle the formatting of the email text or helping with its composition, or anything like that. Its fundamental roles are:

- To make sure that the email is in its own discrete block, so that it is still

Figure 1, An example of WebMin being used to configure and administer Sendmail

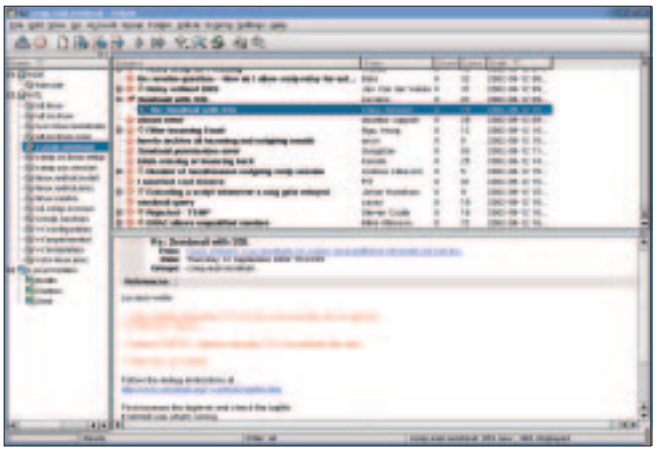

Figure 2: Newsgroups are always a useful source of information to ensure your Sendmail system is working

recognized as being your email once it has come out of the other end of the system, having been lumped together with lots of other emails, as happens in the next stage, with the help of the Mail Transport Agents (MTA).

- The creation of the header part of the email message, which contains the useful information, such as, where the email is going to and who to advise, should it fail in its mission.

Now that the email message has been composed and in its own discrete bundle, the act of starting to serve the email messages can begin, with the MUA passing this bundle onto the MTA.

The MTA routes messages between machines. There are many different MTA's available for Linux, and we will give a round up of the four most popular, Sendmail, Postfix, Exim and Qmail.

The MTA looks at the details in the header section of the email and decides how best to get the message to the recipient. If it is local mail, for local people it will pass it to the delivery agent on that local machine. Should the mail be for some far flung place, then it will try to pass the email on to the designated server as shown in the MX record of the recipients DNS entry.

It can also take some spam blocking actions as part of its work, confirming the authenticity of the email header and the route that it took

If it is not yet on a server in the local domain, another lookup is done to pass it on to the next server in line. If this server is in the local domain of the recipient, the the message is almost delivered and so passed on to the local Delivery Agent.

Sometimes this local Delivery Agent role is handled by the MTA, but there are other options. Procmail is the Delivery Agent of choice for Linux users at the moment. This is where the receiving side gets some control as to what happens to the email from here, both as system administrator and, ultimately, the email's recipient. In its simplest form the email will just add the email received to the mail spool that is to be used later by an MUA. Many do more with it though, getting their mail grouped and filtered and even have applications run on receipt of certain emails.

Popular MTA's

There are many different types of mail transfer agent available for Linux. Here are four of the most popular, open source, MTA's in use today.

Sendmail

Being the first MTA, having a legacy that reaches as far back as 1981, Sendmail is considered to be the de facto standard for mail transport applications. Sendmail powers four times more internet mail domains world-wide than its largest competitor. Eric Allman wrote Sendmail right at the beginning of what we now know as the Internet when it allowed the routing of email between ArpaNet and the University of California at Berkley. Being Open Source, momentum soon increased its list of developers, allowing it to take, and keep, top place.

Being the front runner also makes it first in line for criticism, and these usually revolve around complexity of configuration and security. While I'm sure that security was an issue back when Sendmail was first developed, surely they could not have conceived how the Internet is being used today and so, must always be playing catch up with any new security threats that come along. Patches come through very quickly, covering over any security flaws that are found. Because of this, some people in the industry consider Sendmail to have a low security rating. The alterna-

E-mail clients

There are many to choose from, from the delights of *Kmail* and *Evolution*, the default MUA's for KDE and Gnome users respectively, and include the email clients that come as part of a bigger package such as in *Mozilla* or *Netscape*, for those who prefer to have their email and web browsing functionality lumped together in one application. At the moment, the most popular to try out, of the graphical email clients, is *Sylpheed*, with its claims of speed and small memory footprint.

For those that do not want to, or cannot afford to have the impact on memory of running a graphical client, there are many console based email clients such as *pine*, *elm* and the ever popular *mutt*.

At the very bottom, some might say top, of the tree, stands *mail*, father to them all.

The choice of which you choose really depends on how much control you want to take over its configuration and who you want to have that control. The more graphical, the less control the end user has.

tive view is that, because you constantly have to work closely with Sendmail, constantly updating and adding security patches you are much closer to the problem of security, more aware of the problems, with a better understanding of the risks and are more secure as a result.

Installation of Sendmail is very easy, and it is most likely to be provided as the default email server for most distributions, Debian being the most notable exception, but is easy to get hold of, thanks to apt-get. Configuration is a different matter and many consider this to be Sendmail's greatest weakness. Sendmail's configuration files are designed to be easy for Sendmail to deal with, leaving the configurator to suffer in its complexity. Thankfully, some efforts have been made to make this much easier and most users will use a set of macros called *m4*.

Sendmail offers immediate support for the **IPv6** protocol.

Much of the challenge of configuration has been removed by applications like WebMin, figure 1, which allows you to configure and administer Sendmail, amongst many other applications, through a web interface, remotely if needs be.

With its long legacy, Sendmail does have backing and a range of professional and enterprise products and services

Exim

Exim has been developed at Cambridge University by Philip Hazel since 1995 and released under the GPL. It is the default mail server included with the Debian distribution. Taking what he had learnt from the Smail project, one of the earlier GNU MTA programs to have been developed, Philip Hazel wanted to keep the same lightweight approach, but add a great deal more functionality.

Its interface is that of Sendmail, so it can be used as a drop in replacement for a system that has been installed with Sendmail. The current version 4.05 is a complete rewrite of the previous versions to benefit, as such, from the continual problem of patching code. This has allowed new features to be implemented making the overall structure more modular and the filtering scripts easier to modify.

The new Access Control Lists (ACL) handle the way Exim deals with differing SMTP commands. Previously the policy control was by a series of options that could cause some confusion due to their interactions. By using ACLs an administrator can now write his own tests and know which order, and so what interactions occur as they are performed.

The new version also does away with the directors which handle local domains by incorporating these into the routers which perviously only handles remote domains.

Exim again has native support for the **IPv6** protocol. Exim supports both Maildir and mbox file formats, convenient if you are migrating from other systems.

Scalability seems to have come naturally to the development of Exim. Even though it was not conceived as a high-performance MTA it has successfully been used on systems which have reported that up to 800,000 messages a day have been dealt with. The large ISP *freeserve.co.uk* use Exim for their email systems. Exim is not the most secure of servers available. It is written as a single binary which then has to be run, in effect, as root. This opens up the concern that, should a security exploit be found, there is a greater chance of someone gaining unauthorized root access. This is a remote possibility though.

Configuration is much easier than Sendmail, especially for those who do not want to give over their lives to keeping the system running. There are some very useful examples of how Exim can be configured to cater for single user systems to very large mail servers, these can be found at *http://sysadmin.oreilly.com/news/exim_0701.html*

Philip Hazel has also written the definitive book about Exim, details of which you will find at the O'Reilly site, from the above link.

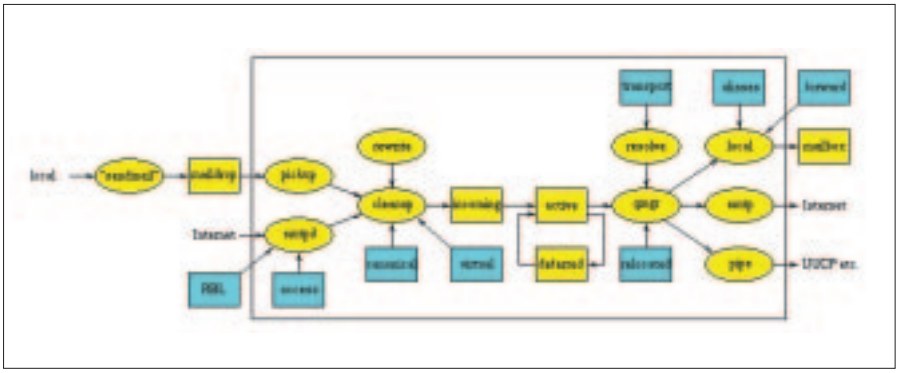

Figure 3: Here the anatomy of the Postfix system is laid out for all to see

Postfix

Wieste Venema developed Postfix while he was working for IBM, released in 1998 as the IBM Secure Mailer and then released to the wider community under the IBM Public License as Postfix.

Realizing the strengths and popularity of Sendmail, Wieste developed Postfix to maintain compatibility with it as much as possible, while also ensuring Postfix would be faster, more secure and much easier to configure. At the moment, Postfix does not offer inbuilt support for IPv6 but this can be added as a patch. You have access to both Maildir and Mbox email file formats as well as all of the Sendmail file layout, utilizing */var/spool/mail*, */etc/aliases*, and *~/.forward* files, etc.

Postfix has multiple layers of defence to protect the local system against security breaches. There is no direct path from the network to the security-sensitive local delivery programs – an

GLOSSARY

IPv6: *The internet works because of IP addresses, those dotted quads you see, such as 192.168.0.1. Unfortunately, the world has run out of addresses like this, which are part of the IPv4 numbering system. The solution is IPv6 which will use an address length of 128 bits as opposed to the 32 bit addresses currently in use.*

As an example of how many more IP addresses will be available after the introduction of IPv6, if you said that all of the address available with IPv4 was just 1 millimeter long, then the address space available from IPv6 would equal 80 times the diameter of our galactic system.

intruder has to break through several other programs first. Postfix does not even trust the contents of its own queue files, or the contents of its IPC messages. No part of the Postfix program needs to be run as root set-uid.

Performance wise, Postfix is very fast, with claims of being up to three times faster than its nearest competitor.

Postfix has some very useful Unsolicited Commercial E-mail (UCE) control features, needed in these days of spam. You have control over which hosts can relay messages through your system implementing things such as black lists and DNS lookups. Content filtering is not available, for that, you will need to use a product such as SpamAssassin.

The Postfix website contains lots of useful information to get this mailserver up and running, figure 2 shows an example, which is explaining the anatomy of how it works.

Qmail

Qmail is the only mail server to offer a cash reward for anyone able to prove they have found a security hole in Qmail. In March 1997, D.J. Bernstein, the author, offered $500 to the first person to publish a verifiable security hole in the current version of Qmail. No one has won so far.

The Qmail license does not allow the redistribution of modified qmail source code packages. The upshot of this mean you are unlikely to find it being installed by default in any of the mainstream distributions. Qmail uses its own method of installation, which, while simple to do, does not sit comfortably with those who like to keep their personal RPM databases accurate.

Qmail doesn't seem to have been updated in some time, the current version 1.03 was released in 1998. It does feature support for IPv6 via a patch, and many other features are available as patches, for which the other Mail Servers have included in their current version, like support for Authenticated SMTP and backend access for LDAP.

The danger of running an open relay

As part of the route that an email must take to get from A to B it will need to pass through, or be relayed by, other mail servers. Imagine the case where your local mail system, *home.com* is sending out a message for someone at *away.com*.

The first hop that your email might take could be from your own local network to the email server of your ISP. From there the email is relayed to mail servers that are nearer the final destination, say, the incoming mail server of the recipients ISP. In this chain of relays there is an obvious reason for the connection between one machine and the next.

It used to be the case that mail servers would happily relay messages from one server to another. As part of the design of the internet, where there is no 'fixed' route to go from one server to another, only the most convenient route at the time of transmission, a mail server might have found itself with an email following some obscure path. It was quite happy to pass on the email to a server nearer the destination, in order to keep the system running as smoothly as possible.

This open, good natured approach to running email throughout the Internet was, of course, going to be abused by people keen to make a fast buck. Someone sending spam could directly address their emails to go to a mail server and, by forging the information in the header of that email, they could be confident that their mail would be passed onto the system for eventual delivery to someone who did not really want to receive it. The mail server was open to pass on any mail, which also meant it was open to abuse. It was an Open Relay.

The promiscuous relaying of email was the default mode for *sendmail* prior to version 8.9, and seeing as it powers the largest number of mail servers, the possibilities for abuse were everywhere.

Now is is deemed necessary to have much tighter control over what you allow your mail servers to relay. The need for servers to relay messages is still there, but only for well defined cases, which now need to be configured for. The most common case being that where a mail server is also acting as a gateway to some other network which may not have a permanent connection.

If you are going to run your own mail server, you take on the responsibility for making sure it cannot be used as an open relay. The scourge of such relays is enough to prompt the creation of 'blacklists', such as the one at *http://www.ordb.org*. These 'blacklists' can be accessed by mail servers directly while they are handling a piece of email. Should that email have followed a route that includes one of these 'blacklisted' mail servers then the email is handled with extreme prejudice, usually being bounced back to the originator, with a note saying why. So, should you be administering the mail server used by others and it becomes 'blacklisted', because it has been poorly configured, your users are going to come down on you like a tonne of bricks, wanting to

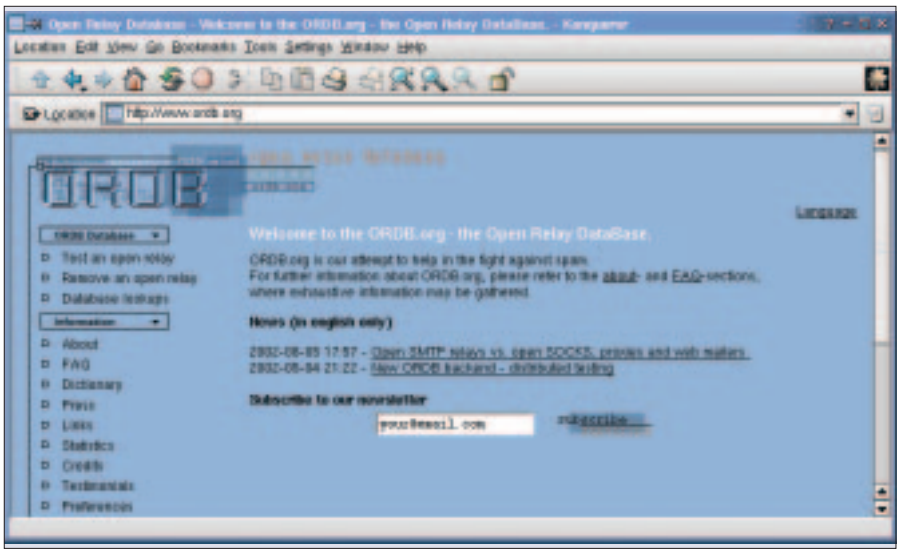

Figure 4: The Open Relay Database site whrer you can get help for fighting spam

know why their email is not getting out. Best to get it configured right first time.

For many though, this is not likely to be a problem, as all of the recent mail servers will come with relaying option turned off.

SpamAssassin

UCE, also known as spam, is the scourge of the earth I'm sure you will agree. Direct marketing via email might have been alright if those doing the marketing had the ability to be a little more selective to those that they have decided to spam, with far too big a percentage offering products and services of a most unsavoury nature.

Users of small, home networks have to accept the responsibility of stopping UCE being delivered to kith and kin. Administrators must equally take on the responsibility to stop company staff receiving UCE, the deluge of spam can mean a real waste of time and there is the danger that bona fide email will get lost in amongst it all.

What tools are available then? The most spoken about at the moment is SpamAssassin, which works as a mail filter, analysing each mail for common attributes to be found in spam. These attributes include searches for certain well known phrases in both the header and body section of emails, phrases like "MAKE MONEY NOW!!!" and counting the number of times a '$' is repeated. The more it is repeated, the more likely it is spam. There are also tests done to the header section of an email to add credence to its validity, many spam emails will have some element of the header forged. A ruleset is built up of these attributes.

Each element in the ruleset is given a weight of importance, a number of points. These points get totalled as the elements are found in the email. If an email receives enough points it will be deemed worthy of being called spam. You as administrator can fine tune the points per element and the threshold value to suit your needs if necessary. The ruleset that comes as default seems to catch most cases.

This is good, but it gets even better if the rulesets are shared. Now, when a new piece of spam is written which happens to slip through SpamAssassin's net, a new rule element will be written to block it in the future and propagated out to everyone who is registered to use this shared database.

With any luck, most of the people taking advantage of this distributed ruleset will never be inconvenienced by this new piece of spam because the new ruleset will be in place before the message reaches them for the first time. The Razor project, *http://razor.sourceforge.net*, helps to provide this distributed database of rulesets which SpamAssassin can then call upon.

SpamAssassin will also refer to some of the 'blacklists' that are available, like *http://www.ordb.org/* and *http://mail.abuse.org/*. These 'blacklists' contain details of Open Relays which are very often used as conduits for sending spam.

Buying domain names and access to MX records

If you are going to run your own mail servers, you will want a domain to call your own from which you can send email and be more easily remembered because of it, so people can send you email. You may not value the domain name give to you by your ISP, usually because it is just downright long winded or clumsy – *colin@murphy.org* is far snappier than *colin@murphy.name-ofisp.co.uk*, for example – ONLY!

Getting your own domain is simple enough, just enter the phrase *'domain registration'* into your favourite search engine to find a domain name registrar and have your credit card handy, to pay for the privilege. Be warned though, you need to take care to make sure you get the amount of control that you want and need. The most important being the control needed to get your email to do what you want.

The amount of control you have revolves around how much access you get to changing the details in your DNS entry. This is done via some web front end to your registrar's administration program. For full control over how your email is handled for this domain you need to be able to amend the MX record in the DNS entry, because you need them to point to your mail server. Listing 1 shows the output from a dnsquery.

Listing 1: Output of a dnsquery

```
colin@murphy.me.uk:~> dnsquery -n 192.67.202.2 murphy.me.uk
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 20266
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 2, ADDITIONAL: 5
;;      murphy.me.uk, type = ANY, class = IN
murphy.me.uk.       1D IN A       10.30.167.39
murphy.me.uk.       1D IN MX      10 mx0.domain-reg.co.uk.
murphy.me.uk.       1D IN MX      20 mx1.domain-reg.co.uk.
murphy.me.uk.       1D IN NS      ns.hosteurope.com.
murphy.me.uk.       1D IN NS      ns2.hosteurope.com.
murphy.me.uk.       1D IN SOA     ns.hosteurope.com.
hostmaster.murphy.me.uk.
(
                                  2002090710      ; serial
                                  8H              ; refresh
                                  2H              ; retry
                                  1W              ; expiry
                                  1D )            ; minimum

murphy.me.uk.       1D IN NS      ns.hosteurope.com.
murphy.me.uk.       1D IN NS      ns2.hosteurope.com.
mx0.domain-reg.co.uk.       4H IN A         192.67.202.235
mx0.domain-reg.co.uk.       4H IN A         192.67.202.237
mx1.domain-reg.co.uk.       4H IN A         192.67.202.241
ns.hosteurope.com.      13h21m13s IN A  192.67.202.2
ns2.hosteurope.com.     13h21m13s IN A  192.67.203.246
```

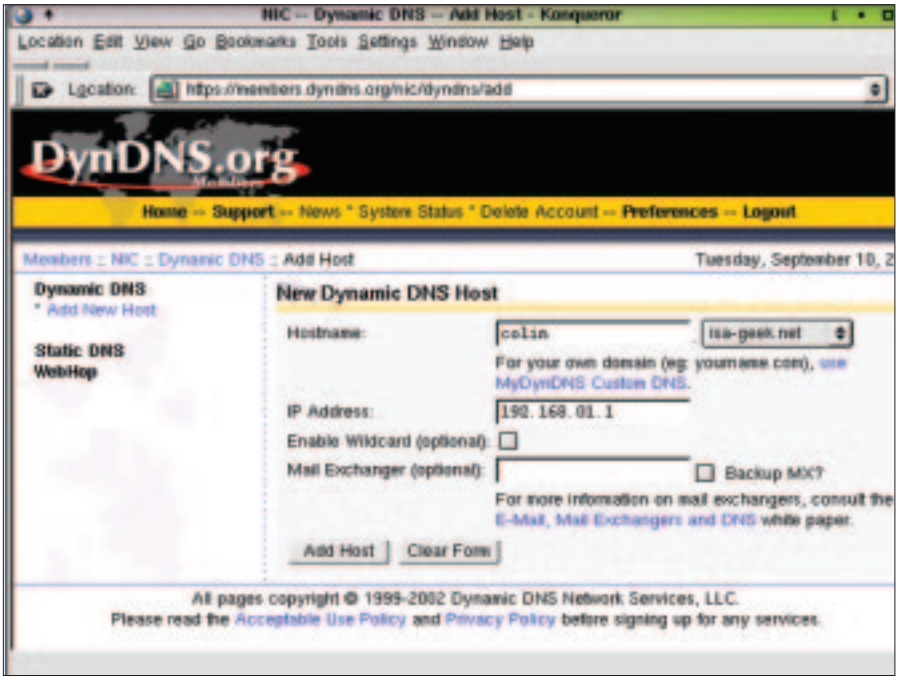

Figure 5. The web based administration tool used by DynDNS.

The dnsquery command allows you to see the detail held in the DNS database about a given domain, in this case *murphy.me.uk* , selecting the domain name server with the -n switch. Here you can see the the **MX** records firmly pointing to *domain-reg.co.uk*, the company used to register the domain. Unfortunately, for me, I discovered that you can not change the **MX** records for this type of registration, at least for this type of package – the cheapest available! On further research I could not find anyone that openly gave you the option of controlling this record. This means that the registration company will have control over any email sent to this address. There is nothing wrong with this, and they provide the facility to forward on mail to somewhere else, it's just that I still have to rely on their server not just my own.

Should this functionality be of a greater importance to you, then you will need to make sure that you can change the **MX** as you want. You will need to ask the domain registrars directly to confirm that they offer this feature. Alternatively, you may find the services of dynamic DNS providers useful.

No fixed abode

If you have control over your **MX** record, then you will want to configure it so that it points to the IP address of the machine you are going to run as a server. But what do you do if you don't have a fixed IP address because you are using a cable modem or similar.

There are Dynamic DNS services which will allow IP addresses to be mapped to other addresses, so you can have the IP address of your domain mapped to the IP address of the actual mail server. But the IP address of the mail server is still dynamic – likely to change at the whim and fancy of your service provider, so how does the dynamic DNS service know where to map the address to?

This can be taken care of by using one of the small clients that will, periodically, question your server machine about what IP it currently has and update this information automatically to the dynamic DNS service provider.

Some dynamic DNS services, like DynDNS offer a free service where you can register one of their subdomain. Now you can have an email address like *me@colin.isa.geek.net*, or something more serious if it takes your fancy like *me@colin.dyndns.org*, and have any mail addressed to it sent straight to your mail server. Figure 5 shows you the configuration screen of a newly created host which will map to the specified, and wholly made up, IP address.

If you want to use your own domain name and not a subdomain of DynDNS, then you can still do this, but there is a one off charge. This then gives you room in the DynDNS name server for the domain you had previously registered. Now you can expect email sent to your domain to reach you, even though you have a dynamic IP address.

Options

The option to control your own email is valuable in its own right, maybe the system you have in place at the moment works well enough. You only want to consider running your own mail server as a means of providing a back-up email service should you find your ISP's provisions have let you down. There is a certain amount of effort involved, how much depends on your circumstance. But you have the option, at least. ■

The importance of being MX

MX records in the DNS entry allow for email traffic being sent to the domain to be sent to some other IP address, the address of the mail servers. There can be more than one entry because you can have more than one server running as a fallback, each with different priority settings.

In the example above *mx0.domain-reg.co.uk* has a priority of 10, making it the first port of call over *mx1.domain-reg.co.uk* with its priority of 20. Should *mx0* not be available for some reason, mail will be sent to *mx1* instead. If the MX record is blank then it takes the information in the A record.

INFO

[1] Red Hat: *http://www.redhat.com*

[2] Starline Computer: *http://www.starline.de/produkte/easyraid/easyraid_x12/easyraid_x12.htm*

[3] Easy-RAID X12: *http://www.phertron.com/products/easyraid_x16/erx16_fc.htm*

[4] Cluster-Guide: *http://www.redhat.com/docs/manuals/advserver/RHLAS-2.1-Manual/cluster-manager*