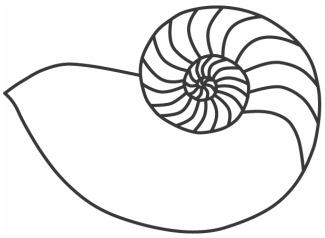

MUUGlines

The Manitoba UNIX User Group Newsletter

Next Meeting: September 14, 2004: Let's build a highly available web server!

How do you put together a redundant fail-over linux cluster suitable for use with Apache and MySQL? How do you solve the problem of real time data replication when you have a database that is doing frequent updates? Why MySQLs built-in replication doesn't work!

And, and a bonus, lets have it do some load balancing as well. After all, no sense having that second machine sitting there doing nothing, right? And all this for less than $3000.00.

Come see and hear John Lange speak on this feat of derring-do!

Tools used: Linux-HA, DRBD, openMosix, MySQL and Apache.

Where to find the Meeting

Meetings are held at the IBM offices at 400 Ellice Ave. (between Edmonton and Kennedy). When you arrive, you will have to sign in at the reception desk, and then wait for someone to take you up (in groups) to the meeting room. Please try to arrive by about 7:15pm, so the meeting can start promptly at 7:30pm. Don't be late or you may not get in.

Limited parking is available for free on the street, or in a lot across Elice from IBM, for $1.00 for the evening. Indoor parking is also available nearby, at Portage Place, for $2.00 for the evening.

Network Security Hacks

By Andrew Lockhart

ISBN: 0-596-00643-8

Reviewed by Shawn Wallbridge

This book took me a long time to read, but for a good reason, I kept implementing the various hacks in the book on a server I had started setting up.

The book is mostly Unix related, but there is some Windows related 'hacks' as well. I think the Windows coverage was lacking a bit though. For Unix, it talks about Linux, the BSD's and a bit on Mac OS X and Solaris. Most of the topics are general enough to apply to any Unix based Operating System, but some are specific to an operating system.

The hacks are all 'hyperlinked' to each other, if a hack mentions something that relates to another hack, it is highlighted in blue and the hack that it references is listed. I did find a few places where this wasn't done (***#84 Real-Time Monitoring***, first mentions Barnyard but doesn't provide any information on it or mention that it is one of the later hacks).

One of the great things about the Hacks series of books by O'Reilly is that the information is presented in nice small chunks that you can read in a few minutes if you have some spare time.

Lots of the hacks in the book could be found by doing some reading on the internet, but finding such a variety of topics all in one place, with enough

information to get you started is really nice. Even though I consider myself to be fairly security conscious, I still found quite a few things in this book that I hadn't thought of, or plain didn't realize were possible or even existed. I would recommend this book to anyone that is interested in security or anyone responsible for maintaining a server (whether or not it is on the internet).

More information is available on the O'Reilly website at http://www.oreilly.com/catalog/netsechacks/ which includes some sample hacks.

For a more detailed review of this book, see **http://www.geekbooks.net/review.php?isbn=0596006438**

Switching to vsftpd with Class

Or: How I Learned to Stop Worrying and Love the New File Server

By Gilbert Detillieux, 17 June 2004.

When Red Hat released their 9th and final free Linux distribution over a year ago, they dropped support for WU-FTPD, in favour of vsftpd, "probably the most secure and fastest FTP server for UNIX-like systems," according to the developers. The Fedora Project continued this trend with their Core 1 and 2 releases. In almost every way, `vsftpd` is an improvement, providing a more lean-and-mean server that promises better performance, tighter security, and more control.

The only draw-back for us at MUUG was that we had come to rely on WU-FTPD's "class" feature, to allow connections to be grouped into various classes based on the remote IP address or domain name. This feature allowed us to set different limits on total connections per host and class-wide, depending on where users were connecting. For example, since we are a Manitoba-based organisation, we wanted to favour those connections, and allow people on those networks to connect even if we had reached allowable limits in other classes, such as those networks in the rest of Canada, or in other parts of the world.

This provided a crude but rather effective way of allowing access to our members, even during busy times, without having to resort to non-anonymous access methods. Since `vsftpd` lacked an equivalent feature, we held off on migrating to it.

Enter tcp_wrappers

Fortunately, recent versions of `vsftpd`, such as the one in Fedora Core 2, and the updates for Core 1, include `tcp_wrappers` support as an option. Using the `hosts_options(5)` host access control language extensions in `tcp_wrappers`, it's possible to not only allow or deny connections based on the remote address, but also set options that will affect the processes handling those connections. Once such option is the ability to set environment variables.

VSFTPD_LOAD_CONF

According to the `vsftpd.conf(5)` man page, with the `tcp_wrappers` option enabled, "there is a mechanism for per-IP based configuration. If `tcp_wrappers` sets the `VSFTPD_LOAD_CONF` environment variable, then the vsftpd session will try and load the vsftpd configuration file specified in this variable."

The combination of the `hosts.allow` file (used by `tcp_wrappers`) and multiple `vsftpd` configuration files (specified via `VSFTPD_LOAD_CONF`) could be sufficient to implement a class-like mechanism, similar to the one in WU-FTPD.

Implementing the Classes

Under WU-FTPD on the MUUG server, users were divided into four classes, using class definitions similar to the following. (The actual definitions we use are more complicated, and include much longer lists of domains, but the simplified versions included here illustrate the point more clearly.)

```
class local anonymous *.muug.mb.ca
```

```
   *.umanitoba.ca

class mb anonymous *.mb.ca *.mts.net
   *.wp.shawcable.net

class ca anonymous *.ca *.shawcable.net
   *.rogers.com

class anon anonymous *
```

In our `hosts.allow` file, we would add the following lines to correspond to this.

```
vsftpd: *.muug.mb.ca *.umanitoba.ca : setenv
   VSFTPD_LOAD_CONF /etc/vsftpd/local.class
vsftpd: *.mb.ca *.mts.net *.wp.shawcable.net :
   setenv VSFTPD_LOAD_CONF /etc/vsftpd/mb.class
vsftpd: *.ca *.shawcable.net *.rogers.com :
   setenv VSFTPD_LOAD_CONF /etc/vsftpd/ca.class
vsftpd: ALL : setenv VSFTPD_LOAD_CONF
   /etc/vsftpd/anon.class
```

Note that in both cases, the first line containing a match for the domain is the one that is used. The first of our classes is for local users, which are those in the MUUG domain itself, as well as the University of Manitoba (which hosts the server). The second class is for users within Manitoba, the community MUUG serves (and home to most of our membership). The third class covers the rest of Canada, and the fourth, everyone else.

We now need to add the four class-specific configuration files for `vsftpd`. I chose to put them in the `/etc/vsftpd` directory, along with the default configuration file (at least that's the case for Red Hat and Fedora distributions). However, it's important that these files not have a `.conf` suffix, since the `vsftpd` init script starts a separate `vsftpd` for each such file. (This could be used, for example, to set up separate servers for each of a number of virtual hosts.) I arbitrarily chose to use `.class` as the suffix instead. (It made sense for our setup; your mileage may vary.)

Limiting Connections by Class

In WU-FTPD, we would limit connections by class with configuration options such as the following.

```
limit local 10 Any /etc/ftpmsgs/toomanyusers
limit mb  10 Any /etc/ftpmsgs/toomanyusers
limit ca 10 Any /etc/ftpmsgs/toomanyusers
limit anon 10 Any /etc/ftpmsgs/toomanyusers
```

This would allow a maximum of 10 connections per class, for an overall maximum of 40 connections. (We run a small server.) WU-FTPD tracks current connections separately by class.

With `vsftpd` and our class-specific configuration files, we could use the `max_clients` option to set our limits. There's a snag, though, in that all connections are counted together, regardless of these arbitrary classes we've created. We have no way of duplicating WU-FTPD's behaviour here, but we can get close enough for our purposes.

The reason for dividing our users into classes was to favour some over others, depending on where they're coming from. The main reason for this is to allow those users more local to us to connect without being overwhelmed by users that are coming from farther away (who could use other mirror sites instead). So, what we can do is allow each class a higher maximum number of clients than next class down. For example, if we set `max_clients=10` in the `anon.class` file, we would then set `max_clients=10` in the `ca.class` file, and likewise keep increasing our count by 10 for each higher class.

There is the possibility with this approach of a higher-class "starving" a lower class, if the number of connections continues to exceed the maximum for that lower class. (This was not a problem with WU-FTPD, since each class was allowed its own pool of connections.) Conversely, though, with this approach we may have better overall use of available connections, since we don't have to worry about some classes being underused while others are overused. Also, since `vsftpd` is much more efficient than WU-FTPD, we may be able to get away with a much higher overall maximum, without overloading the server.

Limiting Connections per Address

Another very nice feature in WU-FTPD was the

`host-limit` option, which allows us to limit (again, on a class-by-class basis) the number of connections allowed per client address. (OK, the author is biased, since he wrote the patch that implemented it.) Here we have a rather direct equivalent in `vsftpd`, with the `max_per_ip` option.

It's a good idea to set this option to something fairly low, to prevent so-called "download accelerators" from using up too many of your limited pool of connections. You could set larger limits on this for certain classes you can trust to play nice, such as local users, or to avoid blocking known proxy servers (where you may want to allow a larger number of connections).

Close Enough?

With the above tricks, we were able to get a reasonable setup on our FTP server using vsftpd. We can now be a bit more confident, now that we're running a more up-to-date, secure, and efficient piece of software, without worrying that our members and other "preferred clients" might get shut out due to an overwhelming demand for our server. With this, we're hopefully ready for the next wave, such as when Fedora Core 3 is released!

This article was written after the author tried to find similar information on the Internet, but couldn't, and after much struggling and experimenting with this setup, given the rather terse descriptions in the man pages.

This article and others like it are available on the tutorial section of the MUUG website.

http://www.muug.mb.ca/tutorials/

OpenIDS 1.3 is released!

OpenIDS a prepackaged Intrusion Detection System based on OpenBSD and Snort has been released.

New in 1.3:
* OpenIDS 1.3 is based on OpenBSD 3.5
* Updated Mysql to version 4
* Updated Snort to 2.2.0
* Updated Snortalog to 2.2.1
* Updated Oinkmaster to 1.0
* Updated the installation script.
* Added symon, symux and syweb.

It's available here:
http://www.prowling.nu/main/openids/openids.html

Rare, Collectible MUUG Shirts for Sale!

Don't miss out on the rare (only 35 made) and highly collectible MUUG Golf Shirts. We only have three left, one medium blue and two XL beige. They are $40 and are available at the monthly MUUG meeting. Get them while they last!

Sending Us E-Mail?

Due to the amount of e-mail MUUG receives, we've set up an auto-reply to give you immediate feedback, and redirect some of the e-mail to the appropriate places. Why not look at **http://www.muug.mb.ca/about.html#contacts** first?

Share Your Thoughts

E-mail us with your comments on the newsletter, whether it's criticisms or commendations, and continue to send in articles or ideas for the same. Specifically, what sort of material you would rather see: Announcements, technical articles, new products, or…?

If you have a How-To or other idea, and aren't ready to give a presentation at MUUG, an article is a great alternative! If you can write better than the editor, that's terrific; if you can't, submit it anyway and we'll get it into shape for publication. We know that many of you have some great ideas and lots of knowledge. Why not share? Send Mail to: **editor@muug.mb.ca**.