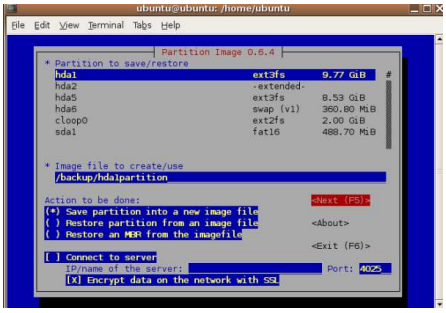
MUUGLines

The Manitoba UNIX User Group Newsletter

February 2010

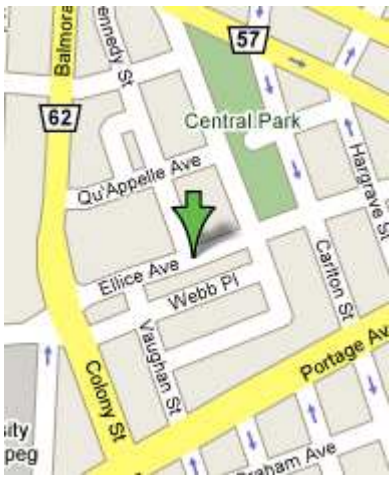
Next Meeting: February 9th, 2010



Adam Thompson will be covering disk imaging: who, what, why, where, when. Maybe even "how" if there's time! The presentation is expected to focus on the use of **Partimage** and related tools.

Before the break, as this month's **RTFM** topic, Sean Cody will deliver a remote, live presentation on a topic yet to be determined.

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. Please try to arrive by about 7:15pm, so the meeting can start promptly at 7:30pm.

Limited parking is available for free on the street, either on Ellice Ave. or on some of the intersecting streets. Indoor parking is also available nearby, at Portage Place, for $5.00 for the evening. Bicycle parking is available in a bike rack under video surveillance located behind the building on Webb Place.

Upcoming Meeting: March 9th, 2010

Next month, Kevin McGregor will discuss and demonstrate the use of a Marvell SheevaPlug, an ARM-compatible "plug computer" (i.e. a computer built into a wall plug power adaptor) running Linux. Kevin has been using one of these to implement a home-based remote backup server.

Before the break, we'll have our usual round-table discussion and RTFM topic. Details to follow.

We Think Your Friends are Awesome!

We Want To Meet Them!

Do you have a friend that is working on an interesting Linux/Unix project?

Why not ask them to present what they are doing at a future MUUG meeting?

Have them email **board@muug.mb.ca**

"So long, old friend…"

As of January 21, 2010, both the US and the European Union have approved the purchase of Sun Microsystems by Oracle. This was the last hurdle to the eclipse of one of the great creative companies closely allied to the widespread use of UNIX.

On Thursday, January 21st, James Gosling posted a simple blog entry with the above heading, no text, and the cartoon you see reproduced above. The blog entry can be found here:
http://blogs.sun.com/jag/entry/so_long_old_friend

James Gosling joined Sun Microsystems in 1984, where he created the Java language. He is Canadian and an Officer of the Order of Canada.

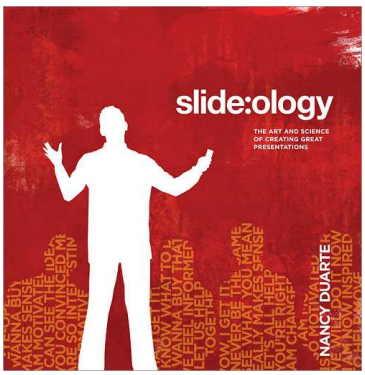
Book Review

slide:ology - The Art and Science of Creating Great Presentations
Nancy Duarte

Reviewed by Michael Doob

We now know all about the new Apple Ipad. It was displayed to the public, as usual, with great flair, even swagger, by Steve Jobs in one of his trade make fabulous presentations. What makes them so good? Why do they generate coverage by all media great and small? What are the keys to a great presentation?

The book under review tries to answer these questions by considering the slides and related media used. It proceeds with both didactic descriptions of good presentations and case studies such as Al Gore's "An Inconvenient Truth."



The essence of the book is to change the reader's viewpoint: a slide is not a written mini-document but rather an imaging tool for creating and transmitting ideas. Some of the chapter titles give the message:

- Creating Ideas, Not Slides
- Creating Diagrams
- Displaying Data
- Arranging Elements
- Using Visual Elements: Background, Color, and Text
- Using Visual Elements: Images

For the experienced presenter, many of the ideas will be self-evident. For example, don't put too many words on a slide. Nonetheless, there is much to be gained from this book for those at every level.

The book itself is an illustration of many of the concepts. It's really beautiful and quite unusual. For example, the entire book is in a light sans serif font (think Helvetica). Normally this would be dreadful, but the small line width and larger interline spacing balances with the font and makes it quite readable. Many wonderful illustration accompany the concepts.

If you make any presentations, even if only infrequently, this book is worthwhile.

The X-Box 360 & OpenBSD Can Co-exist

By Sean Cody

So over the holidays I received a new game that enthralled me, namely *Call of Duty: Modern Warfare 2*, a first person shooter that is actually quite fun when it

isn't frustrating. (I am particularly terrible at this genre of game.) After completing the single player campaign, I decided to move over to the multi-player component but that's where the love was lost. Like my experience with XBox games online, just finding people to play with was horrible: spotty coverage, long time-outs and general frustration. Almost ready to give up on the entire concept, I noticed on the screen a note stating my NAT type was "strict." When I saw that, my first response was "damn right it is," though it gave me pause for thought. I can't be the only person on the planet with a strict firewall and an Xbox, so there must be a means employed that I don't use and then it clicked: UPnP or Universal Plug and Play, which in my circle is known as the "firewall nullification protocol."

My firewall at home is a Soekris NET4501 screwed to the wall and running OpenBSD 4.5. (I've been travelling too much to keep it on –current.) OpenBSD and UPnP are pretty much polar opposites with respect to security and they have a sorted relationship where the end result was OpenBSD will never support or implement the protocol (for many reasons it would bore you to hear). The next step was searching the ports tree but that didn't give me any decent leads; so off to the colloquial last ditch search on Google, where I found a project named miniUPnPd (**http://miniupnp.free.fr**). This was exactly what I was looking for, as it directly supported OpenBSD and PF, was very lightweight and wasn't very difficult to get going. The nice thing about this particular product is that supports other OS's (*BSD,*Solaris, and Linux [with netfilter]) and now also supports NAT-PMP. Since this was a promising lead, I rolled up my metaphorical sleeves and got to work.

Installation was a breeze. After downloading the source tarball and extracting it, all that was required was a "`make && make install`" command to get the binaries in place. Setting up with PF was as simple as adding a few lines to my PF.conf. The slick implementation takes advantage of PF's anchor system (which in a loose analogy, is kind of like a file system of firewall rules). The two lines in question were "`anchor miniupnpd`" which sets up the anchor for the rules and "`rdr-anchor miniupnpd`" which setups the anchor for redirection rules. With those two lines in place, you should first check to make sure the new ruleset is valid by invoking the test option to `pfctl`, "`pfctl -nf /etc/pf.conf`" and then once you are sure it's copasetic, inject the new rule set with "`pfctl -f /etc/pf.conf`". Now your firewall is ready to accept firewall changes from the miniUPnPd daemon, so we just need to configure that and run it as root (due to it needing to modify firewall rules via the noted anchors). If you are not using *BSD and PF, then this setup will be different, but it is reasonably well documented in the "INSTALL" document which is in the root of the source package.

The configuration for the daemon wasn't difficult. The configuration file is simply `/etc/miniupnpd.conf` and the defaults mostly sufficed, though you will need to change "`ext_ifname`" to the external interface on the firewall (in my case "`ext_ifname=sis0`"), "`listening_ip=172.16.0.1/24`" (IP & network to listen on/for), "`enable_upnp=yes`". That should get you enough to get running.

Now as I mentioned before, UPnP isn't exactly in line with the whole concept of firewalling, so I would feel better just allowing my X-Box access to the service, which is done easily in the `miniupnpd.conf` we've been editing so far. Like PF, the rule set is last best match, so we only need two lines: one to allow the particular host and one to default deny the rest.

```
allow 1024-65535 172.16.0.4/24 1024-65535
deny 0-65535 0.0.0.0/0 0-65535
```

The number ranges are the port ranges allowed to be mapped externally and from said ranges internally. Nice fine grained control, which is helpful if you run an IPS/IDS. (One less false positive is like 5 minutes more to my day.)

There are number options worth looking at; so take a peek, but the above is the minimum.

Now you just need to start up the service (`/sbin/miniupnpd`), add that to `/etc/rc.local` to start it at boot, and you're done.

I then started up the XBox and, sure enough, my online experience was night and day. COD:MW2

reported my NAT type was Open, for which I shed a small tear, but since Availability is a key component to a security model, I'm willing to live with a devil I know instead of opening up the service wide.

UPnP has other uses (as does NAT-PMP) such as for bit-torrent clients, maybe a UPnP capable FTP client, and other such unfriendly. Though, keep in mind UPnP (especially on Windows) is a security disaster waiting to happen, as its purpose is to port map INSIDE your firewall, effectively negating the whole purpose of having one. This middle ground solution allows me to maintain my security posture, but still enjoy getting schooled by the thousands of teen-age psychopaths that plague XBox Live.

MUUG Annual Financial Statement

Once a year, as our by-laws require, we publish the group's Financial Statement.

```
Manitoba UNIX User Group
2008-2009 Season
Income Statement

Income
        Chequeing Interest          $0.07
        GIC Interest              $612.86
        Membership sales        $1,060.00

        Total income            $1,672.93

Expenses
        Bank charges               $11.04
        Coffee supplies            $32.41
        Mailbox                   $275.10
        MUUG Online               $289.32
        Newsletter                $643.15

        Total Expenses          $1,251.02

Net Income                        $421.91

Balance sheet

Assets
        Cash                    $1,352.64
        Investments (GICs)     $18,946.90
        Total                  $20,299.54

Equity
        Retained Earnings      $20,299.54
        Total                  $20,299.54
```

Shell Shortcut: Geo-locate IP Address

Want to know what country that IP address comes from? Here's a quick command line bash function (compliments of Sean Walberg and Sean Cody) to find out:

```
geo() { dig +short TXT `echo $1 | awk -F. \
'{print $4"."$3"."$2"."$1".cc.iploc.org"}'`; }
```

Just add the line to the bottom of your `.bashrc` file. Then running "`geo 130.179.16.8`" will return the ISO country code "`CA`" and "`geo 171.64.64.31`" will return "`US`".

Sending Us E-Mail?

Due to the amount of e-mail MUUG receives, we've set up an auto-reply to give you jaunty 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…?

What Do You Think?

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**.