June 2004 Vol. 16 No. 9

MUUGlines

The Manitoba UNIX User Group Newsletter

Next Meeting: May 11th – Linux in the Enterprise - Configuration and Management

Willem van Schaik from Sun Microsystems will present on Linux in the Enterprise. As Linux moves into the enterprise, configuration and management tools are needed. Sun is working on a facility called APOC (A Point of Control) that manages desktop configuration and management. This talk describes what APOC is and how it works.

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.

A Transparent Proxy Using OpenBSD 3.5

By Kevin McGregor

Maintaining a large number of Windows-based PCs can be problematic, especially when they have Internet access: They are exposed to a wide variety of malicious programs, which means you have to keep up with the patches Microsoft issues.

If just one PC has Internet access and it gets infected, it can quickly infect the rest of the Windows PCs on your network. This can be a laptop that someone took home to play with and brought back with a worm running.

In a complex network, there is the potential that at any one time, some kind of malicious program will be running somewhere, which makes you wonder: How can I do a network install and get the latest patches installed before the new PC gets infected?

I was having that very problem. I could, of course, install Windows (XP, in my case) directly from the CD followed by the patches, but the process requires a lot of babysitting, and I've got better things to do. Microsoft's RIS (Remote Installation Service) makes the network install (via PXE) very easy and fully automatic.

Knowing I could set up a firewall, I set about looking for a way to allow the PC I was installing to communicate only with a small number of trusted-to-be-uninfected servers which would be sufficient to complete the install along with the patches.

I had previously set up a couple of Red Hat systems as regular firewalls, but I wanted a small, quick, secure install. OpenBSD fits this bill. Very little time with Google (keywords: OpenBSD transparent bridge) produced some helpful documents. The remainder of this article describes how I set up my OpenBSD transparent filtering bridge.

The key to all of this is that I wanted not a regular firewall, but something that would connect a PC to a network as though there was nothing in between the two of them and yet filter the traffic. With a regular firewall, I would have spent a fair bit of time figuring out how to proxy DHCP traffic, for starters!

Anyway, on with the set up. The first order of business is to get the OpenBSD 3.5 boot floppy from http://sunsite.ualberta.ca/pub/OpenBSD/3.5/i386/floppy35.fs. Write this image to a diskette and boot it. Wait for this prompt to appear:

```
erase ^?, werase ^W, kill ^U, intr ^C,
status ^T
(I)nstall, (U)pgrade or (S)hell?
```

Press 'i' and Return. You'll see the initial banner like below:

```
Welcome to the OpenBSD/i386 3.5 install
program.

This program will help you install OpenBSD
in a simple and rational way. At any prompt
except password promts you can run a shell
command by typing '!foo', or escape to a
shell by typing '!'. Default answers are
shown in []'s and are selected by pressing
RETURN. At any time you can exit this
program by pressing Control-C and then
RETURN, but quitting during an install can
leave your system in an inconsistent state.
```

Press Return at the next two prompts to get the install going:

```
Terminal type? [vt220]
Do you wish to select a keyboard encoding
table? [no]
```

At this point you are warned to make a backup. This article assumes you've made one, or are using a disk with no data on it which you wish to keep. You are then asked:

```
Proceed with install? [no]
```

Type 'yes' and press Return. The installer then wants you to select the root disk, meaning the one which will contain the 'root' (or '/') filesystem. For an ordinary IDE system, this is typically referred to as 'wd0':

```
Available disks are: wd0.
Which one is the root disk? (or 'done')
[wd0]
Do you want to use *all* of wd0 for
OpenBSD? [no]
```

For this simple system set up, press Return, type 'yes' and press Return. Now comes the obscure part, for those of you not familiar with BSD-type systems. Instead of using the PC partition tables, the BSDs use what they call a 'disklabel', which functions much like a partition table. To make this install easy, we'll set partition 'a' to use the whole disk less a small bit for swap, and partition 'b' will contain the swap partition. As the disk I'm using here is 512 MB in size, I'll make the root partition 448 MB and use the rest for swap:

```
Initial label editor (enter '?' for help at
any prompt)
>
```

Type 'a a' and press Return; then press Return for the 'offset' prompt; then enter '448m' to specify a size of 448 MB; press Return to accept the default 'FS type', and type '/' and press Return to set this partition to mount as the root partition.

To create the swap partition, type 'a b' (partition b is assumed to be the swap partition) and press Return; then press Return three more times to accept the default offset, size and FS type. To check the results, type 'p' and press Return, which will list the current partition table. To quit and save your changes, type 'q' and press Return. Press Return again to write the new label. Since we're really sure about this, type 'yes' and press Return, and the installer will finally partition the disk. Once that's done, we can start configuring the system, beginning with the system's name:

```
System hostname? (short form, e.g. 'foo')
```

First enter the system hostname you've chosen, and press Return. After we complete the system setup, we'll reconfigure the two network interfaces, but for now we need one of them to download and install OpenBSD, so we'll set it up to use DHCP; the prompts look like so:

```
Configure the network? [yes]
Available interfaces are: le1 le2.
Which one do you wish to initialize? (or
'done') [le1]
Symbolic (host) name for le1? [padfw]
IP address for le1? (or 'dhcp')
```

Your network cards may be called something else (like fxp0). Above I've pressed Return three times to accept the defaults, after which I typed 'dhcp' and pressed Return to set up the IP information automatically. A pile of detail is printed out (which might help with fixing any problem you have at this point), followed by a prompt regarding the second interface. We'll leave this alone for now by typing 'done' and pressing Return at this prompt.

The DNS domain name is next presented here (it was provided by the DHCP server in my case), and you can press Return to accept the default. This is followed by a prompt to set the DNS nameserver, which once again will be whatever the DHCP server provided. You can change it here if you wish. Press Return to continue.

Assuming you want to use the nameserver now (a good bet), press Return. The default route is the next choice, and with DHCP the default is 'dhcp' (instead of an actual IP address). Press Return, or type a new address and press Return.

If you have some need to set specific entries in the initial 'hosts' file, you can change it at the next prompt:

```
Edit hosts with ed? [no]
```

I'd recommend against it. You can add them after the system is installed. Press Return here, and again to skip any manual network configuration. We are then prompted for the root account's password. Enter one and press Return. Retype it and press Return, for verification.

Now for the easy part, selecting the 'install sets'. Since we only downloaded the one-floppy installation disk, we must download the rest of OpenBSD during the install. As the prompt states:

```
Sets can be located on a (m)ounted
filesystem; a (c)drom, (d)isk or (t)ape
device; or a (f)tp, (n)fs or (h)ttp server.
Where are the install sets? (or 'done')
```

Press 'h' and Return to get the install sets via http. If you need to go through a proxy, you could type it here, but I don't, so press Return at the HTTP/FTP proxy URL prompt. The installer will then ask if you want to display the list of known http servers. Press Return to do so. Press space to skip through the list, looking for an appropriate server. Make a note of the number to the left of the site you wish to use. In my case, it was number 8, sunsite.ualberta.ca/pub/OpenBSD in Edmonton. At the server prompt, type '8' (or the one you want) and press Return.

Press Return again to accept the server choice, and again to use the default directory. You'll see a list of the available sets. By default, essentially everything is selected except for any of the X Window System stuff. Type 'done' at the prompt and press Return, and one more time to start the download and install.

Once all of the required files are downloaded, you'll be asked again about the location of the install sets. Just type 'done' and press Return. For a transparent proxy like we're setting up, sshd is not necessary (or even useful), but you can press Return at the next prompt:

```
Do you wish sshd(8) to be started by
default? [yes]
```

This will make it a little easier to use ssh with this machine at a later date if you want to use it (and OpenBSD) for something else.

The next prompt is a bit spurious, since we haven't downloaded any X stuff:

```
Do you expect to run the X Window System?
[yes]
```

Type no and press Return. Next is the time zone question. I assume you're in my timezone, so type 'Canada/Central' and press Return.

After a bit of processing, the install is done! Do as the prompt says, and type 'halt' followed by Return, then reboot the machine, remembering to remove the floppy. On to the configuration phase.

The changes you need to make are mercifully simple:

Edit /etc/sysctl.conf -- the only non-comment line should read
`net.inet.ip.forwarding=1`

Ensure /etc/bridgename.bridge0 contains only:
add fxp0 add fxp1 up
(If it doesn't exist, create it.) Change fxp0 and fxp1 to reflect the names of your network interface cards.

Ensure both /etc/hostname.<your first network card> and /etc/hostname.<your second network card> both contain just the single word "up".

In rc.conf.local (if it doesn't exist, copy rc.conf to rc.conf.local and comment out the last line), make sure the following lines read this way (the lines need not appear together):

```
pf=YES
portmap=NO
inetd=NO
pf_rules=/etc/pf.conf
```

Last, adapt the pf.conf file below for your use. Note that "ping" packets are allowed, any bootp/dhcp packets are allowed and anything to or from the list of "safehosts" is also allowed. This isn't the tightest ruleset, but it's sufficient for my needs. Note also that one interface allows everything; you only need to pick one interface on which to do all of the filtering.

At the end of the article are some of the URLs that I came across that provide more depth. In case there are any errors in the information provided in this article, check the MUUG web pages for updates.

```
pf.conf:
#     $OpenBSD: pf.conf,v 1.21 2003/09/02
20:38:44 david Exp $
#
# See pf.conf(5) and /usr/share/pf for
syntax and examples.
# Required order: options, normalization,
queueing, translation, filtering.
# Macros and tables may be defined and used
anywhere.
# Note that translation rules are first
match while filter rules are last match.

# Macros: define common values, so they can
be referenced and changed easily.
city="fxp0"       # use your actual
external interface name e.g., dc0
protected="fxp1"  # use your actual
internal interface name e.g., dc1

# Tables: similar to macros, but more
flexible for many addresses.
table <safehosts> { 192.197.130.18,
192.197.130.19, 192.168.7.5, 192.168.8.41,
192.168.8.246 }

# Do not filter anything based on the
internal interface.
pass in quick on $protected all
pass out quick on $protected all

# Block everything by default
block in on $city all
block out on $city all

# In rules
pass in on $city inet proto icmp all icmp-
type 8 code 0 keep state
pass in on $city proto udp from any to any
port { bootps, bootpc }
pass in on $city from <safehosts> to any
keep state

# Out rules
pass out on $city inet proto icmp all icmp-
type 8 code 0 keep state
pass out on $city proto udp from any to any
port { bootps, bootpc }
pass out on $city from any to <safehosts>
keep state
```

References

http://www.daemonnews.org/200103/ipf_bridge.html
http://ezine.daemonnews.org/200207/transpfobsd.html
http://cfm.gs.washington.edu/security/firewall/pf-bridge/
http://www.netikus.net/documents/OpenBSDTransparentFirewall/

MUUG Shirts

We have spare MUUG Golf Shirts if people are interested, we have various sizes available for $40 at the regular MUUG meetings. We have one medium blue and two XL beige, get them while they last.

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