An Open Source Layer 2 Switch

GIAC (GSEC) Gold Certification

Author: Jim Wilson

Advisor: Johannes Ullrich, Ph.D.

Accepted: July 29, 2009

Abstract:

Most networks are made up of expensive proprietary hardware and software incorporating a set of unique commands. Cost constraints and lack of knowledge often impede the implementation of such networks in a small business environment. Linux, ebtables, NTOP, samhain and iptables can provide similar functionality at a faction of the cost and within a familiar environment. Our goal is to give a step-by-step explanation of how to implement this solution for a mid-size company. What results is a window onto network performance giving us the ability to identify bottlenecks, and help troubleshoot problems. The network will be more resilient to failure and malicious activity.

Introduction:

Advantages of a Linux Bridge:

Small networks tend to grow and often times the growth is unplanned. The result is a network of daisy-chained switches, not the most reliable solution for a multi switch environment. What is needed is a solution which integrates all switches into a single collision domain or IP space. Most administrators would look at a Cisco solution at this point, but maybe we can use a Linux box instead. The Linux bridging software allows us to create a single LAN segment and combined with other Open Source software provide management and monitoring capabilities.

Why consider this approach? First, Linux is second nature to most Unix administrators . Cisco IOS is something they would seldom use and therefore require a learning period before it could be implemented successfully. Linux allows us to use proven Open Source network monitoring software and we have no licensing issues to deal and no waiting for support to help resolve install issues and bug fixes. The Internet provides valuable information on solutions to issues and strategies for implementation. Of course, the onus is on the administrators to search out the answers needed. With well supported community projects, bug and security fixes are quickly released. An additional benefit is that a Linux based bridge integrates easily into existing patch strategy and price is never an issue; Open Source products tend to have better ROI (Golden, 2005; Navica OpenLogic, 2006; Guhlin, 2007).

A more robust and familiar solution should translate into better reliability. Open Source networking tools will help us to not only provide a secure, redundant, robust networking solution, but also allow us to easily graph network performance and protocol distribution, providing a much needed window into this area of IT infrastructure.

Hardware and Software Requirements:

Hardware and Software requirements are driven by needs, therefore an understanding of the issues we are trying to address is key. The network has expanded over the last four years to include two /24 networks: one Giga-Bit (Gbit) network to deal with an ever expanding Linux cluster and one 100 Mega-bit (Mbit) for everything else. A Linux router connects the two together. The Gbit network now includes five Cisco 2948s. All the desktops and servers have transitioned to this Gbit network and only the internal mail server and one development Sun server remain on the 100 Mbit network. As the Gbit network expanded, the Cisco switches were daisy chained together to support the growth. For the rest of the article, I will refer to the 100 Mbit network as network 1 and the Gbit network as network 2.

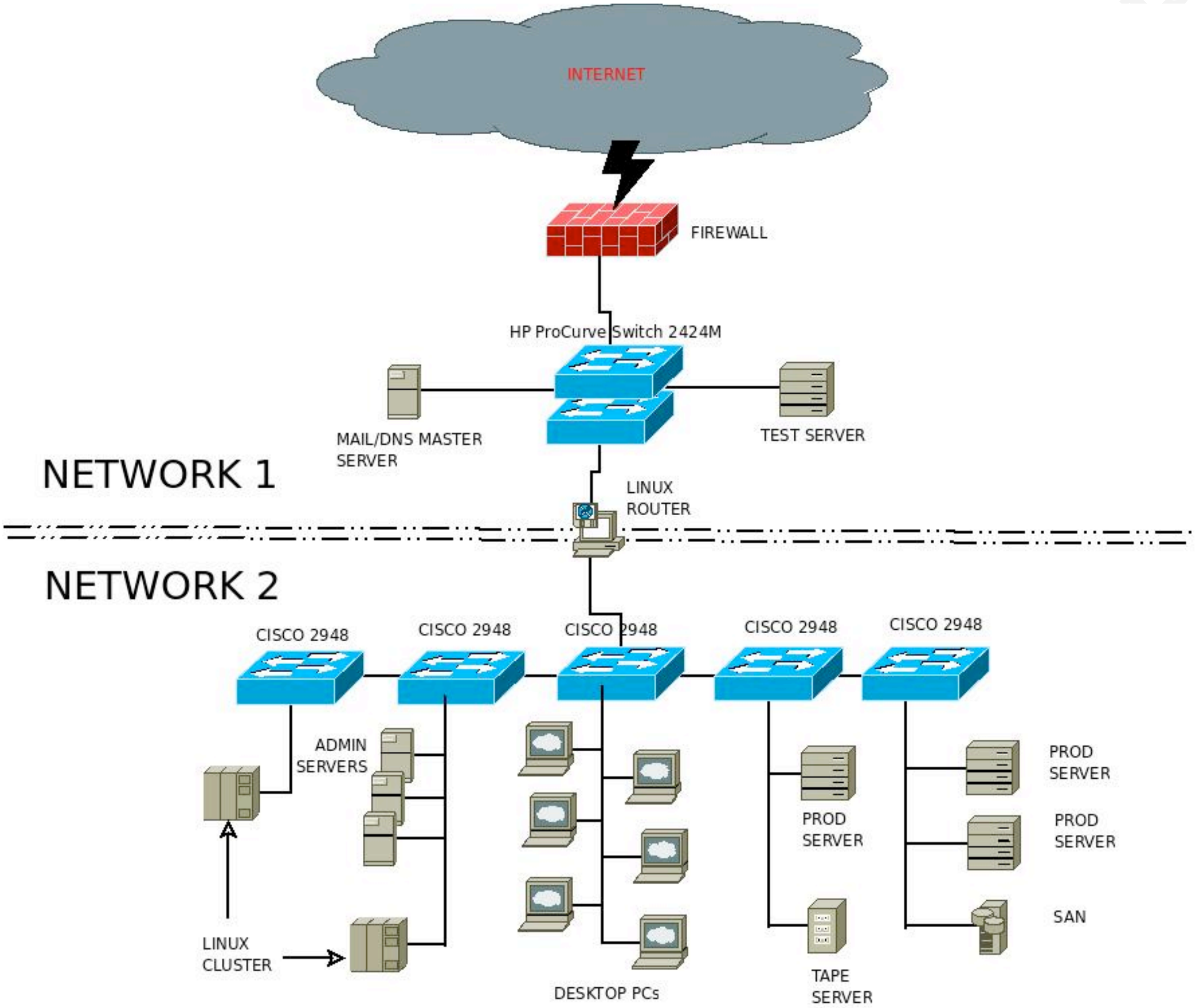


Figure 1.1 Current Internal Network

The goal is to provide a more robust network environment by attaching the Cisco 2948 switches via a layer 2 device into a single collision domain. The firewall will connect to the Linux Bridge so all traffic coming and going from the outside can be monitored and filtered if required. Network 1 will eventually become an Administration network with access tightly monitored and controlled to reduce the possibility of compromise. Only Master Administrative servers and a Test server will be allowed in Network 1 with all Slaves located in Network 2.

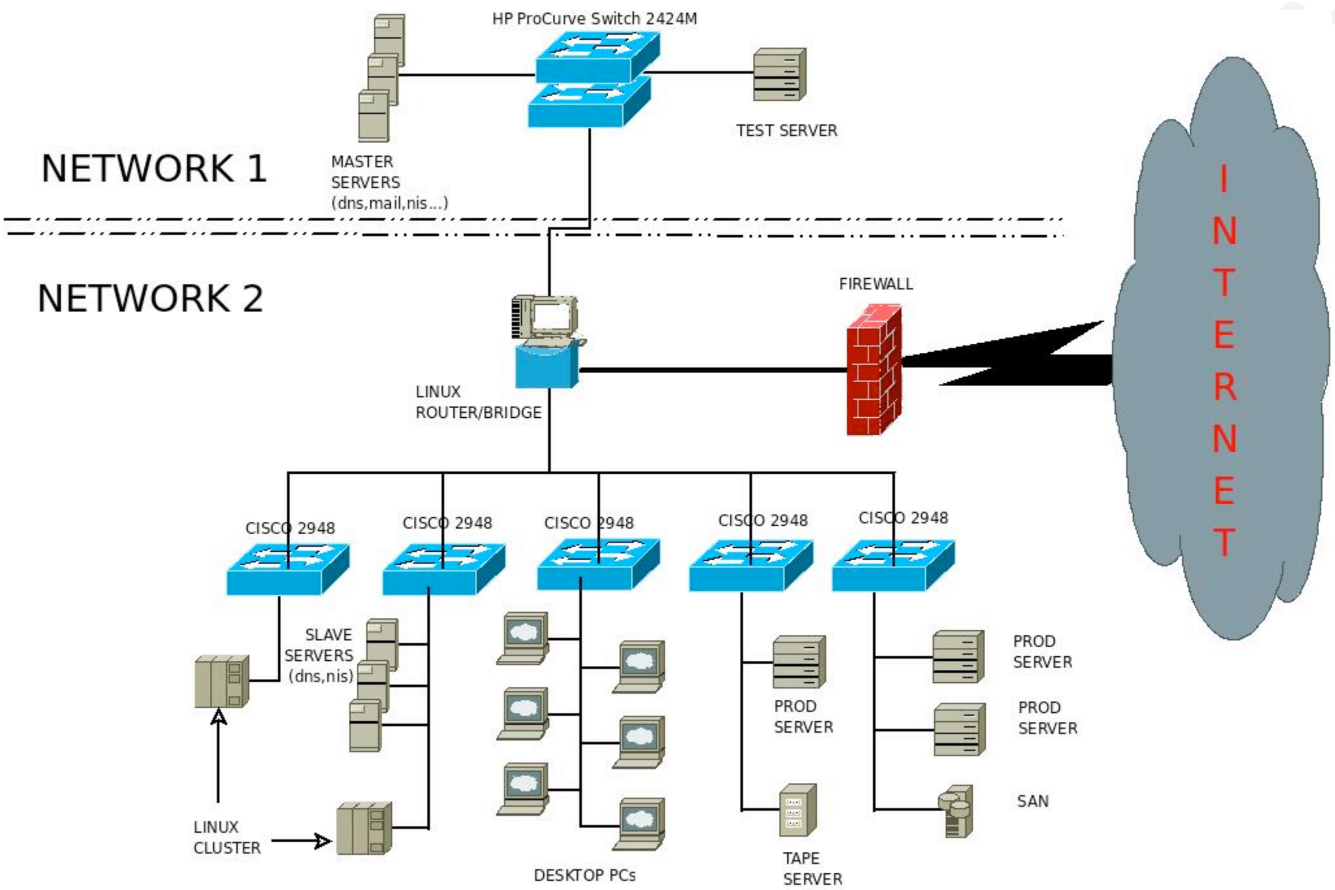


Figure 1.2 Future Internal Network

Setting up the test environment:

To simulate the Network 2 portion of the diagram (see figure 1.3), the following set of hardware will be used:

HP ML350 Server

two dual Intel 1 Gbit server cards

2 Linksys switches

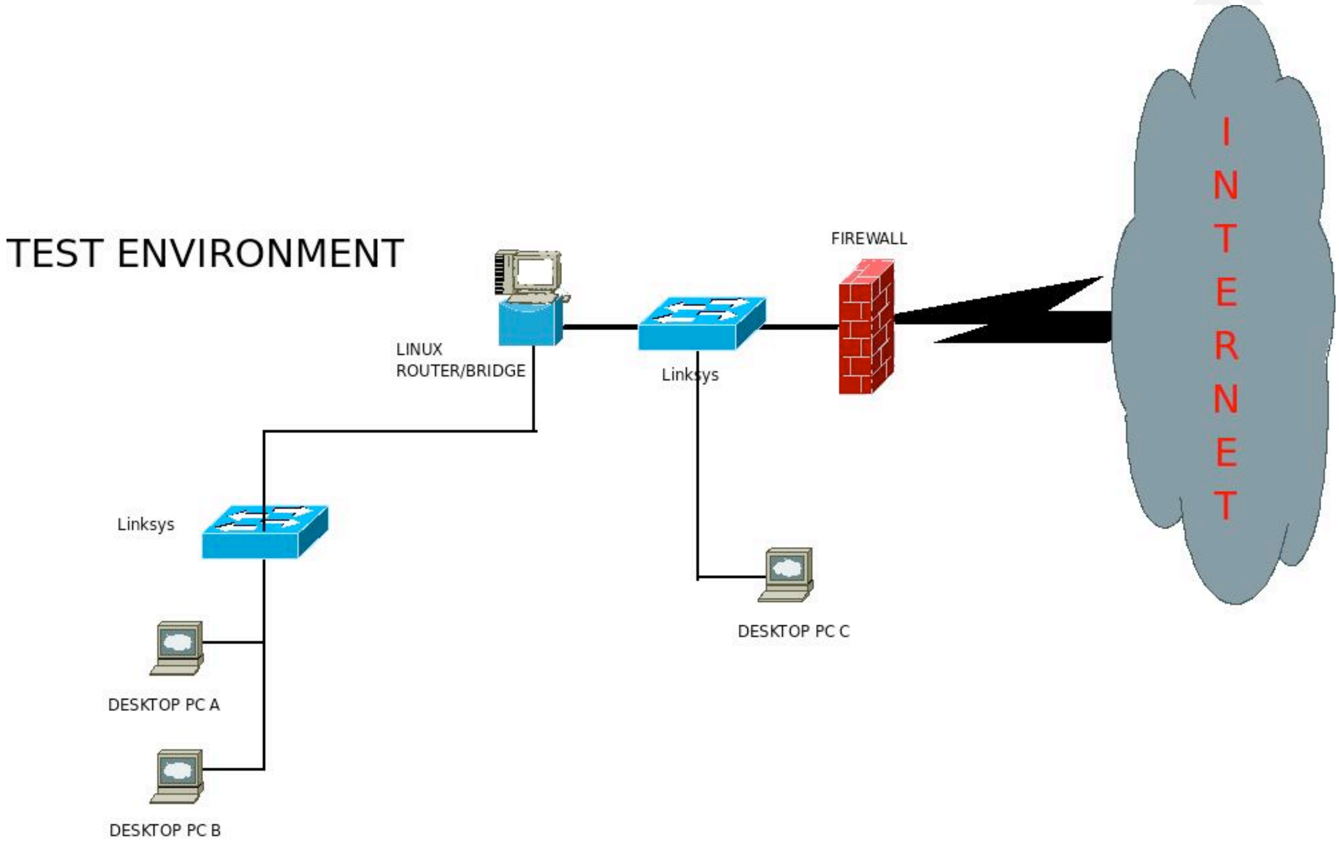


Figure 1.3 Test Environment

The performance of packets running through the bridge will be compared to those that travel across a switch freely and the ratio analyzed. This should show us if the software is up to the task. The whole process will also give us an idea of the maintainability of this solution.

To ensure maximum performance, we need to determine if the server has the necessary horsepower. CPU power and memory need to be benchmarked against the type of workload, research comparing different Unix systems in this type of a role indicates that our selection of server hardware should be up to the task (Adamo Tablò, 2005). To best position the PCI cards for maximum throughput, you need to understand the PCI bus architecture on the server. Network traffic will be traversing these buses on its trip through the bridge. You don't want to put both your cards on the same PCI bus if you can avoid it and you certainly wouldn't use the PCI 33 bus when you have a PCI-X 133 or PCI-Express at your disposal. Spend a little time with the server hardware manual, it will help to ensure you have chosen the best path for the packets to travel. If you have a chance to buy a new server, then I would suggest you look at the design specs for AMD vs Intel. Both processors are extremely fast but design decisions have made each better for certain tasks.

Which OS to use? Here again, research (Adamo Tablò, 2005) points to Linux as the best solution. OpenBSD is an option, but it doesn't seem as will supported and Linux is the faster solution. One of our major requirements is to test the feasibility of using virtualization technology. We want to see if it can provide quick recovery from operating system failure or compromise. Both VMware and Xen are well supported and with their ability to clone guest domains, both virtualization products provide fast recovery. Studies are inconclusive regarding performance comparisons between the two when using Linux as the guest operating system (VMware, Inc, 2007; ideas International, 2007; Virtuatopia, 2008; Xen.org, 2008). Xen stays consistent with our desire to use only Open Source software.

The following tables outlines the tools used:

Role	Software	Justification
Operating System	Centos 5.2	Is my standard OS For server platforms
Virtualization	Xen	Open Source, comes standard with Centos 5.2
Bridging Software	Ebtables	Only option provided with Linux
Firewall	Iptables, fwbuilder (GUI)	Standard with Linux
Network Monitoring*	NTOP	Provide bandwidth usage and traffic statistics by protocol
Host Integrity Monitoring	Samhain	Proven Open Source HIDS/File Integrity checker

NOTE:

* NTOP is a great tool to for network traffic visualization and percentage by type of traffic that is going through your network. cacti is better at tracking bandwidth utilization over time.

SNORT will be used for network intrusion detection but set up on a different server with a sensor installed on this server.

Network Setup:

Dom0 and v12 guest installs:

The HP server ships with an integrated Smart Array 641 Controller and six 36 GB, 15,000 rpm SCSI disks. Create a raid 5 configuration with a hot-spare. The OS is next. The first thing you will need is a copy of the ISO for Centos 5.2. Go to www.centos.org, find a mirror that works for you and download CentOS-5.2-xX-bin-DVD.iso (32-bit or 64-bit depending on your server architecture). It is important to get the DVD and not the CD images as it works best for the Xen guest install. The Centos site provides good documentation on how to install the OS so I will only add the things you need to keep in mind for this situation:

1. Use the following disk partition scheme: /boot, 150MB (to accommodate different kernels as required; /var, 4 GB, to ensure a run away process doesn't fill up the system disk; / for the rest you intend to use for Dom0 (Xen hypervisor). Make sure you don't use all the disk as you will be creating further partitions for the Xen virtual guest (v12).
2. Select Server -GUI and click to the next screen. De-select everything except Web Server as you will need this to install the v12 guest. Go to "Base System" and select System Tools and Administration Tools but de-select Dialup Networking Support. You should have the X Windows System selected and from the "Desktop Environment" section, select Gnome Desktop. Select "Virtualization" cluster and all development tool and libraries from the "Development" section; you will need these packages when you install some of the later software. Check under each heading and remove any tools you don't need (Editors (except vi, of course), Games, Office Productivity tools, Sound, etc.). Just remember the less you install the less you have to worry about in term of security holes and bugs.
3. For now, disable selinux but enable the firewall. We will be working in a controlled, isolated network environment and will reconfigure the firewall when we test the bridging software.

Once the installation is complete, logon and update the server software with the following command:

```
# yum update
```

Shutoff all unnecessary utilities using the “chkconfig –list” and” chkconfig <service> off” commands. Reboot the system to ensure only required services start. Use logwatch to monitor any changes to the system. You can find the configuration information in the /etc/logwatch/conf/logwatch.conf. One change you will want to make is to the “MailTo =” field and maybe increase the “Detail” setting to medium or 5. Logwatch to be a good addition the system integrity checks and there are lots of settings you can play with so it is definitely worth studying the configuration file.

Anyone who has had to try and make sense of iptables will appreciate fwbuilder. This GUI based tool helps you create and audit firewall rules. It maintains configuration files for several firewalls and will allow you to compile and install these rules on different machines. It relies on ssh so you will need the root account and password of the remote servers if you want to install these firewall rules remotely. To setup fwbuilder, use the following procedure:

1. Download the current rpms from the fwbuilder website:
   fwbuilder-3.X.rpm, libfwbuilder-X.rpm.
2. Save the rpms to /tmp/fwbuilder and install with the following command:
   ```
   # cd /tmp/fwbuilder
   # rpm -Uvh *.rpm
   ```
3. Under gnome, fwbuilder will setup an icon in the System > Administration section. To launch fwbuilder:
   ```
   # fwbuilder &
   ```
4. To ensure that the new firewall ruleset is maintained on reboot you will need to do one of the

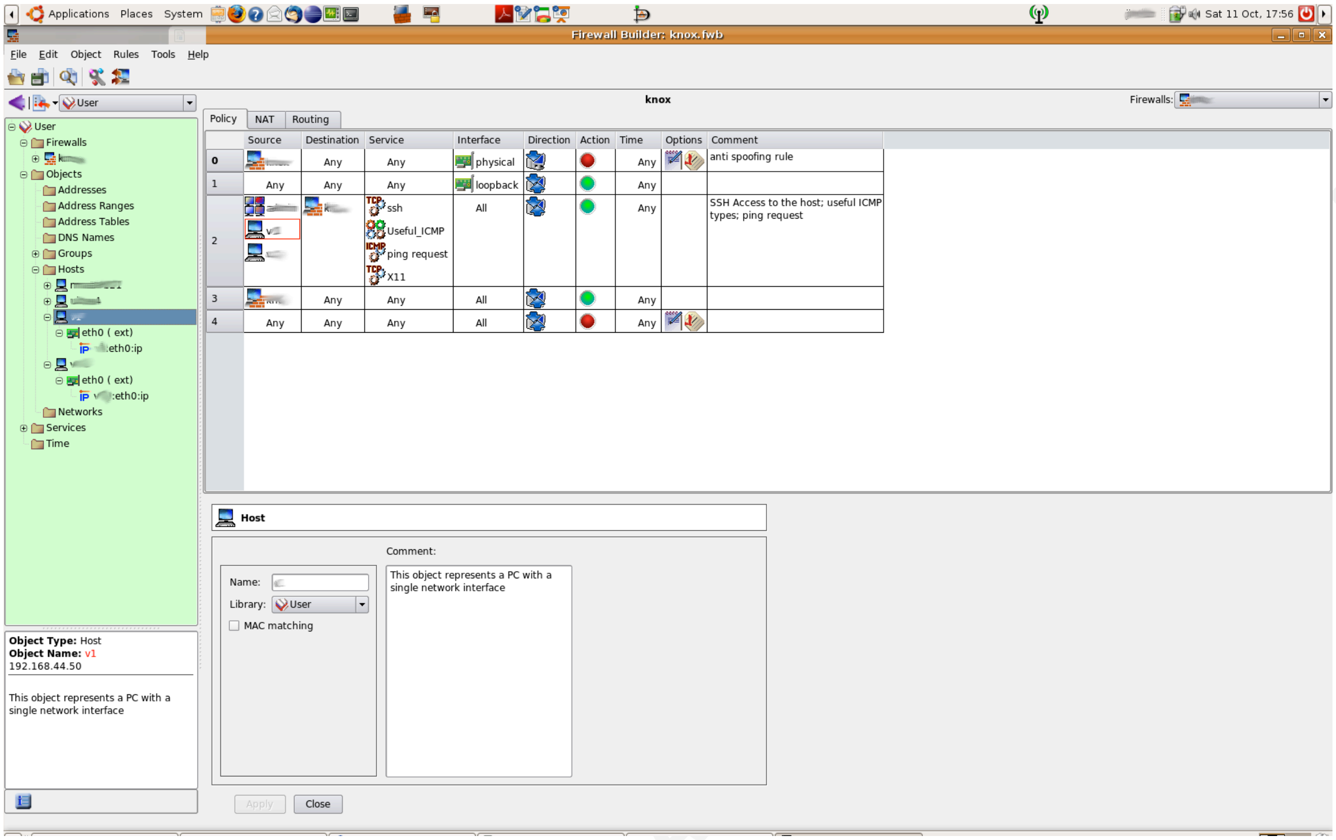
following.

1. install the new firewall from fwbuilder, then, on Redhat systems:

```
# chkconfig iptables off
# chkconfig ip6tables off
# service iptables save
```

2. or, on other systems you could run the .fw script which fwbuilder leaves in the /etc diectory from the rc.local script.

Now create two firewall rules. The first allows only ssh connections from certain hosts and stateful connections originating from the server. The second is for testing the bridge. It allows all traffic to flow but restricts logins through ssh only from specified admin servers. We will use these rules for both the Dom0 and the v12 guest server. Once the rules are created you need to first compile and then install them. All this can be accomplished from the GUI.



	Source	Destination	Service	Interface	Direction	Action	Time	Options	Comment
0	[redacted]	Any	Any	physical		🔴	Any		anti spoofing rule
1	Any	Any	Any	loopback		🟢	Any		
2	[redacted] [redacted] [redacted]	[redacted]	ssh Useful_ICMP ping request X11	All		🟢	Any		SSH Access to the host; useful ICMP types; ping request
3	[redacted]	Any	Any	All		🟢	Any		
4	Any	Any	Any	All		🔴	Any		

Name: [redacted]

Library: User

□ MAC matching

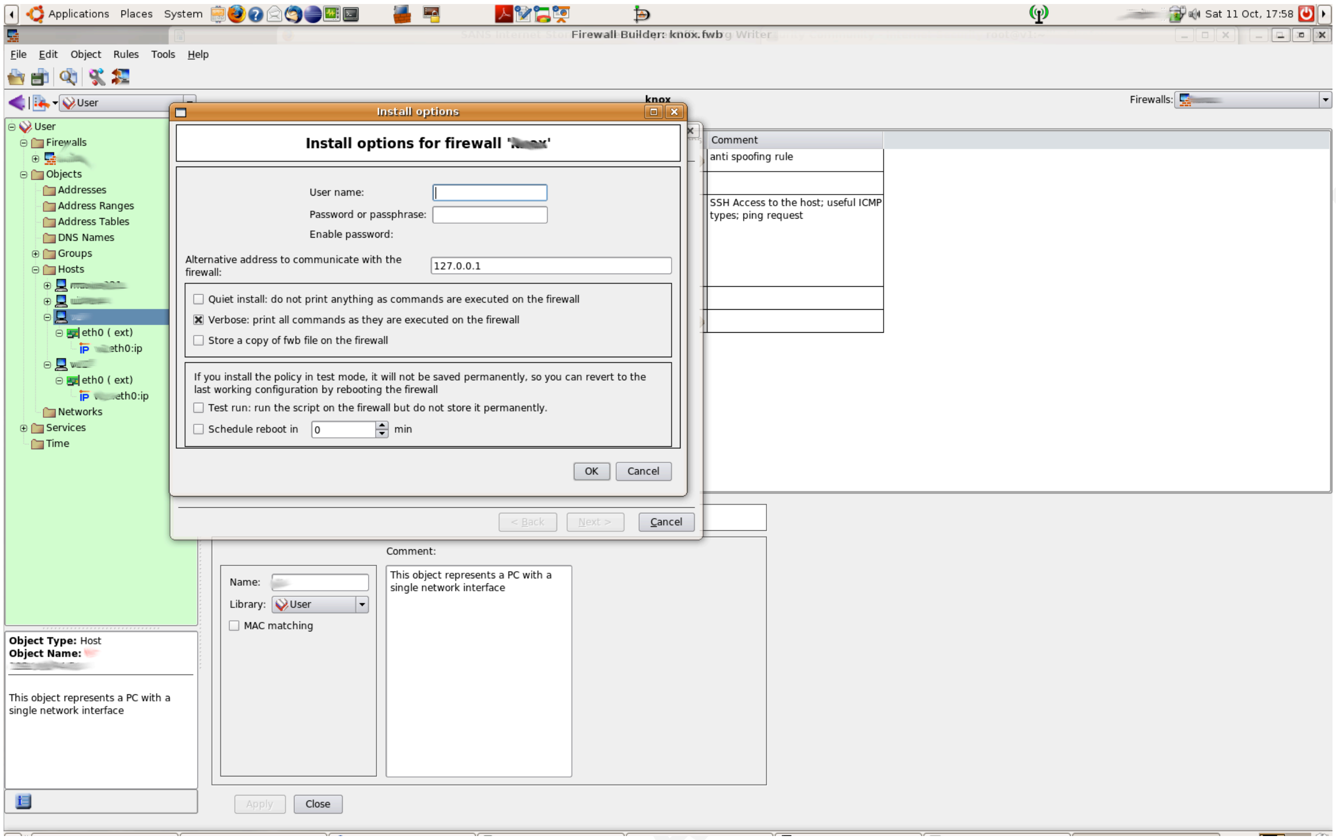
Comment:

This object represents a PC with a single network interface

Object Type: Host
Object Name: v1
192.168.44.50

This object represents a PC with a single network interface

Figure 2.0 Main fwbuilder interface



Install options for firewall '[redacted]'

User name:

Password or passphrase:

Enable password:

Alternative address to communicate with the firewall: 127.0.0.1

☐ Quiet install: do not print anything as commands are executed on the firewall

☒ Verbose: print all commands as they are executed on the firewall

☐ Store a copy of fwb file on the firewall

If you install the policy in test mode, it will not be saved permanently, so you can revert to the last working configuration by rebooting the firewall

☐ Test run: run the script on the firewall but do not store it permanently.

☐ Schedule reboot in 0 min

OK Cancel

anti spoofing rule

SSH Access to the host; useful ICMP types; ping request

Name:

Library: User

☐ MAC matching

Comment: This object represents a PC with a single network interface

Object Type: Host
Object Name: [redacted]

This object represents a PC with a single network interface

Figure 2.1 fwbulder install screen

The last thing we need to do for Dom0 is prepare for the v12 guest install. We need to setup apache so it will share out the directory containing the Centos 5.2 dvd ISO we downloaded previously, we also need to make sure we have a firewall rule in place to only allow access from new server's IP. Here are the steps to follow:

1. create the directory to hold the .ISO files

   ```
   # mkdir -p /var/www/html/isos/centos52
   ```

2. copy the .ISO to the .../ISO directory and setup the correct permissions

   ```
   # cp CentOS-5.2-xX-bin-DVD.iso to /var/www/html/isos
   # chmod 755  CentOS-5.2-xX-bin-DVD.iso
   ```

3. mount the .ISO to the centos52 directory created in step 1

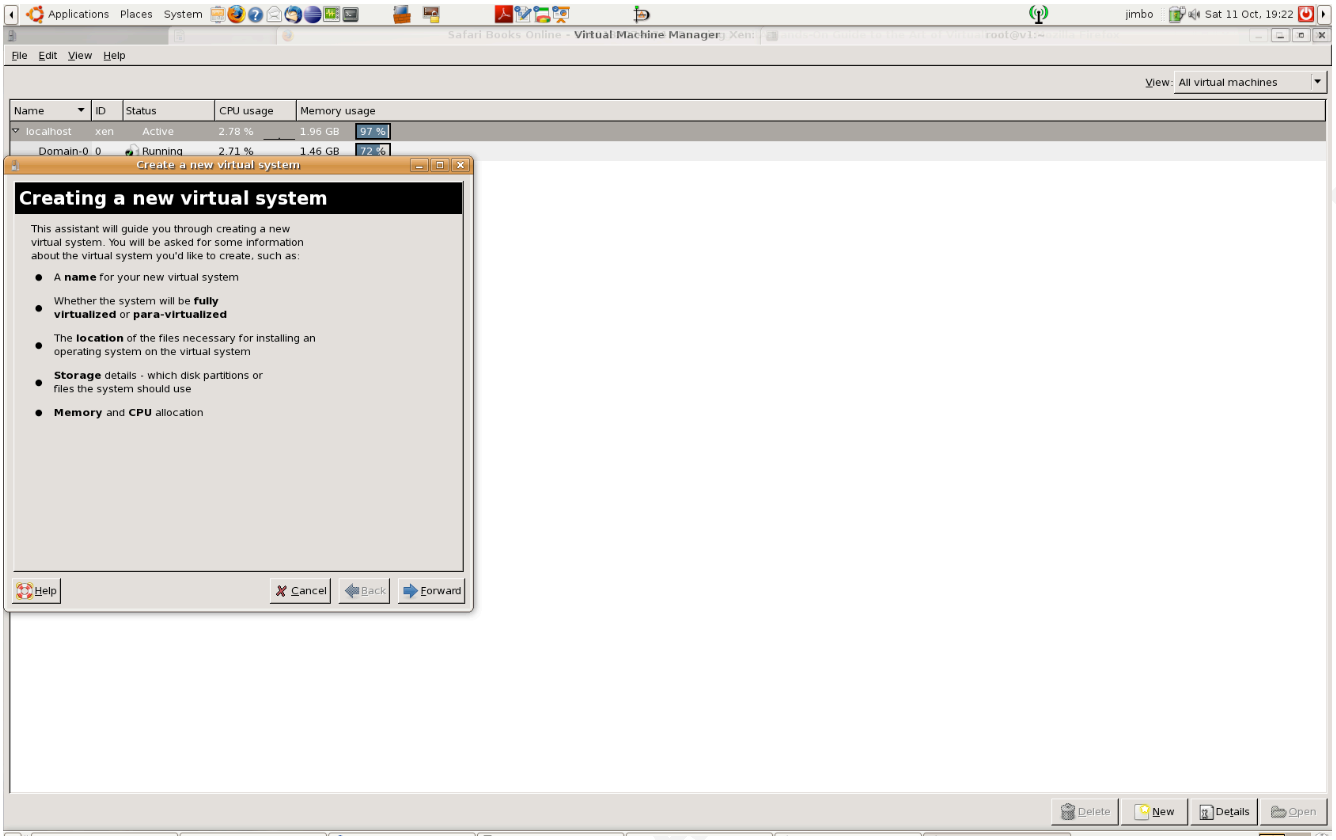
James R. Wilson

```
# mount -o loop CentOS-5.2-xX-bin-DVD.iso centos52
```

4. start the http service and test it with a browser (remember to change or turn off your firewall rules)

```
# service httpd start
```

Installing the v12 guest server OS is similar to the Dom0 procedure. First, create the partition to use as the DomU (v12) "disk." With Xen you have two options for "disks", you can use a file created and stored in an existing partition on the Dom0 server or you can create a separate partition "disk." The latter physically separates the two servers and is better for performance (Matthews Dow Deshane Hu Bongio Wilbur Johnson, 2008). You can use any method you chose to create the partitions. Once the partition is created and the apache server is serving up the .ISO, launch the Xen GUI, virt-manager, and create the virtual client. To start the process, right click on the localhost entry and select "New", the wizard will guide you through the rest..



Name	ID	Status	CPU usage	Memory usage	
localhost	xen	Active	2.78 %	1.96 GB	97 %
Domain-0	0	Running	2.71 %	1.46 GB	72 %

Creating a new virtual system

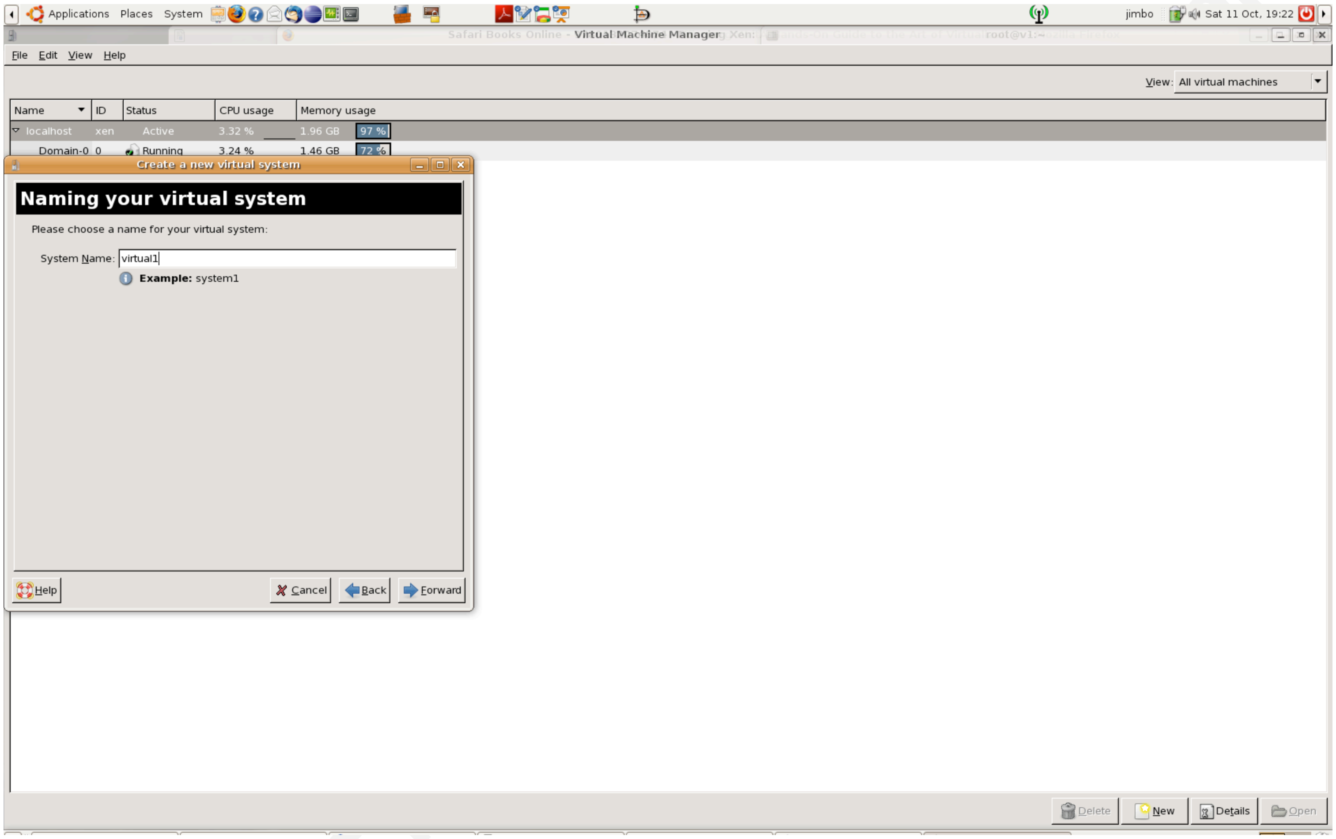
This assistant will guide you through creating a new virtual system. You will be asked for some information about the virtual system you'd like to create, such as:

- A **name** for your new virtual system
- Whether the system will be **fully virtualized** or **para-virtualized**
- The **location** of the files necessary for installing an operating system on the virtual system
- **Storage** details - which disk partitions or files the system should use
- **Memory** and **CPU** allocation

Figure 2.2. virt-manager interface

You will want to select para-virtualization which will likely be the only option available unless your CPU supports hardware virtualization. Then you need to point to the installation directory on your Dom0. For the "Normal Disk Partition", select the "disk" partition you created. Next you will specify the type of network device. You will want to select a "Shared physical device" and in my case that was xenbr0, you will have the opportunity to enter IP information for this interface later. It is important that you are prepared with the IP information (i.e. IP Address, subnet mask, gateway) as this will be required. You have to treat this v12 server as independent from Dom0 in regards to networking. Finally you will determine the Maximum amount of memory to use and the number of VCPUs. 1 GB and 1 respectively should work for us. Performance requirements may dictate changes to these settings later.

Once you have finished with the selection, virt-manager will launch a console within a separate window which will show the boot process and begin the standard Centos install process. The OS Install will be the same as for Dom0 except you will not have the option to select Virtualization. You will also not need to install fwbuilder. Please refer to the Dom0 installation instructions if you have any issues.



Virtual Machine Manager

View: All virtual machines

Name	ID	Status	CPU usage	Memory usage
localhost	xen	Active	3.32 %	1.96 GB 97 %
Domain-0	0	Running	3.24 %	1.46 GB 72 %

Create a new virtual system

Naming your virtual system

Please choose a name for your virtual system:

System Name: virtual1

Example: system1

Figure 2.3 adding a new virtual client

Once the installation procedure is completed and the v12 server is up to date, install the firewall you created in Dom0 with fwbuilder and make sure that it will startup on boot. One last important step is to ensure v12 will boot when Dom0 boots. To accomplish this do the following on Dom0:

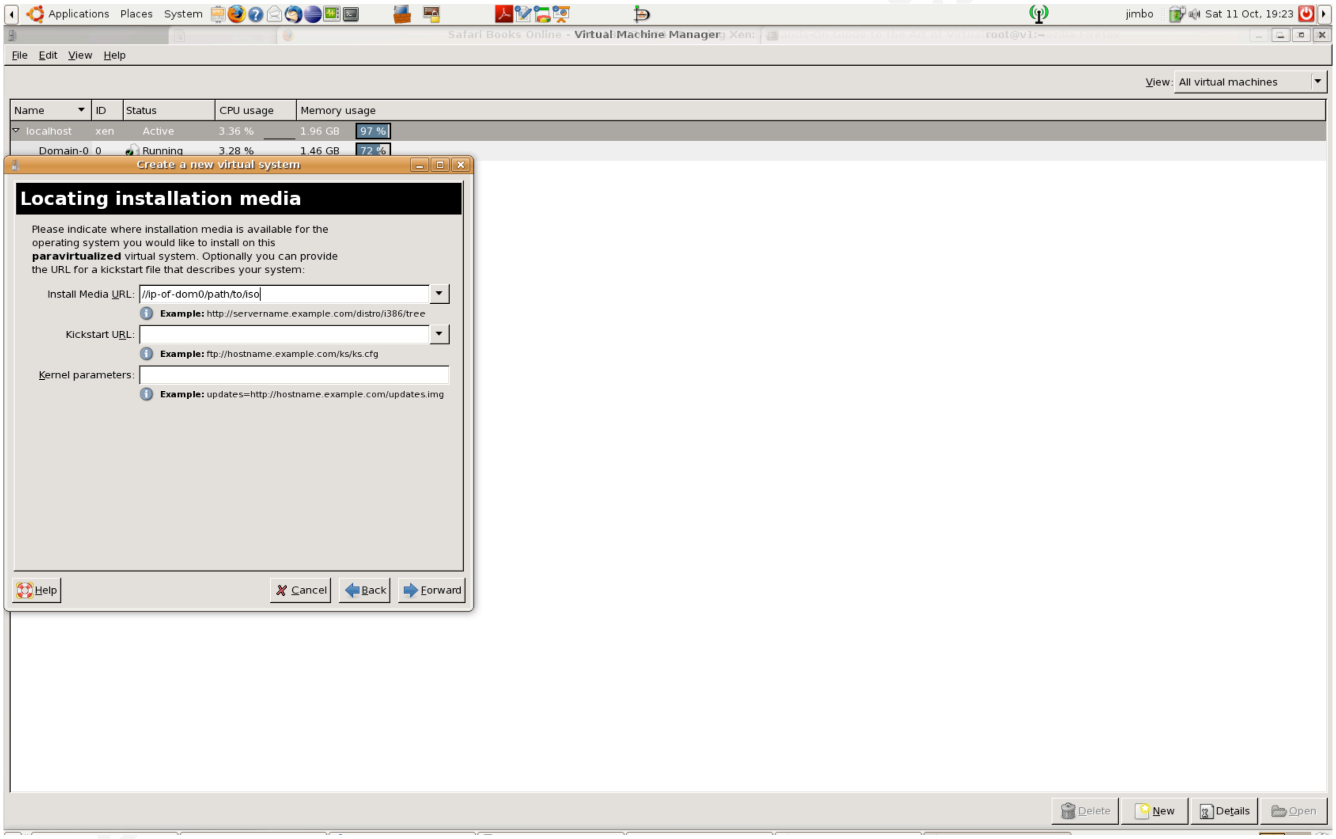
James R. Wilson

1. go to the Xen directory

   ```
   # cd /etc/Xen
   ```

2. create a link from your newly created v12 to the /etc/Xen/auto directory:

   ```
   # ln -s <v12_name> auto/v12_name

   # ls -la auto/

   total 16

   drwxr-xr-x 2 root root 4096 Oct  7 22:17 .

   drwx------ 4 root root 4096 Sep 24 18:48 ..

   lrwxrwxrwx 1 root root    6 Oct  7 22:17 v12 -> ../v12
   ```



View: All virtual machines

Name	ID	Status	CPU usage	Memory usage	
localhost	xen	Active	3.36 %	1.96 GB	97 %
Domain-0	0	Running	3.28 %	1.46 GB	72 %

Create a new virtual system

Locating installation media

Please indicate where installation media is available for the operating system you would like to install on this **paravirtualized** virtual system. Optionally you can provide the URL for a kickstart file that describes your system:

Install Media URL: //ip-of-dom0/path/to/iso

Example: http://servername.example.com/distro/i386/tree

Kickstart URL:

Example: ftp://hostname.example.com/ks/ks.cfg

Kernel parameters:

Example: updates=http://hostname.example.com/updates.img

Figure 2.4 selecting a location for the .ISO

Well, that completes the install of the basic OS for Dom0 and the v12 servers. We are now ready to install the bridging software.

Setting up the bridge with ebtables:

For a good discussion on Xen networking, I point you to (Xen.org, 2008). After reviewing this information, you would think that you could tweak the code and add any subsequent pethX (physical ethX Dom0 NIC) to the first bridge xenbr0, to which peth0 already belongs. It is in a sense all we are doing when we attach our two v12 server interfaces, xenbr0 and xenbrX, to the same bridge but it is a little tricker then it seems. When I tried my hypothesis, I created a black hole which brought my network to a standstill. I suspect I created a routing loop inside my virtual network which sucked in IP packets from the outside and once trapped they could not escape; notice how I have not connected this test network to my production network AT ALL. I think with enough study, you may find a way around setting up what is effectively a bridge of other bridges but I kept things simply and treated the two bridges as regular NICs.

In our test situation, we will need only two virtual network devices, xenbr0 and xenbr1. Xenbr0 is created by default but we will need to create xenbr1 and add it to v12. To start, lets tackle the addition of the new network device to Xen As per the documentation (Xen.org, 2008):

1. Change directories to the /etc/Xen/scripts directory.
2. Create script 2dev-network with the following information:

```
#!/bin/sh
        dir=$(dirname "$0")
        "$dir/network-bridge" "$@" vifnum=0
        "$dir/network-bridge" "$@" vifnum=1
```

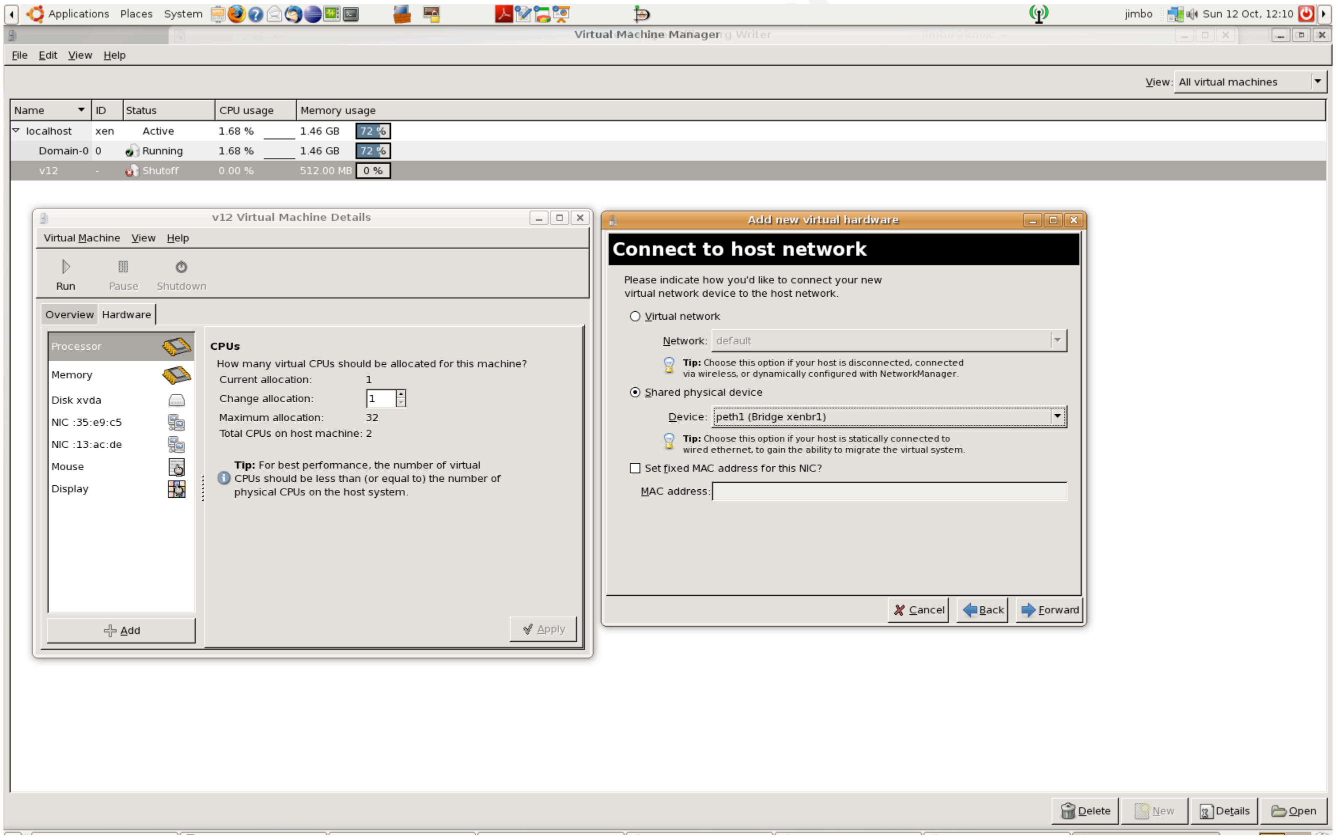
3. Make sure to make it executable:
   `# chmod 755  2dev-network`
4. Now, move up to the /etc/Xen directory and modify the /etc/Xen/xend-config.sxp as

follows. The diff command shows the changes made to the original script:

```
# diff xend-config.sxp xend-config.sxp_orig
                91c91
                < (network-script 2dev-network)
                ---
        > (network-script network-bridge)
```

5. Restart Dom0.
6. The ifconfig will now show two xenbrX devices: xenbr0 (eth0) and xenbr1 (eth1).
7. Add the new network device to v12 using virt-manager on Dom0. It is best to shutdown v12 first:

```
# virt-manager
```



Name	ID	Status	CPU usage	Memory usage	
localhost	xen	Active	1.68 %	1.46 GB	72 %
Domain-0	0	Running	1.68 %	1.46 GB	72 %
v12	-	Shutoff	0.00 %	512.00 MB	0 %

v12 Virtual Machine Details

CPUs

How many virtual CPUs should be allocated for this machine?

Current allocation: 1
Change allocation: 1
Maximum allocation: 32
Total CPUs on host machine: 2

Tip: For best performance, the number of virtual CPUs should be less than (or equal to) the number of physical CPUs on the host system.

Add new virtual hardware

Connect to host network

Please indicate how you'd like to connect your new virtual network device to the host network.

○ Virtual network

Network: default

Tip: Choose this option if your host is disconnected, connected via wireless, or dynamically configured with NetworkManager.

● Shared physical device

Device: peth1 (Bridge xenbr1)

Tip: Choose this option if your host is statically connected to wired ethernet, to gain the ability to migrate the virtual system.

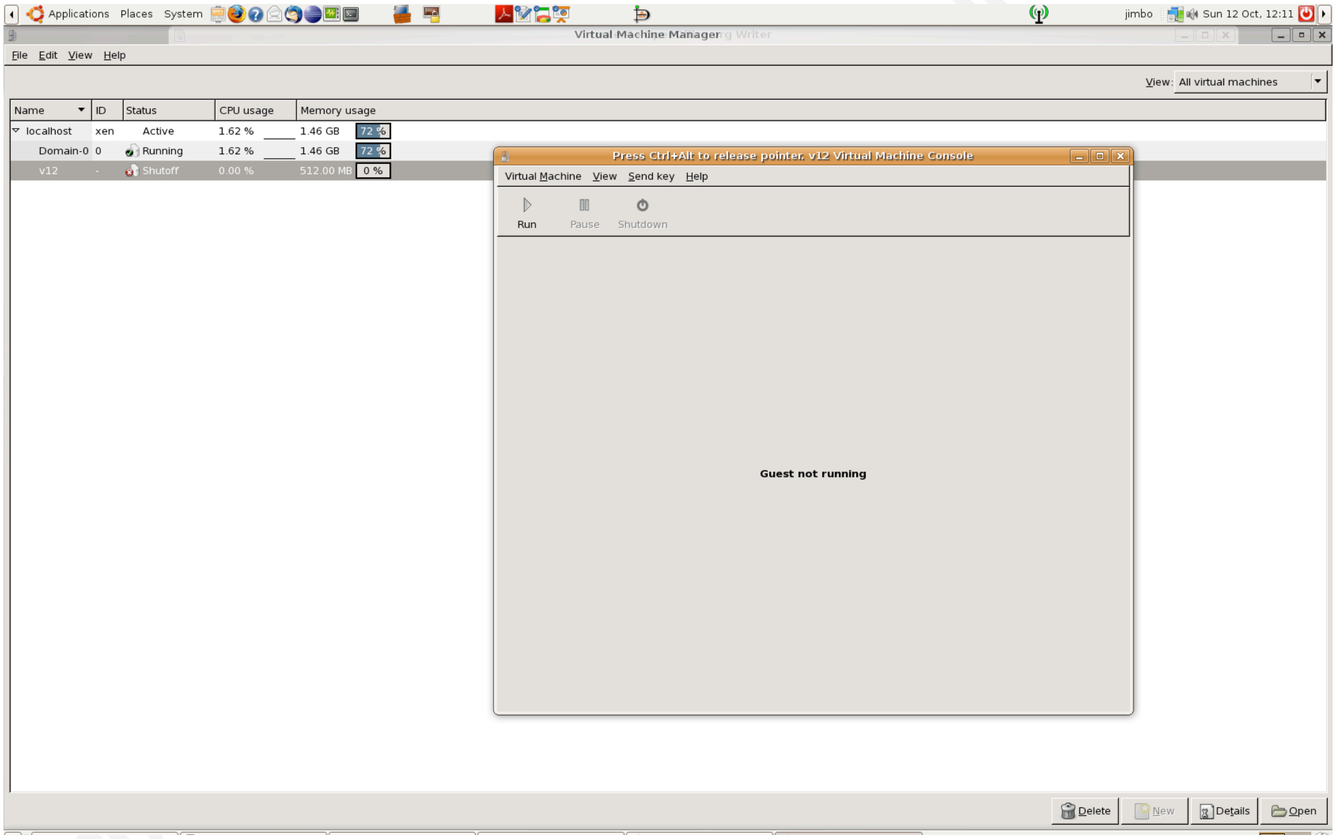
□ Set fixed MAC address for this NIC?

MAC address:

Figure 2.5 Adding the new network device to v12

James R. Wilson

8. make sure you check the /etc/sysconfig/network-scripts directory for the appropriate ifcfg-ethX entry and that it corresponds to what you see on the virt-manager, v12 “Machine Details” screen. In particular, pay attention to the MAC address entries and the ethX number sequence. It may change the order of the the devices so you may have to change the ifcfg-ethX entries accordingly
9. Reboot the Dom0 server to ensure networking starts as expected. You can watch the startup via the console on virt-manager (Figure 2.6)



Name	ID	Status	CPU usage	Memory usage	
localhost	xen	Active	1.62 %	1.46 GB	72 %
Domain-0	0	Running	1.62 %	1.46 GB	72 %
v12	-	Shutoff	0.00 %	512.00 MB	0 %

Press Ctrl+Alt to release pointer. v12 Virtual Machine Console

Guest not running

Figure 2.6 virt-manager Virtual Machine Console

10. Use ifconfig to confirm the existence of the second interface:

James R. Wilson

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:16:3E:19:C2:58
          inet addr:< IP ADDR> Bcast:XXX.XXX.XXX.255  Mask:255.255.255.0
          inet6 addr: fe80::216:3eff:fe19:c258/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12845 errors:0 dropped:0 overruns:0 frame:0
          TX packets:168 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16007907 (15.2 MiB)  TX bytes:21394 (20.8 KiB)

eth1      Link encap:Ethernet  HWaddr 00:16:3E:5A:83:98
          inet6 addr: fe80::216:3eff:fe5a:8398/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:60 (60.0 b)  TX bytes:6542 (6.3 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2906 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2906 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4280904 (4.0 MiB)  TX bytes:4280904 (4.0 MiB)
```

Now let's install the ebtables software and setup the bridge. The DAG repository has released an rpm for ebtables for Centos 5.2. You can either download the .rpm and any required dependencies and install them or add dag's repository to your /etc/yum.repos.d directory and use yum install. I have chosen to do the latter. You will need to accept his gpgkey which is a good thing.

1. Download and install rpmforge-release from http://dag.wieers.com/rpm/packages/rpmforge-release/ this will update the yum repository with the needed information to access the dag repository.

```
# rpm -Uvh rpmforge-release-0.3.6-1.el5.rrrf.i386.rpm
```

2. Update your yum configuration:

```
# yum update
```

3. Now install ebtables:

```
# yum install ebtables
```

We are ready to create the bridge. Let's call it br44:

4. Add the bridge:

```
# brctl addbr br44
```

5. Add the interfaces to the bridge

```
# brctl addif br44 eth0
# brctl addif br44 eth1
```

We need to clear the existing IPs from the interfaces (NOTE: You will need to do this via the Dom0 virt-manager Virtual Machine Console to v12 as you will lose network connectivity until the IP is assigned to the bridge device).

6. Clear the IP information for both network devices:

```
# ifconfig eth0 0.0.0.0 up
# ifconfig eth1 0.0.0.0 up
```

7. And start the bridge:

```
# ifconfig br44 up
```

8. Configure the bridge with the IP of eth0:

```
# ifconfig br44 <IP> netmask 255.255.255.0 up
```

9. and don't forget the default route!

```
# route add default gw <IP of default gw>
```

To survive a reboot, you will need to add these commands to a custom startup script or to the existing rc.local startup script. If you are creating your own startup script, make sure you place the script after the networking and firewall startup scripts.

Once we have a good picture of the type of traffic we should expect on the network, we can tweak our firewall ruleset accordingly. We will now install NTOP. This piece of software will help us to not only quantify but also qualify our network traffic.

Network Monitor Setup:

NTOP creators, Luca Deri and Stefano Suin, wanted a tool that would provide effective network monitoring. NTOP provides a snapshot of network activity and also a history of the type and amount of traffic it has seen. It can be used to identify malicious traffic and seek out compromised machines. This in turn helps to write effective firewall rules. A major plus is the use of a web server to display the information collected; as they say a picture is like a thousand words and this is especially true when it comes to viewing trends in network traffic. For a good list of things NTOP can do for you, refer to the following website: http://www.ntop.org/overview.html.

The latest stable build didn't work for me so I downloaded the latest tarball using svn:

```
# cd to the directory where you want to build NTOP.
# svn co https://svn.ntop.org/svn/ntop/trunk/ntop
```

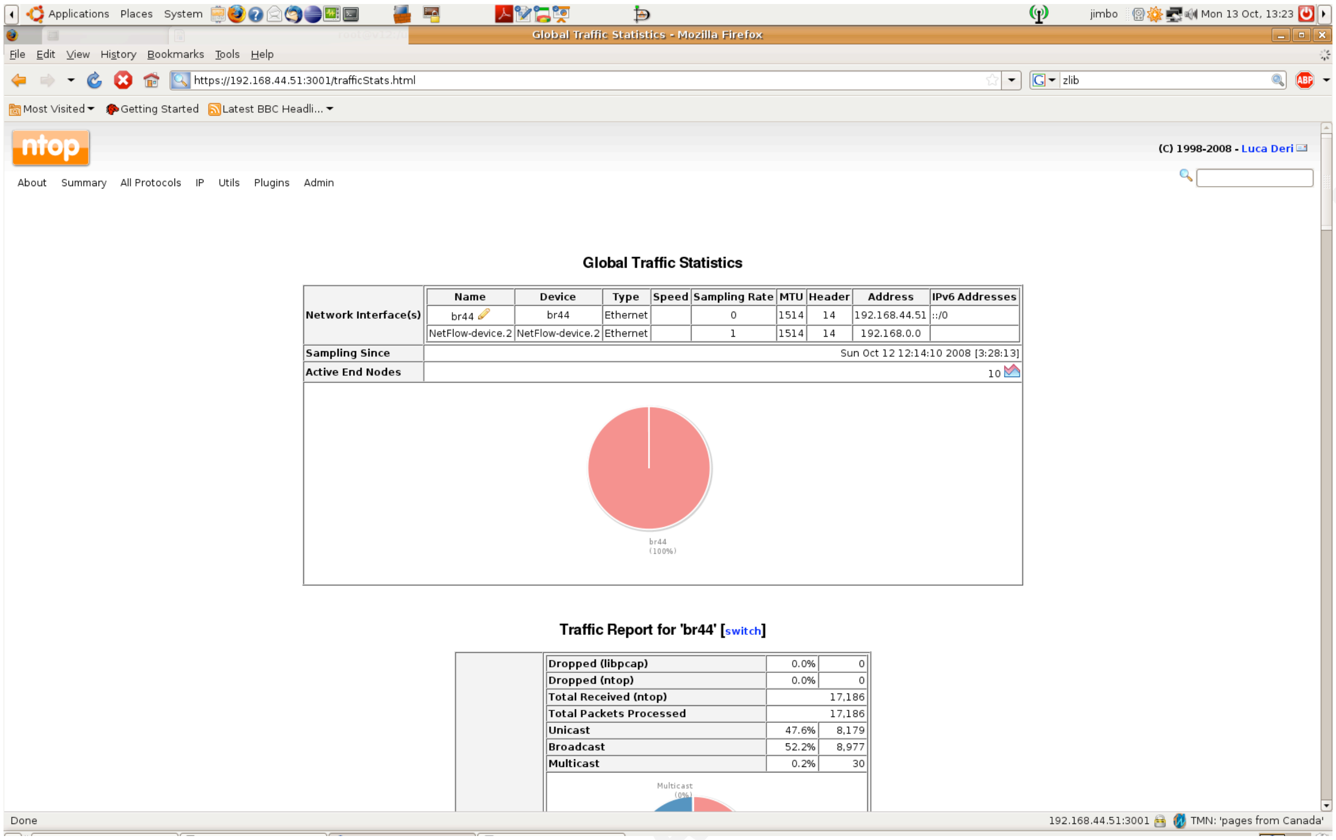
You should have a look at the INSTALL file for information on the install process. Make sure you have all the dependencies, read the doc/BUILD-NTOP.txt file for a list of the mandatory and optional packages. To support https openssl and openssl-devel are not optional. You should also install the graphviz package as NTOP uses dot to generate its graphs. Once you have untared the source, go to the src/ directory and run:

```
# ./autogen.sh
# make
# make install
# make install-data-as
```

Before starting NTOP, you should read the docs/1STRUN.txt and follow the direction listed. After it starts, use your browser and navigate to http://<IP>:3000, you should get a page similar to Figure 2.6.

Go to the Admin tab, enter your username and password and make any configuration changes relative to your environment. Here are some of the configuration changes required for our setup:

1. interface br44
2. Http-server 0
3. Https-server 3001
4. Local Subnet Address 192.168.44.0/24
5. Run as daemon yes
6. Sticky Hosts yes
7. Use syslog for logging


Applications Places System
jimbo
Mon 13 Oct, 13:23
Global Traffic Statistics - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://192.168.44.51:3001/trafficStats.html
zlib
Most Visited
Getting Started
Latest BBC Headli...
ntop
(C) 1998-2008 - Luca Deri
About Summary All Protocols IP Utils Plugins Admin
br44
(100%)
Multicast
(0%)
Done
192.168.44.51:3001
TMN: 'pages from Canada'

Global Traffic Statistics

	Name	Device	Type	Speed	Sampling Rate	MTU	Header	Address	IPv6 Addresses
Network Interface(s)	br44	br44	Ethernet		0	1514	14	192.168.44.51	::/0
	NetFlow-device.2	NetFlow-device.2	Ethernet		1	1514	14	192.168.0.0	
Sampling Since	Sun Oct 12 12:14:10 2008 [3:28:13]								
Active End Nodes	10								

Traffic Report for 'br44' [switch]

Dropped (libpcap)	0.0%	0
Dropped (ntop)	0.0%	0
Total Received (ntop)	17,186	
Total Packets Processed	17,186	
Unicast	47.6%	8,179
Broadcast	52.2%	8,977
Multicast	0.2%	30

Figure 2.6 NTOP homepage

8. /var/log/ntop; (create this directory and make sure you set the appropriate permissions to the ntop user) specifies the log directory for suspicious traffic captured
9. Set debug level to 5 (I like a lot of log information!)

You should go to the plugins and activate those you wish to use. Restart NTOP and check that everything is working according to your new configuration. Start NTOP by running:

```
# /<path to ntop>/ntop -P /<path to configuration directory>/ -u <ntopuser>
```

Put this in rc.local to have it start on system boot. NTOP is great for qualifying and quantifying network traffic. To trend bandwidth, CPU , memory, disk utilization, and other pertinent information, I use a combination of Cacti, http://www.cacti.net, and ganglia, http://ganglia.info/. Both tools have proven very effective in charting system and network performance.

File Integrity Monitoring:

Tripwire has been the favorite software product for this job but since they spun off a commercial product, their Open Source one seems to have lost some of its' developer base, so we decided to look at other options. Samhain is an impressive alternative with a strong community and quality software. The people have a good grip on what is need in a file integrity checker and have implemented some really unique options (Wichmann, 2006).

The install procedure is straightforward and should be familiar to most administrators:

1. Download the source code from samhain website
2. Download the key and verify

```
# gpg --import pgp.key
gpg: key 0F571F6C: "Rainer Wichmann <rwichmann@la-samhna.de>" not changed
gpg: Total number processed: 1
gpg:               unchanged: 1
```

3. Check the fingerprint

```
# gpg --fingerprint 0F571F6C
pub   1024D/0F571F6C 1999-10-31
      Key fingerprint = EF6C EF54 701A 0AFD B86A  F4C3 1AAD 26C8 0F57 1F6C
uid                  Rainer Wichmann <rwichmann@la-samhna.de>
uid                  Rainer Wichmann <rwichmann@hs.uni-hamburg.de>
sub   1024g/9DACAC30 1999-10-31
```

5. Verify the signature:

```
# gpg -verify samhain-2.4.6a.tar.gz.asc samhain-2.4.6
```

```
gpg: Signature made Wed 10 Sep 2008 11:42:27 AM MDT using DSA key ID
0F571F6C
gpg: Good signature from "Rainer Wichmann <rwichmann@la-samhna.de>"
gpg:                 aka "Rainer Wichmann <rwichmann@hs.uni-hamburg.de>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: EF6C EF54 701A 0AFD B86A  F4C3 1AAD 26C8 0F57 1F6C
```

6. It checks out against what is published on the website.
7. Now we can untar the source and begin the install:

   ```
   # tar zxvf samhain-2.4.6a.tar.gz
   # cd  samhain-2.4.6a
   # ./Install.sh
   ```

8. This will start the interactive install process. Once again you should read the user manual (Wichmann, 2009) as there are many setting and some features require careful study and planning. I enabled the following.

   ```
   enable-static
   enable-suidcheck
   enable-login-watch
   ```

   In future, I will look at enabling the rootkit detection and stealth modes.
9. To gpg sign the database we will need to generate a key:

   ```
   # gpg --gen-key
   ```

   Remember to remove the spaces when you put the key into the Install.sh prompt.
10. We are ready to compile the code:

    ```
    # make; make install; make install-boot
    ```

11. The /etc/samhainrc file holds all the configuration information.

We will now look at how to setup our final configurations.

Configuration:

The Firewall Rules:

If you have been running any type of traffic monitoring on your network you should have an idea of the type of traffic you need to support. There is the ubiquitous DNS, portmap, Microsoft related 137/138 UDP, DHCP, SMTP, NTP, HTTP, HTTPS, etc.., you may also have other traffic which is specific to your Intranet. NTOP should give you a good picture of what is traversing the network and is a good place to start.

To manage the firewall rules required, install fwbuilder on Dom0. This tool simplifies the configuration and installation (remotely) of many different firewalls on many different clients. We only have Dom0 and v12 and both firewalls have similar requirements. Allow remote ssh access to Dom0 from specific admin designated boxes. Ssh access to v12 will be through Dom0 exclusively. We will allow all traffic to pass through the bridged interfaces for now. Once NTOP gives us a better picture of what to expect, we will use both the ebtables and iptables to restrict the flow. Ebtables allows us to manage traffic at the layer 2 level while iptables is used to restrict traffic at layer 3.

One nice thing about this setup is the ability to isolate bad traffic quickly. Consider the following scenario: you have just received information from isc.sans.org that the following virus is spreading rapidly via TCP/UDP ports 23456. You can change your firewall ruleset on the Linux Router to isolate any infected hosts and stop all traffic with src/dst port of 23456.

Intrusion Detection:

Which files should remain static and which will likely change on a regular basis, that is the big question when it comes to file integrity. Run the checker with the default setup for a little while on an isolated network just to get a baseline of the file modification history of a server. Most default templates will check for critical configuration files such as /etc/passwd and /etc/shadow and flag them correctly. Once the default behavior has been determined, you can modify the configuration to suit your needs.

One of the challenges to implementing a successful file integrity strategy is to reduce the amount of information you need to sort through while NOT making it so loose that you let “true positives” fall through the cracks. You may also like to see that certain files are being changed as expected for example the /var/log/messages, wtmp, and other log files. This shows that the system is operating as expected.

The main file for samhain configuration is the /etc/samhainrc. This file is divided up into section which relate to pre-defined policies used to perform the required checks on the specified files/directories. A quick browse through will show that samhain is able to do many integrity checks. Here is just a short list of some of the more important checks performed:

- set User ID (SUID) / Set Group ID (SGID) audits
- login/logout activities
- mount checks
- overall file integrity checks
- monitor specific files (.login, .profile)in users' home directories

Samhain uses a database of file signatures to determine if any changes have been made to a file. An in-memory copy of this database is created on startup which is dynamically updated to reflect changes to the filesystem structure. This means that you will receive only one alarm per change. The on disk database is unaffected by these updates. It will need to be re-initialized if you want the changes to survive a reboot. You can have samhain ask you if you wish to update the database after every change by setting the ChecksumTest= (Misc section) to update and start samhain with the –interactive flag. From the documentation (Wichmann, 2009), we see that the file signatures include the following elements:

- a 192-bit cryptographic checksum computed using the TIGER hash algorithm (alternatively SHA-1 or MD5 can be used),

- the inode of the file,
- the type of the file,
- owner and group,
- access permissions,
- on Linux only: flags of the ext2 file system (see **man chattr**),
- the timestamps of the file,
- the file size,
- the number of hard links,
- minor and major device number (devices only)
- and the name of the linked file (if the file is a symbolic link).

You tell samhain to check a file/directory by adding:

```
dir=[recursion depth]/full/path/to/the/dir
file=/full/path/to/the/file
```

to the desired section. Wildcard patterns are supported. For directories you can have a maximum recursion depth of 99. There are numerous different configuration options so you should spend some time reading the documentation to make sure you understand all the formatting conventions. The samhain team has put a lot of thought into how to make there software tamper proof. In the above example, I have shown how to incorporate the use of a gpg signed configuration file, two other options of interest are the ability to detect kernel modifications and rootkits:

"**kcheck=*/path/to/System.map***"
--enable-stealth=*xor_val*

These allow you to hide the fact that you use samhain at all. It makes managing samhain a little more complicated but if you have very high security requirements, it may give you that extra bit of confidence you need. They provide a tool, **samhain_stealth**, to help with the configuration.

Zen Failover:

One of the advantages of this setup is the ability to use Xen's save and restore capabilities to quickly restore the system to a known good state.

1. If you are using LVM, create a copy of your v12 guest partition. System-config-lvm allows you to do this graphically.
2. Once the new logical volume is created you can use dd to make a block for block copy of v12's disk to v12-bkp disk. Make sure you have shutdown the v12 guest before you run this command as it is better not to have any filesystem activity while dd is running:

```
#  dd if=/dev/VolGroup00/v12 of=/dev/VolGroup00/v12-bkp bs=1024
15728640+0 records in
15728640+0 records out
16106127360 bytes (16 GB) copied, 1848.45 seconds, 8.7 MB/s
```

3. Now make a copy of the /etc/xen/v12 configuration file and rename it v12-bkp. You will need to update the “name” and the “disk” fields to reflect the new host information.
4. With the “xm create” command, create the new xen v12-bkp guest:

```
# xm create v12-bkp
```

You now have a copy of v12, named v12-bkp, valid as of the last dd. Have a look at your virt-manager display and you will see the new guest. Make sure you do not run both of the guests at the same time, they both use the same IP information and this will definitely impact your network.

Conclusions and Future Considerations:

This setup succeeds in providing a robust network infrastructure with a window on network activity. It is made up of proven Open Source products known for their reliability.

One of the greatest advantages is the ability to add new features or tools as they become available. Some tools of interest are:

- Cacti: a universal data collection and display tool. It relies on mrtg and snmp to gather statistics on just about anything you can think of.
- Snort: the NIDS tool of choice.
- NEO: a product written and maintained by administrators at MIT. Besides providing much layer 2 information, it helps determine on which switch port a host resides. This could come in handy when trying to determine the location of a rogue laptop, for example.

The test setup is running as anticipated. A rollout to limited production is planned. We did notice a 20-25% slowdown on some of our network related tests when comparing traffic which traversed the bridge to that between hosts directly attached to the same switch. We will be looking into this to determine the root cause. What this paper shows is that you can use Xen to setup a robust network infrastructure using a virtualized bridge to take advantage of the quick fail over capabilities inherent in Virtualization. And all this can be achieved by leveraging Open Source software.

References:

1. Salsburg, M. (2007, May). Xen vs. vmware - the battle of the brands. *Computer Measurement Group*, Retrieved from http://www.cmg.org/measureit/issues/mit41/m_41_1.html
2. VMware, Inc. (2007, February 01). *A Performance comparison of hypervisors*. Retrieved from http://www.vmware.com/pdf/hypervisor_performance.pdf
3. ideas International, . (2007). *X86 virtual machine platforms*. Retrieved from http://ideasint.eval.com/vm/
4. Adamo Tablò, M. M. (2005). Linux vs. openbsd a firewall performance test. *;LOGIN*, 30(6), 35-42.
5. Virtuatopia, . (2008). *Xen virtualization essentials*. Retrieved from

http://www.virtuatopia.com/index.php/Xen_Virtualization_Essentials

6. Xen.org, . (2008). *Xennetworking*. Retrieved from http://wiki.xensource.com/xenwiki/XenNetworking
7. ebtables.sourceforge.net, . (2008). *Ebtables/iptables interaction on a linux-based bridge* . Retrieved from http://ebtables.sourceforge.net/br_fw_ia/br_fw_ia.html
8. Robinson, J. (2005, May 20). Linux as an ethernet bridge. *LINUX Journal*
9. Wong, B. (2004, March 01). Host integrity monitoring: best practices for deployment. *Infocus*, Retrieved from http://www.securityfocus.com/infocus/1771
10. ideas International, . (2007). *X86 virtual machine platforms*. Retrieved from http://ideasint.eval.com/vm/eval.cgi?definition_id=1
11. Matthews Dow Deshane Hu Bongio Wilbur Johnson, J. E. T. W. J. P. B. (2008). *Running xen: a hands-on guide to the art of virtualization*. Retrieved from http://my.safaribooksonline.com/9780132074674
12. Wichmann, R. (2006, December 31). *A Comparison of several host/file integrity checkers (scanners)*. Retrieved from http://www.la-samhna.de/library/scanners.html
13. Wichmann, R. (2001-2009). *The Samhain host integrity monitoring system*. Retrieved from http://www.la-samhna.de/samhain/manual/
14. Golden, B. . (2005, June 15). The Roi of open source. *CIO* , Retrieved from http://www.cio.com/article/6959/The_ROI_of_Open_Source
15. Navica OpenLogic, . (2006, December). *Open source return on investment: achieving the financial promise of open source software*. Retrieved from https://fossbazaar.org/filemanager/active?fid=19
16. Guhlin, M. (2007). Open source and roi: open source has made significant leaps in recent years. what does it have to offer education? . *Technology & Learning*, 27