**Implementation of a Cyber Range on XenServer Hypervisor**

By Robert Skelton

Mentor: Jeffrey A. Nichols, Ph.D.

**Introduction**

A Cyber Range is a computer and network environment in which malware can be safely tested. This summer, we developed a process for creating a Cyber Range using the XenServer hypervisor [xenserver.org] with multiple virtual machines and virtual and physical networking. The Cyber Range will be used for testing nefarious programs on a private network that is isolated from the Oak Ridge National Laboratory (ORNL) network. I created step-by-step guidance for how to create a Cyber Range from scratch for the ORNL Cyber and Information Security Research (CISR) group. To do this, we used the XenServer hypervisor with custom scripts to mass create, start, shutdown, and destroy virtual machines quickly. We used Open vSwitch [openvswitch.org] for virtual networking and to place taps on the virtual network. Taps are used to capture network traffic in an unobservable way, for post-use analysis. The hardware consisted of two Microway 12-core, 64-GB server blades and a physical network switch. To totally isolate the Cyber Range from the ORNL network, two Ethernet cables must be disconnected from the switch. The XenServer machines contain the isolated Cyber Range, and the Virtual Machines on one blade are networked with the other. This Cyber Range will be useful for studying malware, computer virus research, and other experiments that require large, isolated networks for testing. Because of the simplicity, a large network can be created in a matter of minutes. This Cyber Range configuration is very scalable -- massive networks can be instantiated by adding more blades. Having a safe place for malware research and testing will be beneficial to CISR division scientists who perform research on potentially dangerous programs and applications.

**Background**

I will be starting my senior year of study at Auburn University in Auburn, Alabama, working towards a Bachelors of Science in Computer Engineering. Some topics I have studied include: electrical circuits, high-level programming, microprocessor programming, hardware description languages, and digital system design, among others. I had no experience with networks prior to this internship at ORNL, and all of my knowledge of networks has come from experimenting with (and breaking) both physical and virtual networks during the internship.

**Services Used**

I considered many different products and services to create the cyber range, and put time into trying to implement many of these different services and technologies. These will be detailed here.

The first option I considered was Minimega. Minimega is currently under development at Sandia National Laboratories, another Department of Energy (DOE) laboratory, and is used to mass deploy multiple Virtual Machines. This would have been a great option—using a DOE program and deploying many VMs in a short time—but the Application Programming Interface (API) 1.0 will not be finalized until August, which is after my internship is over. This means that the instruction set would constantly be changing, and the documentation for Minimega is not finalized yet. Therefore, I would be using an unstable, unfinished product, so I decided to look elsewhere.

Puppet, by Puppet Labs, and Juju, by Canonical, were also considered, but these programs are for mass rollouts of services and programs, not for creating and connecting lots of VMs in a mesh-style network. Puppet and Juju could easily be used if the VMs in the cyber range needed a certain application, but I was not given any specifics on programs required in the VMs I created. CISR researchers could easily implement Puppet or Juju to mass deploy certain programs to all VMs in the Cyber Range.

XenServer is what I decided to use to deploy VMs. XenServer is a standalone Linux distribution, which was installed on the two blades of Guppy, a GPU-based machine. XenServer is 100% command line, so I also used XenCenter to manage VMs. XenCenter is a Windows Graphical User Interface (GUI) for controlling XenServer. XenCenter made it easy to view the VMs I created without using an extra Remote Desktop Program—the built in console tab of XenCenter shows the desktop of the VM you are currently working on. I created custom scripts, using custom XenServer commands, to help me mass create, start, shutdown, and destroy VMs based on a template. This made managing large of amounts of similar VMs simple and fast. I would install a VM from a disk, and then set it up with XenServer Tools, a useful package that helps XenCenter manage VMs with more visual cues, such as CPU and RAM usage. Then, I would deploy my scripts and make 5, 10, 15 or more copies of that VM. Now I have multiple VMs running, and I can disconnect XenServer from the ORNL network. There is one downside -- XenCenter cannot do virtual networks through the GUI. XenServer’s command line interface supports my needed virtual networking software.

For virtual networking, the de-facto standard is Open vSwitch (OVS), with vSwitch representing virtual switch. OVS can do anything you want to do with virtual networks, and there is great documentation online. XenServer 6.2, the newest version of XenServer and the version that I installed on Guppy, supports OVS networking right out of the box. You just have to enable it with a simple command via the command line, which I will detail in the Process section of this document. The default virtual networking included in XenServer is the Linux Bridge. OVS provides many more features than the Linux Bridge, such as easier to manage VM network configurations and monitoring state spread across many physical hosts in dynamic virtualized environment. In the Cyber Range, OVS is used to put taps on the network, which can be used for monitoring the traffic of the network in an unobservable way.

**Process**

I implemented a cyber range using the following steps. First, I created a XenServer hypervisor on Guppy, by installing the XenServer Operating System on the machines guppy-00 and guppy-01. I installed XenCenter on a Windows computer that was separate from the Guppy cluster so that I could manage all the VMs created on Guppy using a GUI. To connect to Guppy, I used the iMac in my office and the CoRD program in order to remotely connect to the Windows box desktop. I used the XenCenter program’s console (terminal) in order to issue commands. The default virtual networking for XenServer was switched to Open vSwitch.

Scripts for XenServer were created using the vi text editor, and were driven by the powerful xe commands included in XenServer. xe commands are Xen commands that can accomplish a wide range of tasks. I implemented a script for creating a set of VMs based on a template (Figure 1), starting a set of VMs (Figure 2), shutting down a set of VMs (Figure 3), and deleting a set of VMs (Figure 4). Each of the VMs is a full copy, instead of a fast copy. A full copy was used because a fast copy will have the same IP address, which undesirable in a networked configuration.

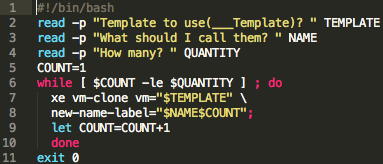


Figure 1. Bash script to create as many VMs as needed based on a template.

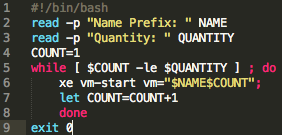


Figure 2. Bash script to start VMs.

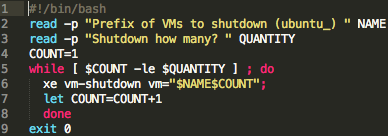


Figure 3. Bash script to shutdown selected VMs.

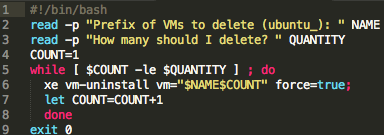


Figure 4. Bash script to delete selected VMs.

I installed five templates for researchers in CISR to use: Windows 7 Professional, Xubuntu 14.04, Honey Drive 3, and Kali Linux 1.06, and Elementary OS. Honey Drive is a modified Ubuntu that comes with copious amounts of Honey Pot software, and Kali is another modified Ubuntu that comes with many penetration-testing (hacker) applications pre-installed. To create each template, I simply installed the VM, updated to the newest software, installed XenServer Tools, and then turned the machine off. This way, my scripts can create many instances of these OS templates for the Cyber Range. In Figure 5 below, I used the scripts to create 45 copies of my original Elementary OS VM, then started all 45. This took around 15 minutes to execute.

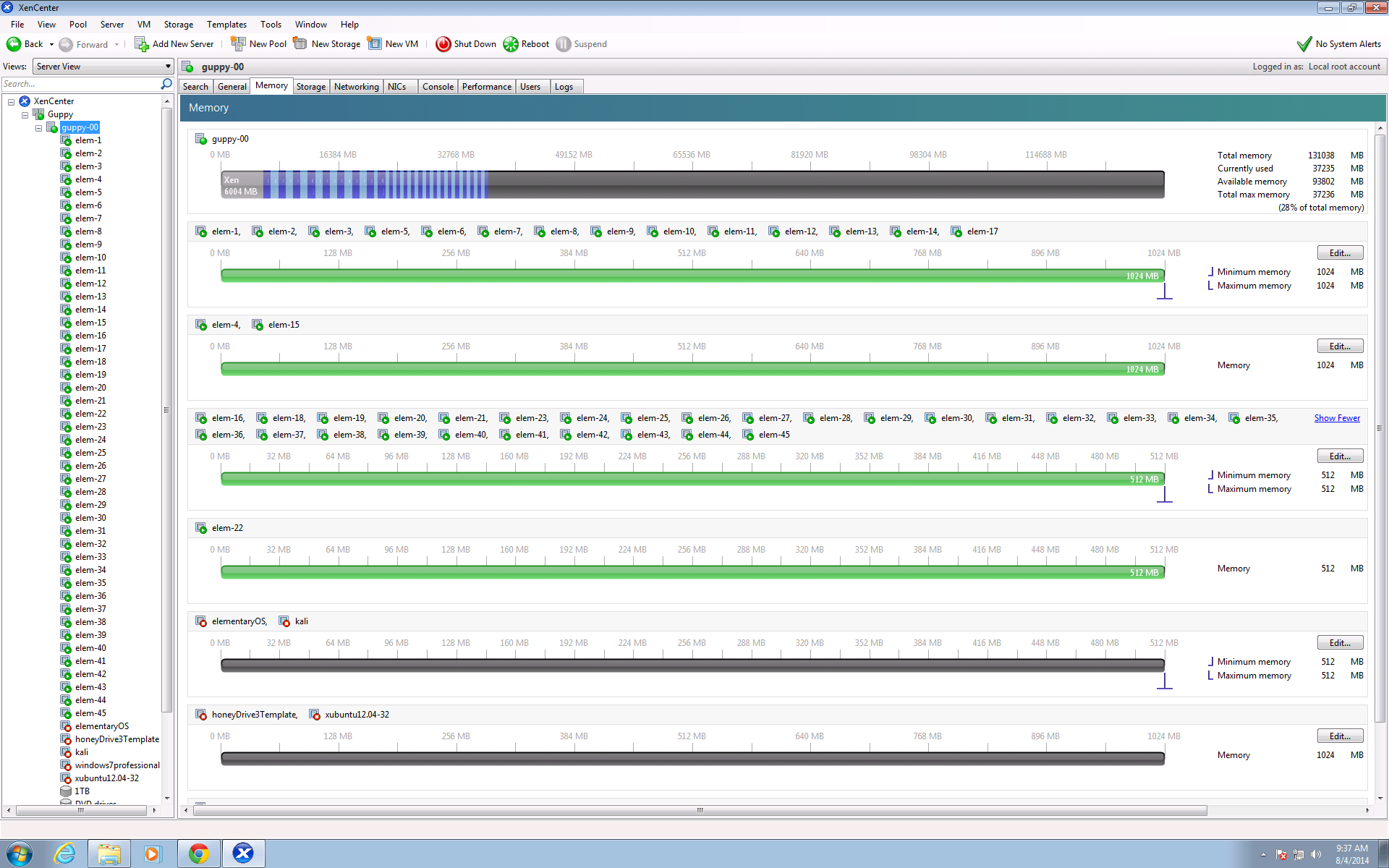


Figure 5. I created 45 instances of Elementary OS for this quick cyber range.

**Issues Encountered**

Virtual Networks involve a huge learning curve. Whenever you disconnect a device from eth0, you have lost all access to it. This happened to me multiple times throughout the project when setting up OVS, and my solution was to reinstall XenServer on that machine. Reinstalling XenServer only takes a few minutes. My solution involved using eth1 for testing the virtual network, since messing up eth0 had severe consequences. Later, I learned that eth0 is the core of keeping the Cyber Range secure, and disabling and enabling it was easier than I thought. However, to re-enable eth0, I had to manually enter “ifconfig eth0 up” on the appropriate machine.

When I installed all the VM templates, I created their virtual disk to have 50 GB of storage space. After created these disks, you can increase the size of the disk, but not decrease the size. Since these disks were quite large, and I only had 1 TB of total storage space, I ran out of space while creating new copies of VMs fast. To fix this, I reinstalled these VM templates from scratch, using the minimum specifications for that OS.

**Conclusion**

This Cyber Range will be useful for malware and virus research, and other experiments that require large networks for testing. Because of the simplicity and quickness of deployment, a large network can be created in a matter of minutes. This Cyber Range configuration is very scalable -- massive networks can be instantiated by adding more blades. Having a safe place for malware research and testing will be beneficial to the employees in the CISR division who perform research on potentially dangerous programs and applications.

**Sources**

1.<http://wiki.xen.org/wiki/Network_Throughput_and_Performance_Guide#Switching_between_Linux_Bridge_and_Open_VSwitch>

2.<http://git.openvswitch.org/cgi-bin/gitweb.cgi?p=openvswitch;a=blob_plain;f=FAQ;hb=HEAD>

3.<http://virtualfuture.info/2010/10/xenserver-fast-clone-a-vm-120-times-in-90-seconds/>

4.<http://softlit.blogspot.com/2014/05/xenserver-62-and-ubuntu-1404-lts.html>

5.<http://support.citrix.com/proddocs/topic/xencenter-61/xs-xc-vms-import-ovf.html>