Communications of the IIMA

Volume 12 | Issue 4 Article 6

2012

A Comparison of Virtual Lab Solutions for Online Cyber Security Education

Joon Son
California State University, San Bernardino

Chinedum Irrechukwu
University of Maryland University College

Patrick Fitzgibbons
University of Maryland University College

Virtual Lab for Online Cyber Security Education

Joon Son
California State University, San Bernardino
json@csusb.edu

Chinedum Irrechukwu
University of Maryland University College (UMUC)
cirrechukwu@umuc.edu

Patrick Fitzgibbons
University of Maryland University College (UMUC)
Patrick.fitzgibbons@umuc.edu

ABSTRACT

In this paper the authors describe their experience of designing a virtual lab architecture capable of providing hundreds of students with a hands on learning experience in support of an online educational setting. The authors discuss alternative approaches of designing a virtual lab and address the criteria in selecting the optimal deployment method. The authors conclude that virtualization offers a significant instructional advantage in delivering a cost effective and flexible hands on learning experience.

Keywords: virtual lab architectures, cyber security education, virtual machine, hypervisor and KVM.

INTRODUCTION

There has been a rapid expansion of using practical laboratory exercises to instruct information security courses using online technology in both academic and commercial settings. WebCT, Blackboard, and UMUC’s internally developed “WebTycho”, are just some examples of learning management systems (LMS), that have been used in support of online higher education degree programs. The primary advantages of these LMS are to facilitate student learning by incorporating a variety of online technologies including web mail, chat rooms, group collaboration and discussion boards along with serving as central repository for course content. However, when it comes to instructing technology based courses, including information security courses, online educational offerings have something in common with traditional face to face instructional methods (e.g., lectures, literature review, reading assignment, etc.) in that while being essential they are not sufficient in themselves. To supplement their online degree programs, several educational institutions have implemented hands-on labs (often called virtual labs) using virtualization technology (Burd, 2009; Fuertes et al., 2009; Li et al., 2009, 2011; Rajendran et al., 2010; Tao et al., 2010; Willems & Meinel, 2008, 2012; Yen, 2010; Zenebe & Anyiwo, 2010).

The use of hands on labs, in support of learning outcomes, is strongly supported by educational theory as a productive and effective pedagogical practice. Major theories that support the use of this technology include Bloom's Taxonomy and Gardener's theory of Multiple Intelligences. It is a commonly held belief that students learn more efficiently when engaged in higher order thinking. Hands-on lab exercises provide the means to challenge students with these higher order tasks. The use of virtual lab technology is focused in the analysis, synthesis, and evaluation areas of the taxonomy. This is evidenced by the use of the technology in the classroom. As the students are using the virtual lab, they are constantly forced to make very quick connections between what they know and what they are experiencing. In addition, the real-time environment provides an excellent opportunity for the students to make predictions regarding network intrusion and hacker behavior and to test assumptions without damaging an existing network infrastructure. This type of learning and experimenting is an essential element of an effective information security curriculum. In addition, a virtual lab infrastructure can provide a flexible and cost-effective platform that allow for running multiple operating systems and for sharing computing resources.

University of Maryland University College (UMUC) founded in 1947, has been offering online courses since 1985. As cyber attacks are being waged all over the world the demand for cyber security professionals has never been greater, UMUC began offering its graduate level online cyber security degree program in Fall 2010 that included launching a computing laboratory based on virtualization technology. The virtual lab requirements included the following objectives:

> R1. Accessible, secure and seamless access must be provided to the remote virtual lab. This means students will not have to reserve a time to use a virtual resource and that online lab service must be available around the clock, 365 days a year.
>
> R2. The remote virtual server must reliably serve a significant number of concurrent users with limited resources. No significant delay should be observed with a large number of concurrent users.
>
> R3. The virtual machine (VM) must be configured with the appropriate operating system(s) and include the required security tools for each lab exercise. In order to minimize requirements for students (e.g., configuring or installing software on their own machines), a pool of virtual machines (VM)s and a cloud based network are necessary.
>
> R4. Students must have privileged access rights on the virtual machines to execute security or network tools. Note that this means students may misuse the system resources by mistake or use malicious tools on purpose. As a result, the virtual lab environment could be jeopardized or significantly slowed down.

Based upon the above requirements, the UMUC virtual lab platform was built and first deployed in Fall 2010. At the beginning, it consisted of 7 Dell Edge Servers with VMware ESXi installed as a hypervisor. A Windows 2008 management server with vCenter server was installed along with a storage area network and 2 gigabit switches. The servers were connected via gigabit layer three switches to the storage area network and the vCenter server could be used to determine on which server the virtual machines would be placed. This entire lab infrastructure was placed in its own network separate from the UMUC intranet. The UMUC cyber security graduate degree program enrolls approximately 1500 students who are geographically located in all 50 states and 20 countries. A significant number of students are involved in information security in both the private and public sector, as well as in the military. In a typical semester over half, around 850 students, are required to participate in two online virtual labs that are included as part of 5 technical courses. Each lab is scheduled to take place over the course of a week and although some attempts have been made to avoid having overlapping labs this is not always feasible because of the nature of the 12 week semester. For example, during some weeks there may be two or more different courses, each consisting of between 10 to 20 sections of 20 students, that will be accessing the virtual labs.

BACKGROUND

At the most basic level, virtualization allows multiple virtual machines (VMs) to run concurrently on a single computer. Each virtual machine shares the resources of a single computer. The different virtual machines can run different operating systems and multiple applications in isolation on the same physical machine. Deploying automated virtualization technology, coupled with cloud based access, provide the ability for applications to be dynamically availalbe to end users. Among many different types of virtualization technologies, two virtualization technologies can be deployed for virtual labs: 1) server-side virtualization for running the virtual machines on a remote server, and 2) desktop virtualization (sometimes called client virtualization or decentralized virtualization) for running virtual machines on user’s own personal computer.

Server virtualization makes it possible to deploy virtual labs which require high-end equipment and resources. Server side virtualization software creates Virtual Machines (VM) on a remote server (VM host machine). The virtual machine (VMs) is an instance of some operating system platform running on any given configuration of server hardware and managed by a virtualization manager/monitor (also known as a hypervisor). A hypervisor is virtualization software that allows several operating systems (or virtual machines) to share a single hardware host without disrupting each other. Since many different operating systems and applications can run on a single piece of hardware, cost savings and efficiency are among the primary benefits.

An operating system image, preconfigured for labs and equipped with security tools, can run as a virtual machine. Students remotely access the virtual lab environment, load a preconfigured operating system image, run it as a virtual machine, complete a lab assignment and exit the system. The most widely deployed server virtualization platform is the VMware vSphere (VMware, 2009; Wang et al., 2010). The major components of vSphere are the VMware ESX (or ESXi), vCenter server and vSphere client. VMware ESX or ESXi is a hypervisor responsible for the creation of virtual machines on a host server. The vCenter server is a service point for administrating and managing ESX (or ESXi) host servers. The vSphere client is an interface which enables user to remotely connect to the vCenter server or ESX (or ESXi) host server.

Using desktop virtualization technology, a decentralized virtual lab approach can be implemented. Students install and run a desktop virtualization software package, like VMware Workstation or Oracle VM VirtualBox, on their notebook computers or personal computers. The prebuilt images are distributed and imported to students' laptop or desktop computers. Students run the prebuilt images (virtual machines) on their machines to complete lab assignments.

INTEGRATING VIRTUAL LABS WITHIN THE ONLINE ENVIRONMENT

Virtual Lab Platform without Virtual Network Boundary

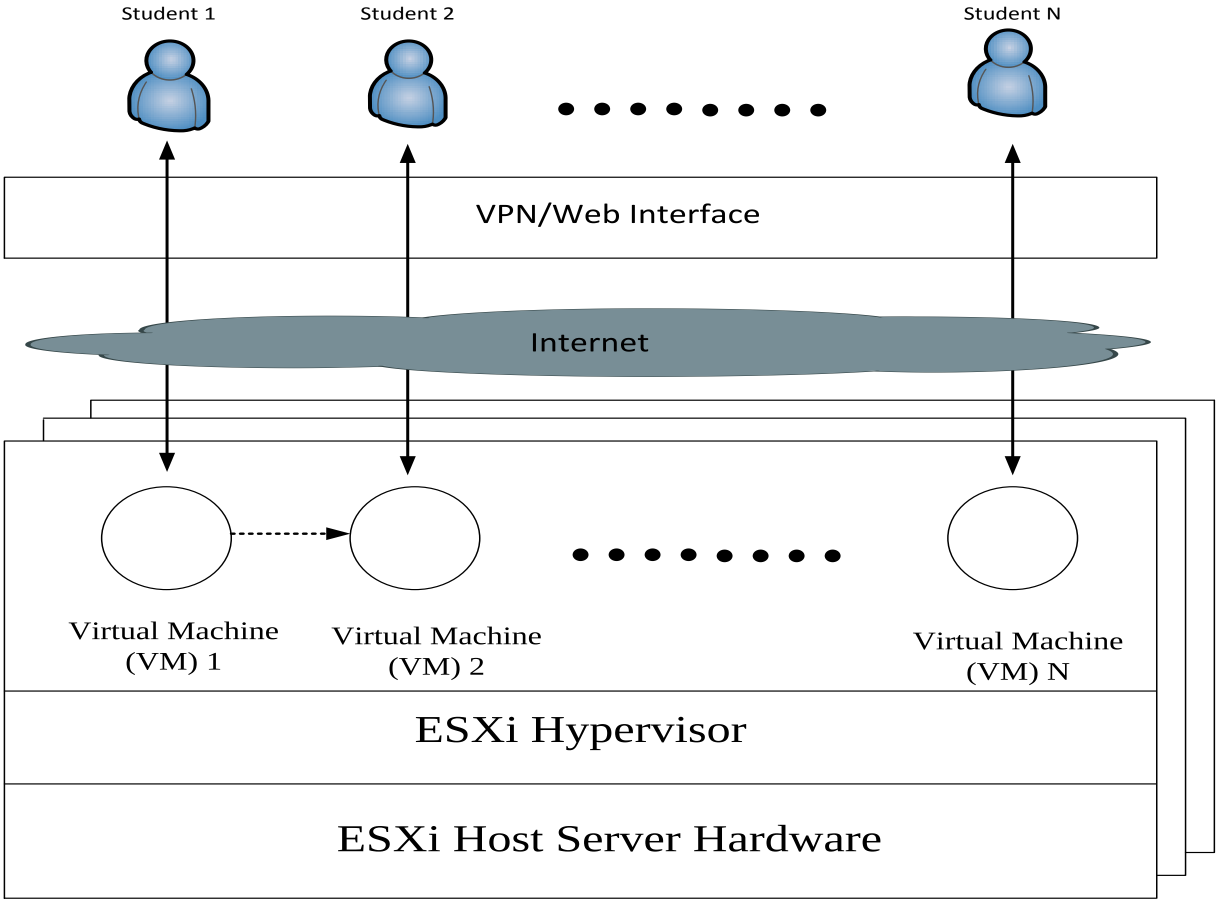


Figure 1: Virtual Lab Platform without Network Boundary.

As stated above, the initial virtual lab network was built using VMWare virtualization technology. VMWare ESXi was installed directly on bare metal Dell Edge servers. To manage these servers, vCenter software was installed on a Windows 2008 server. Virtual machines were created from vCenter which also allows the administrator to decide on which server or SAN the virtual machine would reside. This platform did not support any network segmentation. As such the virtual machines all had to belong to the same flat network and all shared the same network address. This meant that virtual machines could directly communicate with each other without additional configuration as shown in Figure 1. Each virtual machine had an IP address which users connected to using Remote Desktop client. The primary advantage of using this setup is its simplicity. There are a few disadvantages which include lack of scalability, potential high impact in the event of an internal attack. Nodes or virtual machines in a flat network are potentially affected if there is excessive network traffic resulting in congestion. This will reduce the scalability of such a network, whether it is virtual or physical. Any attack crafted by an internal malicious user can be used against other virtual machines operated by others.

Virtual Lab with Virtual Network Boundary

The UMUC cyber virtual lab was designed with the help of Dell computing. vCloud Director (vCloud Director, 2010), a virtual management service allows for several features including the creation of separate networks within the virtual lab. The virtual networks provide a separate workspace for each student as shown in Figure 2. There could include any number of virtual machines within each virtual network all dedicated to one student. In general, there are virtual machine templates with pre-configured software and tools that are spawned when a student logs on and clicks to start a lab exercise. Each virtual network with associated virtual machines loads. The virtual network and virtual machines are accessible via the student's account and are made available through vCloud director's web interface.

Some of the significant features with vCloud Director include the ability to create virtual networks, and to allow or disable communication between virtual networks. It also includes the option to make the virtual networks available or based on user account authentication. This approach is also very scalable. For example, it allows for up to 300 maximum concurrent users. Though that limit has not been tested, the UMUC virtual cyber security lab has experienced over 270 concurrent connections. The lab did not suffer from the limitations of the previous architecture because each student has their own network and is isolated from every other student. Any malicious activities or non-intended network traffic will be contained and restricted to that user's workspace and virtual network.

However, there are two main drawbacks with the current virtual network implementation. The first is sub-optimal performance and the second is lack of support for some web browsers. As stated earlier the theoretical maximum of running concurrent virtual machines is 300. Performance degradation was experienced when the number of running virtual machines approached a number much less than 300 (this also depends on the types of application running in VMs). The servers used for this deployment are high performance seven Dell PowerEdge R710 which have a maximum memory of 288GB and are popular in industry.

The other drawback is the lack of universal web browser support. As previously noted, vCloud director is a web based management interface for the VSphere virtual architecture. It can be used to create virtual machines, facilitate authentication of users, provide different access privileges based on the type of user and provide a convenient graphical tool for managing the virtual environment. vCloud Director does not support every browser nor does it support several browsers of the same version. Internet Explorer and Firefox versions are the most popular web browsers supported and yet, compatibility issues arose when students updated to newer editions of these browsers and they could no longer access the VCloud Director's web interface. This sometimes forced students to install older versions of browsers on their computers. In near future, we are going to overcome this problem by using remote communication utilities such as Remote Desktop Client and VNC which provide a graphical view of the remote virtual machine.

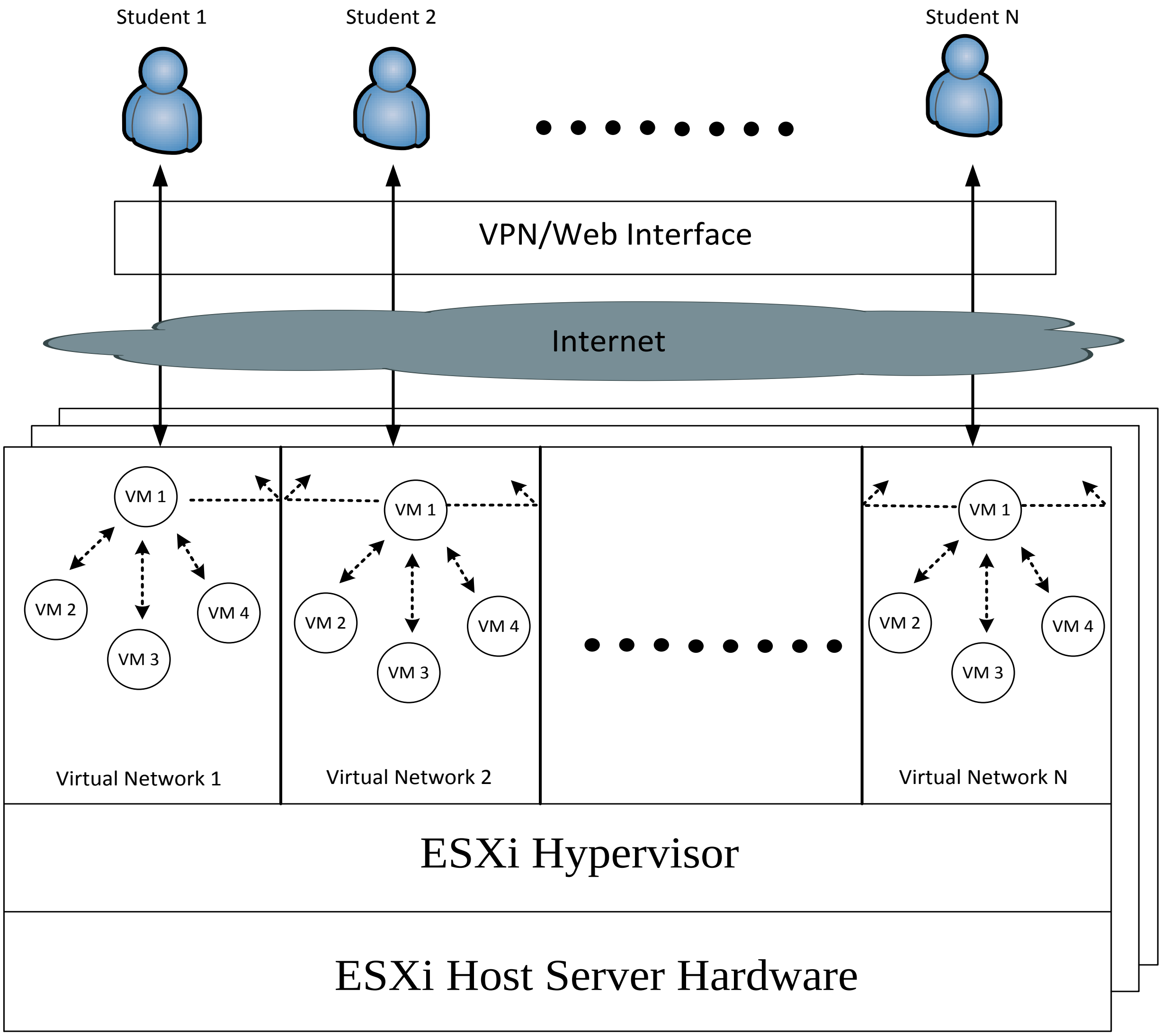
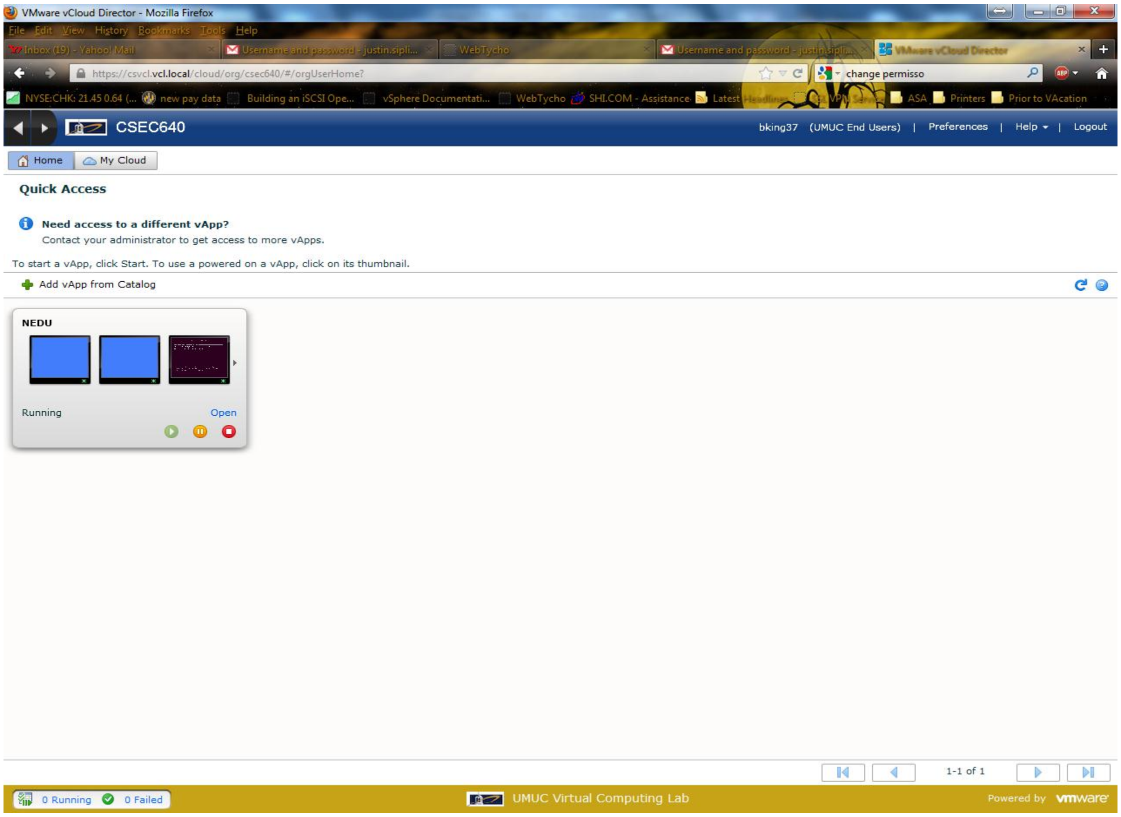


Figure 2: Virtual Lab Platform with Network Boundary.

Example: Vulnerability Scanning Lab

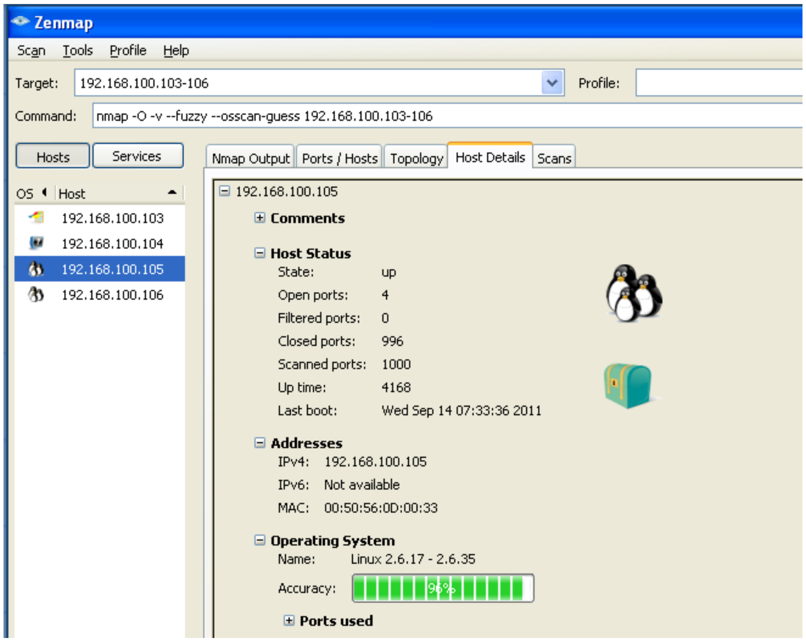
A vulnerability scanning lab is illustrated as an example of how the current UMUC virtual lab platform is used by students. In this lab assignment, students first make a VPN connection to the UMUC virtual lab environment. Through the vCloud Director's web interface, each student imports four operating systems and runs them as VMs in her/his own workspace as shown in Figure 3. The first virtual machine (i.e., VM 1 as shown in Figure 2) is used as a client machine to scan the rest of three virtual machines (i.e. VM 2, VM 3 and VM 4 in Figure 2). VM 2 is a Window server providing services like FTP, Telnet, HTTP, HTTPS, MySQL and more. VM 3 and VM 4 are Linux servers running services like FTP, HTTP, SSL, HTTP, MySQL and DNS. The primary goal of the lab is to provide students with an opportunity to experience the Nmap and Nessus tools (Nmap; Nessus) in order to identify the types of operating systems and services running on VM2, VM3 and VM 4. To successfully complete the lab and answer the lab exercise questions, students must experiment with many features of Nmap and Nessus (Figure 4, 5 and 6 show some Nmap and Nessus features students use to answer lab questions) .


VMware vCloud Director - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Inbox (19) - Yahoo! Mail
Username and password - justin.sipli...
WebTycho
Username and password - justin.sipli...
VMware vCloud Director
https://csvcl.vcl.local/cloud/org/csec640/#/orgUserHome?
change permisso
NYSE:CHK: 21.45 0.64 (...
new pay data
Building an iSCSI Ope...
vSphere Documentati...
WebTycho
SHI.COM - Assistance
Latest Headlines
ASA
Printers
Prior to VAcation
CSEC640
bking37 (UMUC End Users) | Preferences | Help | Logout
Home
My Cloud
Quick Access
Add vApp from Catalog
NEDU
Running
Open
1-1 of 1
0 Running
0 Failed
UMUC Virtual Computing Lab
Powered by vmware

Need access to a different vApp?
Contact your administrator to get access to more vApps.

To start a vApp, click Start. To use a powered on a vApp, click on its thumbnail.

Figure 3: Loading a set of virtual machines via web interface.



Target: 192.168.100.103-106
Profile:
Command: nmap -O -v --fuzzy --osscan-guess 192.168.100.103-106

192.168.100.103
192.168.100.104
192.168.100.105
192.168.100.106

192.168.100.105

Comments

Host Status

State:	up
Open ports:	4
Filtered ports:	0
Closed ports:	996
Scanned ports:	1000
Up time:	4168
Last boot:	Wed Sep 14 07:33:36 2011

Addresses

IPv4:	192.168.100.105
IPv6:	Not available
MAC:	00:50:56:0D:00:33

Operating System

Name:	Linux 2.6.17 - 2.6.35
Accuracy:	96%

Ports used

Figure 4: Nmap - Successful OS guess detection with osscan-guess filter.

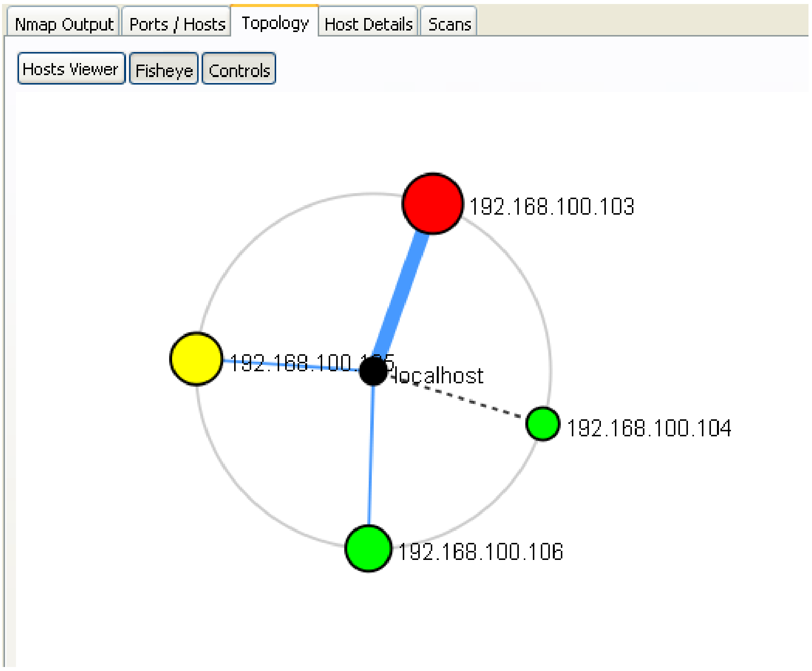
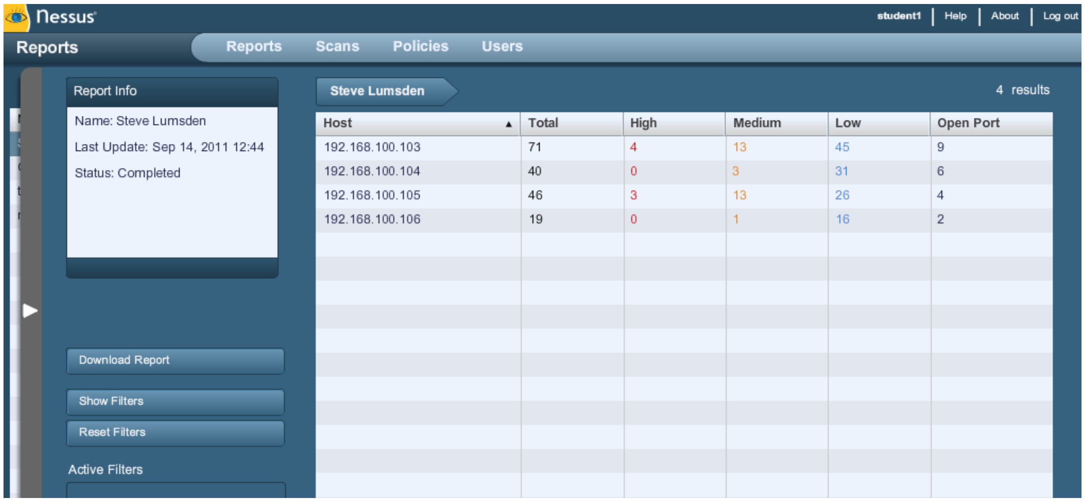


Figure 5: Nmap - Sample Toplogy diagram of the virtual network.



Report Info

Name: Steve Lumsden

Last Update: Sep 14, 2011 12:44

Status: Completed

Steve Lumsden

4 results

Host	Total	High	Medium	Low	Open Port
192.168.100.103	71	4	13	45	9
192.168.100.104	40	0	3	31	6
192.168.100.105	46	3	13	26	4
192.168.100.106	19	0	1	16	2

Figure 6: Sample Nessus report scan result from UMUC virtual lab.

HYBRID APPROACH WITH DEDICATED TEST SERVERS

The major problem with UMUC's current virtual lab setting is performance degradation when a number of concurrent users reaches a certain threshold point. This is mainly due to the large number VMs running on each ESXi server which maximizes CUP and memory usages of the ESXi servers. For instance, for the vulnerability scanning lab, 100 concurrent students mean 400 VMs since 4 dedicated VMs are assigned to each student. Thus, one way to avoid the serious performance slowdown is to reduce a number of running VMs in each host server and build a pool of dedicated standalone test (or virtualized test) servers in the same network as shown in Figure 7. The idea is to move the functionalities of some of VMs to the dedicated standalone test servers, thereby reducing a number of VMs running on each ESXi server. For example, the vulnerability scanning lab can be implemented in a way that only VM 1 is created and dedicated to each student and the functionalities of rest of VMs (i.e., VM 2, VM 3 and VM 4) are moved to the standalone servers as shown in Figure 7. Thus, the set of standalone servers are prebuilt and configured as one window server (serves the same service as VM 2) and two Linux servers (serve the same services as VM 3 and VM 4). Since most security labs typically require one client machine (or machine needed for a significant modification or scanning other machines) and multiple machines providing a set of functions and services for the client machine.

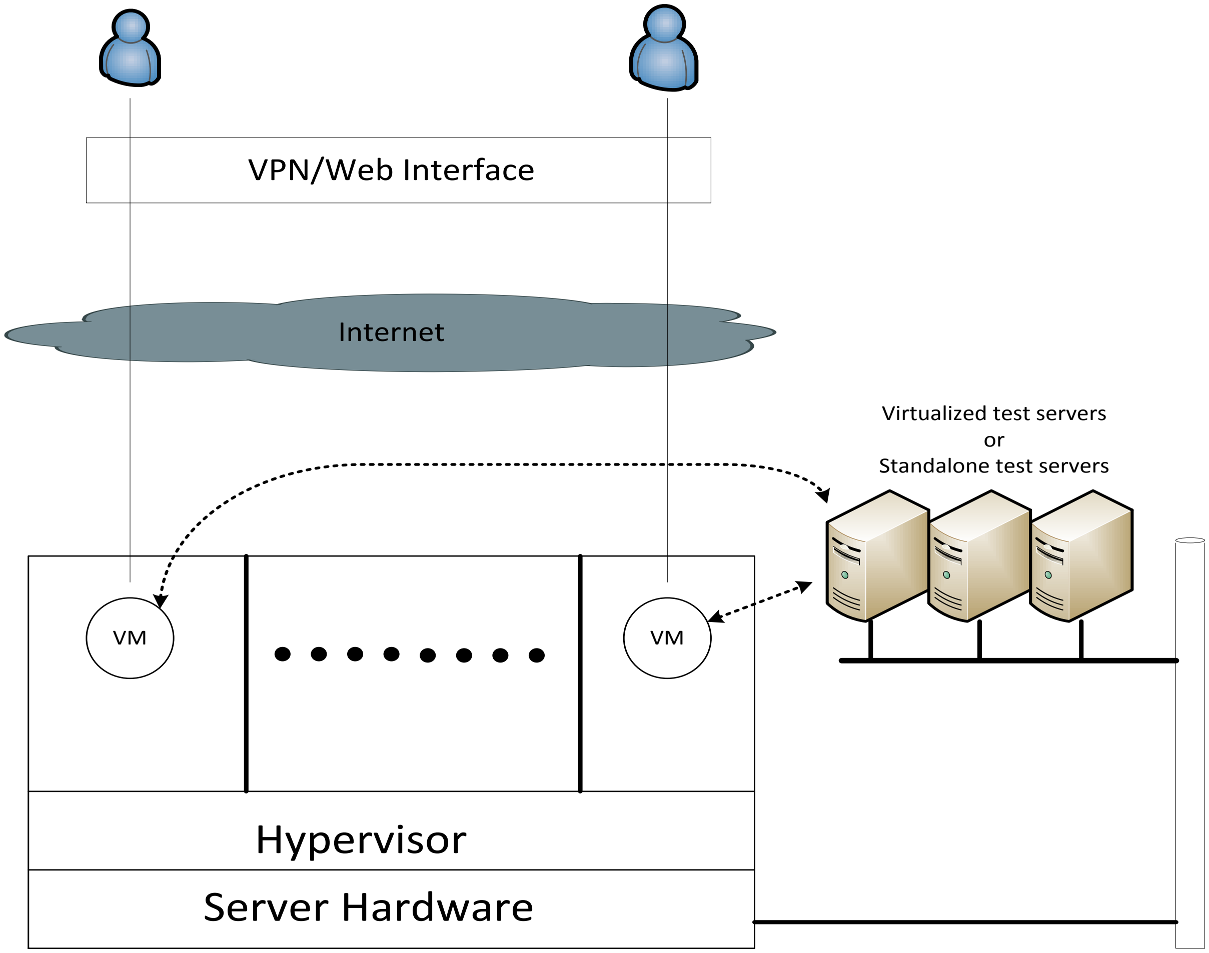


Figure 7: Hybrid approach: VM host servers with dedicated standalone servers.

DESKTOP VIRTUALIZATION APPROACH FOR CYBER LAB

The major advantages of desktop (client side) virtualization approach are (Tao et al., 2010):

1. There is no need for a university to invest to adopt virtual labs and there is also no recurring cost.
2. Virtual machine images can be easily distributed to students and the faculty through web downloading, USB flash disk or DVD disk.

There are a few important drawbacks which are not discussed in Tao's paper (2010):

1. Proprietary software is computer software licensed under exclusive legal right of the copyright holder. The software license is given the right to use the software under certain restriction conditions such as modification or further distribution. To distribute proprietary operating system images (e.g., Window, Mac OS, etc.) as well as proprietary software tools (e.g. Nagios monitoring tool) , a university must contact the operating system and software vendors to resolve any legal issues. Even for free

software tools and operating systems, their distribution agreement must be reviewed and checked. The cost of using proprietary software must be paid before distribution. However, in case of server virtualization, the costs of proprietary software tools and OSs are not recurring since they can be continuously used by students once a university pays their license fees. In addition, software vendors are more willing to make their product free of charge under academic licenses if their software products run on a university server and are strictly controlled by a university IT department.

2. Students may have a problem installing desktop virtualization software or running virtual machines on their PC. For online courses, instructors may not be able to help their students and resolve any installation issues. In general, it is easier for the instructor to monitor the lab activities and for students to seek help in a server side virtualization environment.

3. The desktop virtualization approach may not scale well for labs requiring multiple virtual machines. For example, our vulnerability scanning lab requires at least 3~4 GB RAM. Not all students' personal computer are powerful enough to execute 4~5 virtual machines.

COMPARISON

In this section, we compare four different virtualization approaches to identify their advantages and disadvantages in configuring a virtual lab based on the following attributes: cost, performance, software license conformance, web interface & network connection, management & configuration effort and software installation & support (refer to Table 1).

The following is a list of the attributes and their definition:

- Cost: the cost of implementing a virtual lab environment.

- Performance: the delay (or interaction latency) a user experiences when using a set of security tools in a virtual lab environment.

- Software license conformance: any issue or difficulty to identify the scope of software license and distribution agreement for all the software products deployed and resolve license conflicts. This applies to both open source and commercial software licenses.

- Web interface & network connection: any issue or difficulty students may be facing when they remotely access virtual machines through a web browser or remote client software.

- Management & configuration effort: a level of effort to configure or maintain a virtual lab environment (based upon lab assignments).

- Software installation & support: a level of difficulty students may be facing when installing or updating software packages including desktop virtualization software, security and network tools, etc.

	Cost	Performance	Software License Conformance	Web interface & Network Connection	Management & Configuration effort	Software Installation & Support
A1: Server Virtualization w/o Network Boundary	High	Depending upon a number of concurrent users. Could be severely impacted.	Easy to identify and manage the scope of license issues.	Web & VPN interfaces are required	Medium (relatively simple compared to A2 & A3 approaches).	Minimal (only install VPN client program & a supported web browser.)
A2: Server Virtualization with Network Boundary	High	Depending upon a number of concurrent users. Could be severely impacted.	Easy to identify and manage the scope of license issues.	Web & VPN interfaces are required	High (configure VM host servers with segmentation)	Minimal (only install VPN client program & a supported web browser.)
A3: Server Virtualization – Hybrid approach	High	Depending upon a number of concurrent users. Could be severely impacted. Better than A1 & A2 approaches.	Easy to identify and manage the scope of license issues.	Web & VPN interface are required	High (Higher than A2 approach. Need to configure and maintain additional a set of standalone test servers.)	Minimal (only install VPN client program & a supported web browser.)

A4: Desktop Virtualization	Very Low	Depending on student's PC capacity. Could be Severe.	Hard to identify and manage the scope of license issues (especially software distribution issues).	No special issue	Minimal	Medium (must install and configure desktop virtualization package.)

Table 1: Comparison of four different virtual lab deployment methods.

SUMMARY & FUTURE RESEARCH

As described in this paper, it is possible to design an effective virtual machine architecture to support information security hands on labs for instruction in a highly scalable and cost effective basis. The virtual design approach selected must not only be able to provide acceptable performance, but also provide the users with a consistent environment that is designed to support multiple courses and potentially hundreds of students. In designing and building a virtual lab environment, academic institutions should consider those six attributes (i.e., cost, performance, software license, network connectivity, virtual lab management and support) and select a right deployment model for them.

As an alternative solution to VMware virtualization technology, recently, more and more IT professionals have made the decision to use the open source Kernel-based Virtual Machine (KVM) virtualization infrastructure for migrating IT resources to a virtualized environment. More academic institutions are beginning to use KVM as their choice of virtualization technology (KVM; Yen, 2010). KVM virtualization technology is a open source Linux based virtualization technology. Its biggest potential advantages over traditional virtualization technologies are cost and performance (Younge et al., 2011). There is no cost for installation and it is a part of the Linux kernel. Being a part of the Linux kernel, an assumption can be made about improved performance. Furthermore, KVM which stands for Kernel Virtual Machine is known to provide a very efficient use of memory. KVM can reclaim the memory previously allocated to Linux virtual machines once they become idle allowing more memory to be made available to other active virtual machines and to the system. This occurs even though the idle virtual machines are powered on and not shut off. The speed with which virtual machines were created from a template was always fast and the longest recorded time in our test was 35 seconds. Furthermore, the speed with which they booted to a logon screen was always less than twelve seconds. For this test we used a home PC with 8 GB of RAM and an Intel Core i3 3.1GHz CPU. The KVM virtual machines (Window operating system machines) were only assigned 256Kb of RAM and still delivered these impressive numbers. We noted that the more memory that was allocated to a virtual machine, the quicker the response.

KVM offers administrators a variety of features that can be used to enhance the experience of users of the system. KVM supports network segmentation by allowing the creation of multiple

virtual networks (Appendix shows XML configuration files we used to create two virtual networks). This allows each user to work in their own network workspace without affecting other users. Virtual machine networks can also be configured using NAT or in a flat network. Internet access can be configured or denied using KVM's built in firewall.
The authors contend that Linux KVM is a better fit in the long run because of the following reasons:

- Cost of the deployment is significantly low since KVM is an open source and free. KVM is a right choice for academic institutions with tight budgets.

- It has superior performance because there is minimal to no overhead and its memory management is innovative as we have discussed above.

However, the primary drawback or limitation to KVM is the lack of high quality management tools useful in managing KVM and its new nature to the market. The primary user interface tools are virsh which is a non-user friendly command line tool, and the virtual-manager, a GUI tool which does not support automation that an administrator might need. In our opinion, a feature rich user friendly VM management tool is what lacks most in KVM. Preferably a web management tool that can also provide limited access privileges to users would go a long way to improve KVM adoption in the market place.

REFERENCES

Burd, S., Seazzu, A., & Conway C. (2009). Virtual computing laboratories: A case study with comparisons to physical computing laboratories. *Journal of Information Technology Education: Innovations in Practice,* 8(8), 55-78.

Fuertes, W., Vergara, J. E., & Meneses, F. (2009). Educational platform using virtualization technologies: teaching-learning applications and research uses cases. In *Proceedings II ACE Seminar: Knowledge Construction in Online Collaborative Communities.*

KVM. Kernel Based Virtual Machine. [Computer software]. Retrieved from: http://www.linux-kvm.org/page/Main_Page.

Li, P., Jones, J., & Augustus, K. (2011). Incorporating virtual lab automation systems in IT education. Proceedings of ASEE Annual Conference and Exposition.

Li, P., Toderick, L., & Lunsford, P. (2009). Experiencing virtual computing lab in information technology education. *Proceedings of the 10th ACM conference on SIG-information technology education.*

Nessus. [Vulnerability and configuration tool]. Retrieved from: http://www.tenable.com/products/nessus/.

Nmap. [Network Port Scanner]. Retrieved from: http://nmap.org.

Rajendran, L., Veilumuthu, R., & J, Divya. (2010). A study on the effectiveness of virtual lab in E-learning. *International Journal on Computer Science and Engineering*, 2(6), 2173-2175.

Tao, L., Chen, L., & Lin, C. (2010). Virtual open-source labs for web security education. *Proceedings of the World Congress on Engineering and Computer Science (WCECS).*

VMware. (2009). vSphere Installation and Setup. Retrieved from: https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-pubs.html.

vCloud Director. (2010). vCloud Director Installation and Upgrade Guide. Retrieved from: http://www.vmware.com/support/pubs/vcd_pubs.html

Wang X., Hembroff, G., & Yedica, R. (2010). Using VMware Vcenter lab manager in undergraduate education for system administration and network security. *Proceeding of the 2010 ACM conference on Information Technology Education*, 43-52.

Willems C. & Meinel C. (2008). Tele-Lab IT-Security: An architecture for an online virtual IT security lab. *International Journal of Online Engineering*, 4(2).

Willems, C., & Meinel C. (2012). Online assessment for hands-on cyber security training in a virtual lab. *Proceedings of the 3rd IEEE Global Engineering Education Conference (EDUCON).*

Yen, T. (2010). The management of Linux virtual lab by dual load-balancing. *International Conference on Computers and Industrial Engineering*, 1-5.

Younge, A., Henschel, R., Brown, J., Laszewski, G., Qiu, J., & Fox, G. (2011). Analysis of virtualization technologies for high performance computing environments. *Proceedings of the 4th International Conference on Cloud Computing.*

Zenebe A. & Anyiwo D. (2010). Virtual lab for information assurance education. *Proceedings of the 14th Colloquium for Information Systems Security Education.*

APPENDIX

In this Appendix, we show two xml configuration files which were used to create virtual networks in our KVM test server. These two xml configuration files were read by libvirt (KVM toolkit) to create two virtual segments.

```
<network>
  <name>default</name>
  <bridge name="virbr%d" />
  <forward/>
  <ip address="192.168.122.1" netmask="255.255.255.0">
    <dhcp>
       <range start="192.168.122.2" end="192.168.122.254" />
    </dhcp>
  </ip>
</network>
```

With the above configuration, a default network segment whose IP address ranges from 192.168.122.2 to 192.168.122.254 was created.

```
<network>
  <name>net1</name>
  <uuid>5156cb69-58dd-3fd4-a643-13f1dd859327</uuid>
  <forward mode='nat'/>
  <bridge name='virbr1' stp='on' delay='0' />
  <mac address='52:54:00:F4:87:D9'/>
  <ip address='192.168.100.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.100.128' end='192.168.100.254' />
    </dhcp>
  </ip>
</network>
```

With the above configuration, a virtual network (net1) was created and the IP address of net1 ranges from 192.168.100.128 to 192.168.100.254.