Communications of the IIMA

Volume 12 | Issue 4 Article 5

2012

Virtual Lab 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

A Comparison of Virtual Lab Solutions for Online Cyber Security Education

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

Chinedum Irrechukwu
University of Maryland University College
cirrechukwu@umuc.edu

Patrick Fitzgibbons
University of Maryland University College
patrick.fitzgibbons@umuc.edu

ABSTRACT

In this paper, the authors describe their experience of designing a virtual lab architecture capable of potentially providing thousands of students with a hands-on learning experience in support of an online educational offering. The authors discuss alternative approaches of designing a virtual lab and address the criteria in selecting the optimal deployment method. The authors suggest 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

Over the past decade, 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*, *Desire to Learn*, *Pearson Learning Studio* and proprietary systems such as UMUC's *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 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, Seazzu, & Conway, 2009; Fuertes, Lopez de Vergara, & Meneses, 2009; Li, Jones, & Augustus,

2011; Li, Toderick & Lunsford, 2009; Rajendran, Veilumuthu, & Divya, 2010; Tao, Chen, & Lin, 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 Tomei's (2001) taxonomy and Gardner's (1993) theory of multiple intelligences. Tomei's taxonomy is a widely accepted educational technology model that provides the framework for the proper use of technology in the classroom. The virtual lab technology touches on many of the levels of Tomei's taxonomy and provides students with valuable higher order technology experiences. In the exploration-teaching paradigm, students begin with a directed experience of the fundamental principles underlying the concepts being taught. This experience is then modified systematically to demonstrate refinements of these principles. Ultimately, the students can use the ways these refinements are structured to try out additional modifications on their own initiative. Tomei's Taxonomy is a widely accepted educational technology model that provides the framework for the proper use of technology in the classroom (Powell et al., 2008). Instructional technology at this level of the taxonomy offers numerous strategies that encourage learning by infusing technology into the curriculum. The application of technology for integration represents "the creation of new technology-based materials, combining otherwise disparate technologies to teach" (Tomei, 2001, p. 20). The objective of technology integration is to develop new, previously non-existent, innovative instructional materials to enhance the learning experience.

For example, technology infusion aligns itself well with the decision-making and integration levels of the Tomei's taxonomy. At the decision-making level, students must "apply electronic tools for research and problem solving" (Tomei, 2001, p. 20). Additionally, the virtual lab exercises allow both students and instructors to "consider the consequences of inappropriate uses of technology" and also allows them to "assimilate technology into a personal learning style" (Tomei, 2001, p. 20. These instructional activities align with the Integration level of Tomei's Taxonomy and further reinforce the higher order technology skills that provide students with an enriching online learning experience. Table 3 in **Appendix B** illustrates how the technology infusion of virtual labs for two UMUC Cybersecurity courses corresponds to Tomei's taxonomy.

It is a commonly held belief that students learn more efficiently when engaged in higher order thinking (Gardner, 1993). 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 supporting an online technology based curriculum. As the students are experiencing hands on virtual lab, they are constantly forced to make 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.

University of Maryland University College (UMUC), founded in 1947, is the largest public university in the U.S. with over 90,000 students enrolled in undergraduate and graduate

education. UMUC has been offering online courses extensively since 1985. As cyber attacks are being waged worldwide, the demand for highly skilled cyber security professionals has never been greater. UMUC began offering its graduate level online cyber security degree program in fall 2010; this included launching a virtual cyber laboratory. The virtual cyber lab requirements included the following objectives:

R1. Accessible, secure and seamless access must be provided to the remote virtual cyber lab. This means students will not have to reserve a time to use a virtual resource and that online lab access 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 dedicated 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 images including the required security tools to support lab exercises. In order to minimize requirements for students (e.g., configuring or installing software on their own machines), a pool of Virtual Machines (VMs) along with a cloud based network access were deemed necessary.

R4. Students must have privileged access rights on the virtual machines to execute security or network tools. Note that this implies that students may potentially abuse system resources intentionally or unintentionally. As a result, the virtual lab environment must be monitored to avoid these adverse consequences.

Based upon the above requirements, the UMUC virtual lab platform was built and first deployed in fall 2010. Initially, it consisted of seven Dell Edge Servers with VMware ESXi installed as a hypervisor. A Windows 2008 management server as a vCenter server was installed along with a storage area network and 2 gigabit switches. The servers were connected via multiple gigabit layer links connecting the switches to a storage area network. The vCenter server was used to determine on which server the virtual machines would be placed. This entire virtual lab infrastructure was placed in its own network, completely 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, a significant contingency are affiliated with the U.S. military. In a typical semester approximately 1,000 students in the graduate degree program, are required to participate in two online virtual labs that are included as part of five 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 long graduate term. For example, during some weeks there may be two or more different courses, each consisting of between 10 to 20 sections, that are will be accessing the virtual labs. Figure 1 below displays a number of Virtual Machines running and used by students in the week of September 17 to 23, 2013. This indicates that UMUC virtual cyber lab environment is capable of providing reliable 24x7 access and supporting at least 220 to 230 concurrent virtual machines (about 220 to 230 virtual machines were running concurrently at 6:30 P.M. on September 23, 2012).

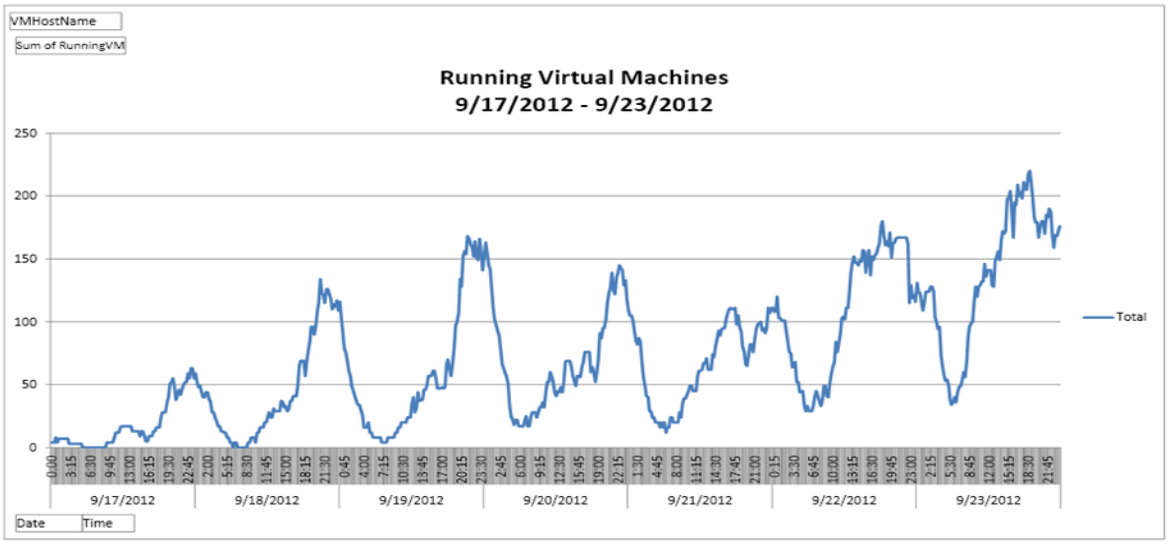

Figure 1: Number of Virtual Machines Running Week of September 17 to 23, 2013.

BACKGROUND

At the most fundamental level, virtualization allows multiple virtual machines to run concurrently on a single computer. Each virtual machine shares the resources of a single computer. 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 available to end users. Among many different types of virtualization technologies, two virtualization technologies are particularly well suited to support 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 whereas client virtualization may not scale well. This is especially the case for labs requiring multiple virtual machines (Refer to the section on desktop virtualization on page 90 for more detail). Server side virtualization software creates Virtual Machines on a remote server (VM host machine). The virtual machine 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, Hembroff, & Yedica, 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 a user to connect remotely to the vCenter server or ESX (or ESXi) host server.

By deploying virtual desktop integration (VDI) 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. Prebuilt images are then 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

As stated previously the initial UMUC virtual cyber 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 network segmentation, hence the virtual machines all had to belong to the same flat network and all shared the same network address. This configuration allowed virtual machines to communicate directly with each other as illustrated in Figure 2. 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.

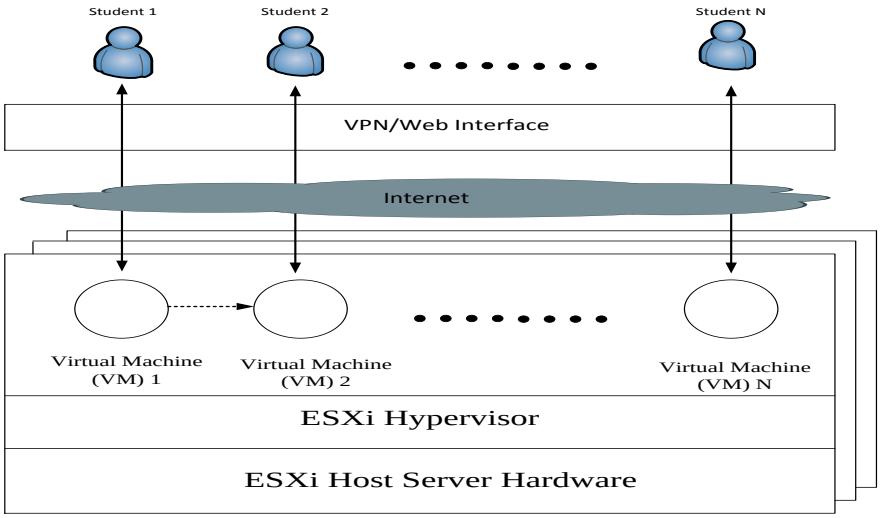

Figure 2: Virtual Lab Platform without Network Boundary.

There are some disadvantages, which include lack of scalability, as well as limited fault tolerance, where a potential high impact event occurs, such as 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 (VMware, 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 3. This may include any number of virtual machines within each virtual network all dedicated to the user. In general, there are virtual machine templates with pre-configured software and tools that are automatically generated when a student logs on and begins a lab exercise (Figure 4). The virtual network and virtual machines are accessible via the student's account and are made available through vCloud director's web interface.

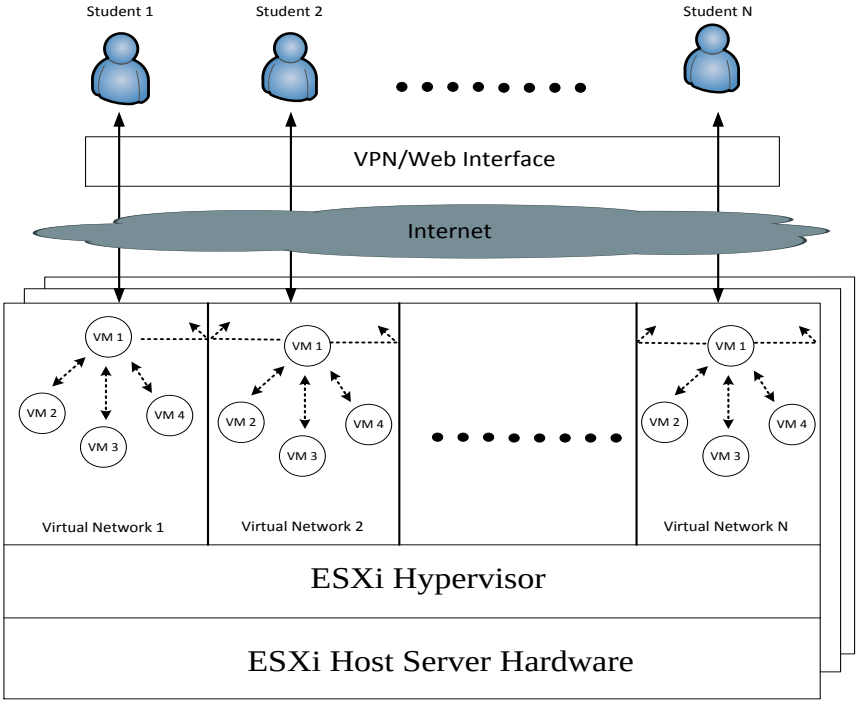

Figure 3: Virtual Lab Platform with Network Boundary.

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 based on user account authentication. This approach is also scalable. For example, it allows for up to 300 maximum concurrent VMs. 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 effectively has their own network that 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.

There are two main drawbacks with the current UMUC virtual cyber lab network configuration. The first is sub-optimal performance and the second is lack of support for some web browsers. According to the vendor, the theoretical maximum of running concurrent virtual machines is approximately 300. However, in reality performance degradation was experienced when the number of running virtual machines approached a number substantially less than 300 (Note: 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.

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 the 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.

Example of a 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 the operating systems as VMs in her/his own workspace as shown in Figure 4. The first virtual machine (i.e., VM 1 (Windows XP) as shown in Figure 3) 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 3). VM 2 is a Window 2008 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, n.d.; Tenable Network Security, n.d.) in order to identify network vulnerabilities in 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 (Figures 5, 6, and 7 show some Nmap and Nessus features students use to answer lab questions).

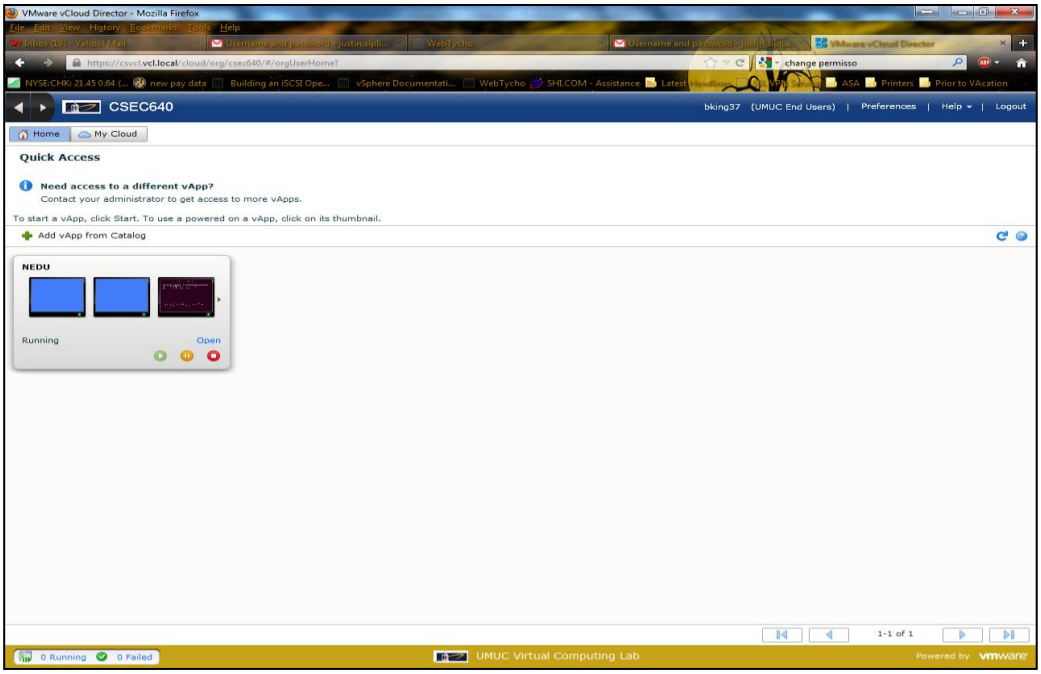

Figure 4: Loading a Set of Virtual Machines (V2-Window Server, V3-Linux, and V4-Linux) via web interface. The MENU panel shows three consoles for V1, V2, and V3.

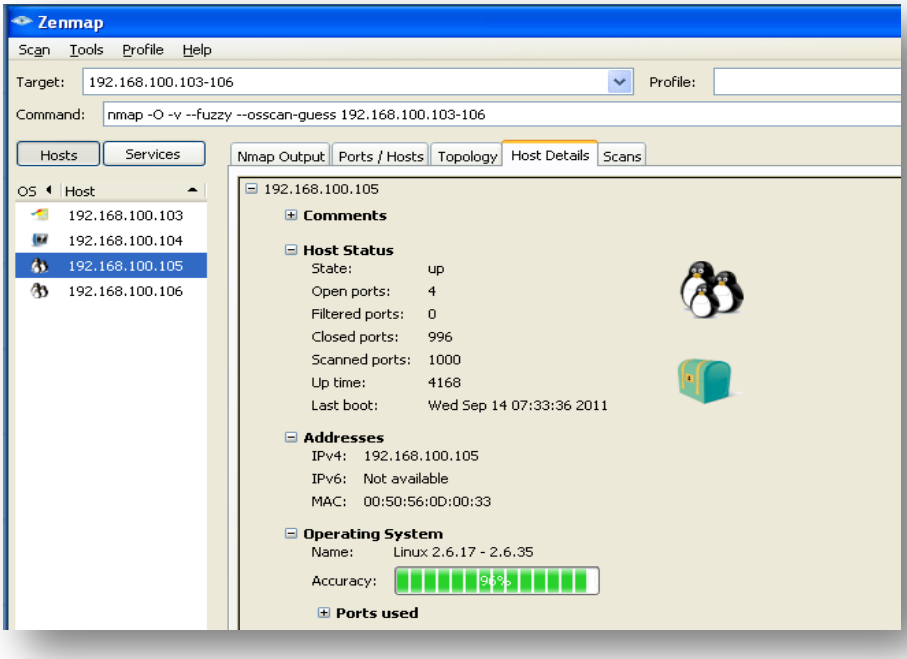

Figure 5: Nmap - Successful OS Guess Detection (with osscan-guess filter).

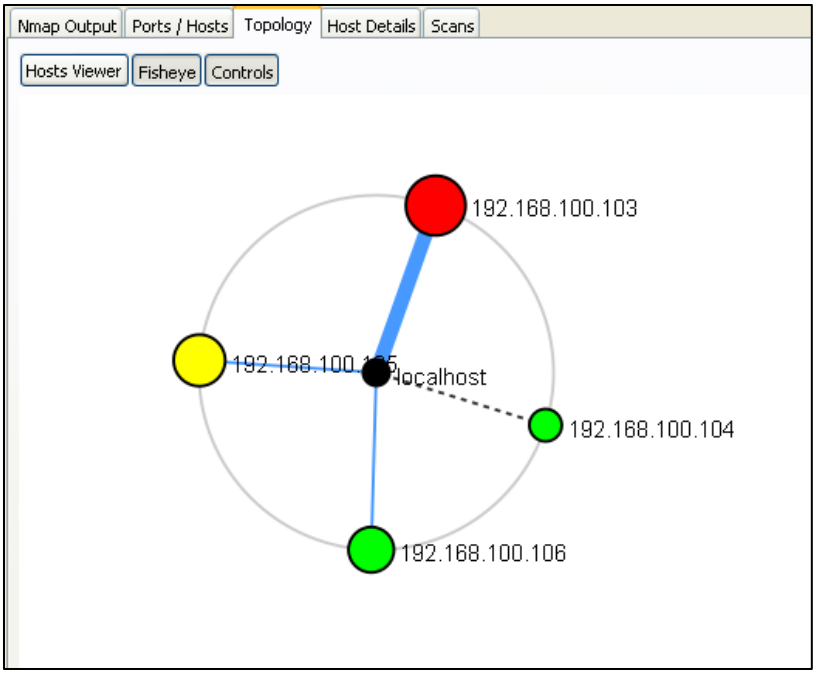

Figure 6: Nmap - Sample Topology Diagram of the Virtual Network.

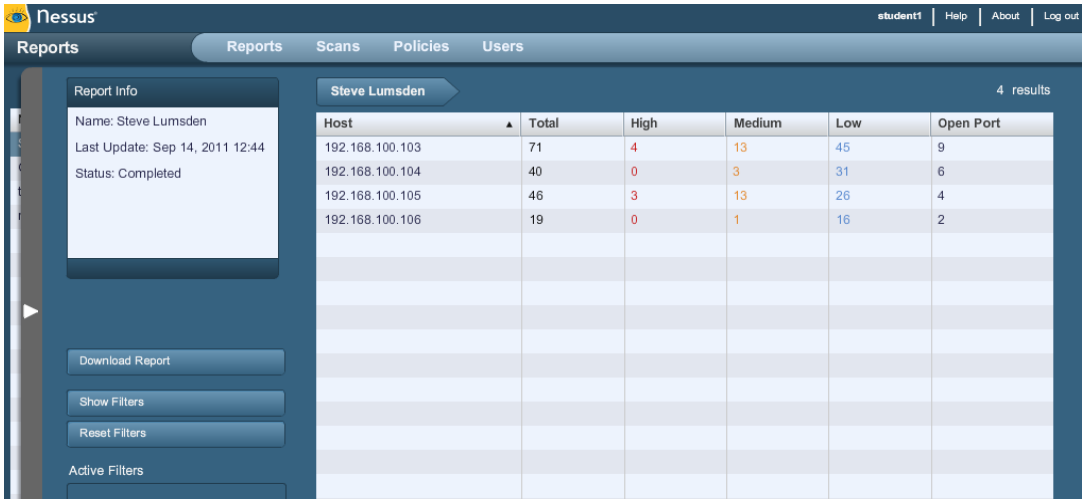

Figure 7: Sample Nessus Report Scan Result from UMUC Virtual Lab.

Hybrid Approach with Dedicated Test Servers

As reported the major downside with UMUC's current virtual cyber lab configuration is performance degradation experienced by users when a number of concurrent users reaches a certain threshold point. This is primarily due to the large number VMs running on each ESXi server, which maximizes CUP and memory usage of the ESXi servers. For instance, for the vulnerability scanning lab, 100 concurrent students mean 400 VMs since four dedicated VMs are assigned to each student. Thus, one way to avoid the serious performance slowdown is to reduce

a number of VMs running in each host server and build a pool of dedicated standalone test (or virtualized test) servers in the same network as shown in Figure 8. The catalyst 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 8. 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.

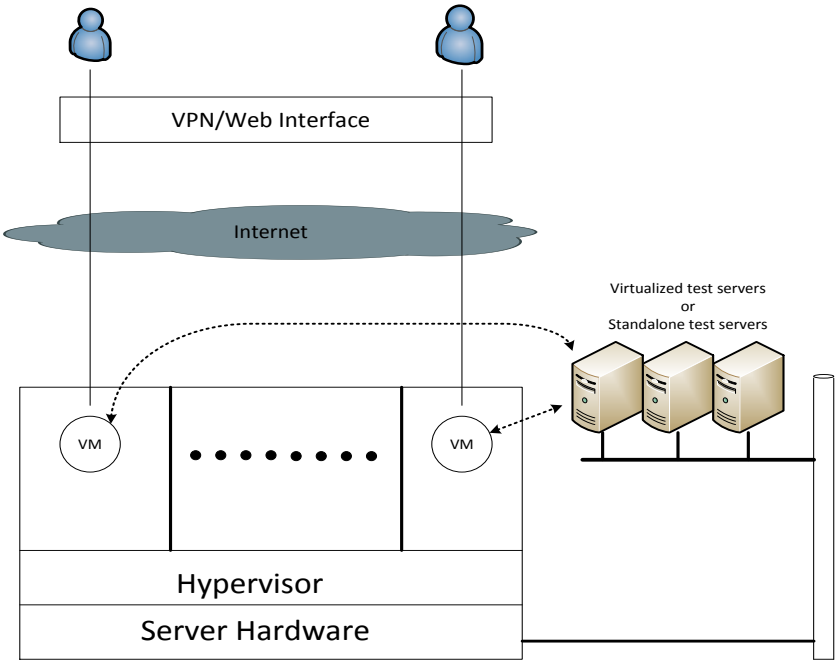

Figure 8: Hybrid Approach: VM Host Servers with Dedicated Standalone Servers.

Desktop Virtualization Approach for Cyber Lab

The major advantages of desktop (client side) virtualization approach are (Tao, Chen, & Lin, 2010):

1. There is no need for a university to invest to adopt virtual labs and there is 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 notable drawbacks, which are not discussed in the paper by Tao, Chen and Lin (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 open source software tools and operating systems, the distribution agreement must be reviewed and approved by the organization. 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 non-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 products free of charge under academic licenses if their software products run on a university server and are strictly controlled by a university's 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, monitoring lab activities is easier for the instructor 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 to 4 GB RAM (in reality, 8GB of RAM is recommended). Not all students' personal computer are powerful enough to execute 4 to 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, management, and 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.
- Management and configuration effort: a level of effort to configure or maintain a virtual lab environment (based upon lab assignments).
- Software installation and 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	Management and Configuration Effort	Software Installation and Support
A1: Server Virtualization w/o Network Boundary				
High (See Table 3)	Depending upon the number of concurrent VMs, performance could be severely impacted if a malicious user exploits a virtual lab environment. Prone to network congestions.	Easy to identify and manage the scope of license issues.	Medium (relatively simple compared to approaches A2 and A3).	Minimal (only install VPN client program and a supported web browser.)
A2: Server Virtualization with Network Boundary				
High (See Table 3)	Depending upon a number of concurrent VMs, performance will not be affected by any network activities by a malicious user. Network congestion can be restricted and no influence is exerted on other VMs.	Easy to identify and manage the scope of license issues.	Medium-High (configure VM host servers with segmentation)	Minimal (only install VPN client program and a supported web browser.)
A3: Server Virtualization – Hybrid Approach				
Highest (See Tables 3 and 4)	Depending upon a number of concurrent VMs, performance could be better than approaches A1 and A2 depending upon lab exercise (a number of concurrent VMs could be significantly reduced).	Easy to identify and manage the scope of license issues.	Highest (Higher than approach A2. Need to configure and maintain additional a set of standalone test servers.	Minimal (only install VPN client program and a supported web browser.)
A4: Desktop Virtualization				
Very Low (See Table 2)	Depending on student's PC capacity, performance could be severe with a low-end PC	Hard to identify and manage the scope of license issues (especially software distribution issues).	Minimal (setting up a web site for download tools/ instructions)	Medium-High (must install and configure desktop virtualization package as well as security/network tools for each student in class. Instructors/students may have serious installation issues. As a result, cannot focus on learning goals. Online students cannot receive an immediate assistance

Table 1: Comparison of Four Virtual Lab Deployment Methods.

Virtualization Approaches	Cost
A1 and A2	$1,459,025 (Table 3)
A3 (Hybrid)	$1,511,617 (Table 3 and 4)
A4	$0 to 220

Table 2: Cost Analysis Summary.

A more in-depth analysis of the comparison table is provided as follows:

- Cost and performance tradeoff: Based upon our experience, we come up with a suggested list of hardware devices, which can support 300 *concurrent* Virtual Machines (VMs) as shown in the Table 4 in **Appendix C**. The Table 4 also shows the associated hardware cost. Note that, in evaluating the cost associated with the virtual lab, we do not include software (e.g., VMware Vsphere license, software maintenance fee, etc.) cost as well as labor charge. In addition, a list of hardware devices and their costs for standalone servers are presented in the Table 5 in **Appendix C**. Desktop virtualization solutions range from VMware workstation ($220 without academic alliance) to VirtualBox (free). As shown in the Table 5, building a cloud-based virtual lab solution capable of supporting a large number of concurrent VMs is not easy and expensive. Note that the cost difference between A1, A2, and A3 is relatively small and we believe that it is worth implementing standalone-dedicated test servers (A3 approach) in the virtual lab to reduce a total number of concurrent VMs. For example, assume that 300 concurrent VMs are being used for the vulnerability scanning lab (Refer to the example of vulnerability scanning lab on page 87). With the hybrid approach, a number of virtual machines running in hypervisor machines can be reduced down to 75 since one fourth of VMs are used as dedicated servers. This huge reduction of VMs can result in performance increase although setting up standalone dedicated servers incurs additional cost and configuration effort. For the A4 approach, students' experiences of virtual lab vary significantly depending upon the capacity of their laptop or desktop PCs.

- Software installation and support: Software installation issues could be a big burden to information technology instructors. Teaching even face-to-face courses, instructors could waste a lot of time helping students with configuration or installation issues. This is because students may potentially have multiple operating systems (e.g., Windows XP, Vista, Windows 7, Windows 8, Ubuntu, CentOS, MAC OS X 10.6, MAC OS X 10.5, etc.) installed on their PCs and lack knowledge of the selected operating system. It is not feasible for instructors to develop a lab manual based upon every operating systems in use. When it comes to online teaching, this problem can escalate when students are unable to receive immediate assistance. This is one reason why it is strongly advised that students should be provided with a set of preconfigured security and network tools. Both students and instructors can than focus on the primary activity and achieve learning goals with minimal delay. In addition, instructors can monitor and help students' lab activities as a root user in a virtual lab environment.

SUMMARY AND FUTURE RESEARCH

As described in this paper, it is feasible to design an effective virtual machine architecture to support virtual 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 thousands 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 the appropriate deployment model based on their individual needs.

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, n.d.; Yen, 2010). KVM virtualization technology is an 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 as it is a part of the Linux kernel. Additionally by being 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 an 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 12 seconds. For this test, we used a PC equipped 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 (n.d.) 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 A** 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.

However, KVM (n.d.) remains untested on a large scale. To fully replace more established technologies such as VMware, Citrix Xen, or Microsoft's Hyper-V, KVM will need to be deployed on a large scale and integrated with an organization's IT infrastructure. Observations need to be made about its performance under different conditions and more information needs to be gathered before the authors can confirm when KVM offers a better solution server virtualization solution.

The authors contend that the Linux KVM is a good candidate for future research for 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 good 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 that 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. The authors intend to pursue future research with KVM and Openstack (Openstack Cloud Software), a web based enterprise management interface. It remains to be seen if there would be significant performance degradation when Openstack (n.d.) is integrated with KVM. It is hoped that web management software such as this would significantly enhance KVM's adoption in the market place.

REFERENCES

Burd, S. D., Seazzu, A. F., & 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., Lopez de Vergara, J. E., & Meneses, F. (2009). Educational platform using virtualization technologies: Teaching-learning applications and research uses cases. In *Proceedings of II ACE Seminar: Knowledge Construction in Online Collaborative Communities.* Albuquerque, NM, USA.

Gardner, H. (1993). *Frames of mind: The theory of multiple intelligences (.* New York, NY: Basic Books.

KVM. (n.d.). *Kernel based virtual machine* [computer software]. Retrieved from: http://www.linux-kvm.org/page/ Main_Page

Li, P., Jones, J. M., & Augustus, K. K. (2011). Incorporating virtual lab automation systems in IT education. In *Proceedings of American Society for Engineering Education Annual Conference and Exposition 2011.* Vancouver, BC, Canada.

Li, P., Toderick, L. W., & Lunsford, P. J. (2009). Experiencing virtual computing lab in information technology education. In *Proceedings of the 10th ACM Conference on SIG-Information Technology Education.* doi: 10.1145/1631728.1631747.

Nmap. (n.d.). *Nmap security scanner* [computer software]. Retrieved from http://nmap.org

Openstack. (n.d.). *Open source software for building private and public clouds* [computer software]. Retrieved from http://www.openstack.org

Powell, V. J. H., Johnson, R. S., Davis, C. T., Turchek, J. C., & Powell, J. C. (2008). Designing hands-on network instruction using virtualization. In Kinshuk, D. G. Sampson, J. M. Spector, P. Isaias, & D. Ifenthaler (Eds.), *Proceedings* of IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2008), Freiburg, Germany. Retrieved from http://celstec.org/system/files/file/conference_proceedings/celda2008/CELDA2008.pdf

Rajendran, L., Veilumuthu, R., & Divya, J. (2010). A study on the effectiveness of virtual lab in E-learning. *International Journal on Computer Science and Engineering, 2*, 2173-2175. Retrieved from http://www.enggjournals.com/ijcse/doc/IJCSE10-02-06-91.pdf

Tao, L., Chen, L. –C., & Lin, C. (2010). Virtual open-source labs for web security education. In *Proceedings of the World Congress on Engineering and Computer Science 2010 (WCECS), San Francisco, CA, USA.* Retrieved from http://www.iaeng.org/publication/WCECS2010/WCECS2010_pp280-285.pdf

Tenable Network Security. (2013). *Nessus vulnerability scanner*. Retrieved from http://www.tenable.com/products/nessus/

Tomei, L. A. (2001). *Teaching digitally: A guide for integrating technology into the classroom curriculum*. Norwood, MA: Christopher-Gordon Publishers.

VMware. (2009). *vSphere installation and setup*. Retrieved from http://pubs.vmware.com/vsphere-51/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-51-installation-setup-guide.pdf

VMware. (2010). *VMware vCloud Director installation and upgrade guide*. Retrieved from http://pubs.vmware.com/vcd-51/index.jsp?topic=%2Fcom.vmware.vcloud.install.doc_51%2FGUID-F14315CC-B373-4A21-A3D9-270FFCF0A417.html

Wang X., Hembroff, G. C., & Yedica, R. (2010). Using VMware Vcenter lab manager in undergraduate education for system administration and network security. In *Proceeding of the 2010 ACM Conference on Information Technology Education*. doi: 10.1145/1867651.1867665

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), 31.

Willems, C., & Meinel C. (2012). Online assessment for hands-on cyber security training in a virtual lab. *2012 IEEE Global Engineering Education Conference (EDUCON)*. doi: 10.1109/EDUCON.2012.6201008

Yen, T. –C. (2010). The management of Linux virtual lab by dual load-balancing. *2010 40th International Conference on Computers and Industrial Engineering*, 1-5.

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

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

APPENDIX A

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.

APPENDIX B

		Course: CSEC 630	Course: CSEC 640
Tomei Taxonomy	**Course Topic**	**Prevention and Protection Strategies in Cybersecurity**	**Monitoring, Auditing , Intrusion Detection, Intrusion Prevention and Penetration Testing**
Class 1	Literacy, understanding technology and components	1. How to virtually and securely log in 2. How to become accustomed to command-line interface 3. How to edit configuration files 4. How to document what happens (access logs, make captures of packets) 5. How to use tools (netstat, ssh,Wireshark, Snort) 6. How to access and use the tools to verify and modify configuration 7. Use support resources provided 8. Distinguishing unicast, multicast, and broadcast addresses	1. How to virtually and securely log in 2. How to enter commands; become accustomed to command-line interface 3. How to edit configuration (command line and files) 4. How to document what happens (access logs, make capture of packets) 5. How to use tools (Snort, Wirehark, ping, netstat, ssh, Nagios, nmap, logger) 6. Use support resources provided
Class 2	Communication, collaboratively work, use technology to form relationship	1. Develop and implement procedures to capture and document packets (in virtual lab environment)	1. Develop and implement procedures on intrusion, scanning, and packet capture 2. Develop and implement IDS testing
Class 3	Decision- making, using technology in new and concrete situations	1. Identify protocols in captured packets 2. Distinguish and Identify addresses in protocols (port numbers, IP addresses, MAC addresses) 3. Identify and classify addresses and masks	1. Identify which ports virtual machine has open (identify and assess vulnerability)
Class 4	Instruction formulate environment	1. Architecture design, IP addresses, multiple interfaces with different properties 2. IDS configuration design	1. Architecture design, IP addresses, interfaces
Class 5	Integration, creating new materials	1. Documentation for intrusion discovery experiences 2. Documentation for different layers of the protocol stack 3. Use access control lists and firewalls to prevent inappropriate uses of IP technology	1. Documentation for intrusion experiences 2. Learn about restriction to use intrusion tools (scans) in controlled environment 3. Discussion of consequences of improper use of tools 4. Learn about inappropriate uses of technology through intrusion
Class 6	Acculturation, value of technology	1. Use open-source software, discuss open-source concepts	1. Discuss ethical uses of network surveillance and packet capture technology 2. Use open-source software, discuss open-source concepts

Table 3: Alignment of Two UMUC Graduate Cybersecurity Courses with Tomei's Taxonomy

APPENDIX C

We come up with the following hardware list in Table 3 based upon the current UMUC virtual lab architecture, which can support 300 concurrent Virtual Machines (operating systems running various kinds of security/network applications). The total hardware cost is about1.4 million dollars ($1,459,025). This clearly shows the cost associated with building a cloud-based virtual lab is high. The hardware list in Table 4 shows additional hardware list to build a hybrid architecture. Note the addition cost ($52,592) is not a significant amount.

Manufacture	*Application Name*	*Comments*	*No.*	*Price Per Unit*
RADWARE	LinkProof 4008	Load Balancers	1	$49,595
Juniper	NetScreen 5200	ns-5200 chassis + fan tray	2	30
Juniper	NetScreen 5000 Series Mgmt Module	Mgmt Module for Netscreen 5200	2	35,000
Juniper	Secure Port Module for NS5000 Series	8xGigE SPM + Copper Xceivers	2	60,000
Juniper	IDP75 IDS / IPS	IDS / IPS Appliance	2	7,200
Dell	EqualLogic PS6010XV	SAN Array	2	270,000
Dell	PowerEdge R710	R710 Server Chassis: 8x2.5" bays, 256GB RAM, 2xX5570 ZEON 2.93GHz Processors, 8x73GB 15k rpm SCSI Drives	8	70,000
Dell	PowerConnect 8024	PowerConnect SAN switch 24x10GB ports, 4xCombo Ports	2	10,000
Dell	PowerConnect SFP+ 3M TwinAx	Dell PowerConnect 3 meter Cables for SAN	4	20
Dell	PowerConnect SFP+ 5M TwinAx	Dell PowerConnect 5 meter Cables for SAN	12	25
Intel	Dual Port 10GB NIC w/SFP+ Cable Int.	SAN Network Interface Cards for Dell R710s	6	60
Cisco	SFP Transceiver Module	SFP compatible Xceiver GBIC	2	325
Cisco	Catalyst 3750	Cisco 3750 Switch w/SFP + IPB Image	4	11,925
Cisco	ASA-5520 Appliance	Cisco ASA-5520 Chassis includes: 2xCAB-AC, 2xSF-ASA-8.0-K8, 2xASA-VPN-CLNT-K9, 2xASA5520-VPN-PL, 2xASA5500-ENCR-K9, 2xSSM-BLANK, 2xASA-180W-PWR-AC, 2xASA-ANYCONN-CSD-K9	2	12,415
Cisco	AIP Security Services Module-20	ASA IPS module	2	6,000
			Total	$1,459,025

Table 4: Recommended Hardware Devices, with Cost, for the A1 and A2 Approaches.

Manufacture	*Application Name*	*Comments*	*No.*	*Price Per Unit*
HP	ProLiant BL465c Server	Linux/Window 2008 server machines providing many common services (HTTP, HTTPS, FTP, DNS, DHCP, Active Directory, LDAP, Telnet, SSH, cryptographic services such as public/private key encryption and digital certificate, etc.).	10	$3,000
Cisco	Switch	Cisco 3750 Switch w/SFP + IPB Image	1	11,925
Cisco	Router	Cisco 3945-SEC/K9	1	10,667
			Total	$52,592

Table 5: Hardware for Standalone Servers for the A3 Approach.