Proceedings of the 14th Colloquium for Information Systems Security Education
Baltimore Marriott Inner Harbor
Baltimore, Maryland June 7 - 9, 2010

Standardization of Virtualization Efforts in Information Assurance Education for Intrusion Detection/Prevention Learning Modules

Subrata Acharya, *Towson University* and Jungyoo Ryoo, *Pennsylvania State University*

Abstract – ***Virtualization is gaining popularity as a way to offer a more flexible platform for providing hands-on laboratory experiences. Until now most organizations have been much decentralized in the manner of building and providing the virtualization infrastructure. This is not a desirable approach due to the duplicated efforts made in reinventing the wheel. A more efficient approach is to have a template or standard from which practitioners can build their respective customized versions. Furthermore, we propose to incorporate the standardization as a hierarchical process within the various layers of security education curriculum. This paper investigates the current virtualization practices in educational settings and tries to incorporate them into a consistent and coherent approach that could eventually lead to the aforementioned template or standard (both within the institution and amongst likeminded institutions). To demonstrate the steps involved in the proposed methodology, we use a concrete example: the development of a learning module on network intrusion detection/prevention systems using virtualization.***

Index terms – Virtualization, Intrusion Detection System (IDS), Intrusion Prevention System (IPS), Checklist, Standardization, hierarchical

I. Introduction and Background

The concept of virtualization provides a framework and/or methodology for dividing the resources of a computer into multiple execution environments, by applying one or more concepts or technologies such as hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, and many more. This powerful concept has special relevance in both academia and industry as it enables tremendous gains in efficiency and cost-effectiveness, along with increased scalability and improved time to resource fulfillment.

Dr. Subrata Acharya is an Assistant Professor in the Department of Computer and Information Sciences, Towson University, Towson, MD. Dr. Jungyoo Ryoo is an Assistant Professor of Information Sciences and Technology at the Pennsylvania State University-Altoona, PA.

In recent times there are various successful solutions *via* server and operating system virtualization. These concepts help the educational curriculum to gradually move from a *resource constrained* theoretical system to a more hands-on real world design, development, and dissemination of information and concepts.

Though there has been a lot of work in the area of server and operating system virtualization, as noted in *VMware* [1], there has been very little focus on network security virtualization. This has caused the educational curriculum to be constrained by the network resources available in order to design and impart network security education.

Furthermore, with the exponential growth of the Internet and with equal growth in the nature and scope of attacks over the Internet, the concept of network security and IDS/IPS is increased in significance in recent times. In addition, with the advent of cyber crimes and the focus on secure information exchange over distributed communication media, there is a tremendous need to train and provide both the theoretical and hands-on experience in this area.

In addition to the above challenges, the most important impediment to security education is the absence of any standardization in the policies, procedures and practices of developing, designing, implementing and operating in the curriculum.

This paper proposes to address the above challenges and needs *via* the standardized design and development of virtualization of network security education in community colleges and universities throughout the nation.

Furthermore, we propose a novel *hierarchical (tiered)* method of standardization of network security educational curriculum in the various entities of the system, starting from the *laboratory technical staff*, to the *instructors* and *teaching assistants* on to the *students* receiving the materials in the classrooms. We also propose a feedback mechanism amongst the various entities in the curriculum in order to enhance the breadth of the proposed solution.

ISBN 1-933510-99-4 ©2010 CISSE

This proposed standardization is enabled via a very simple *"checklist"* system, which can be included in various courses within a given institution and also helps to provide uniformity for the same type of security courses amongst various institutions (community colleges and universities) nationwide.

II. Virtualization In Security Education

An increasing number of Computer Science educators are using virtualization in their classrooms. Virtualization is gaining popularity mainly due to its flexibility physical machines lack. For example, many Computer Science curriculums today require students to be exposed to multiple operating systems such as Windows and Linux distributions. They also require more control to be given to students for experimentation and exploration, which in turn demands constant restorations and reconfigurations of operating systems.

Theoretically, systems with multi-booting capability along with a baseline image control can handle these requirements. However, it is often very time consuming and tedious to conduct these operations on numerous machines. A virtual machine is an attractive alternative to overcome these well known difficulties associated with physical machine-based system management since it can be much more easily deployed and restored to a restoration point at a very large scale. In fact, the deployment and restoration operations could be as simple as copying files from one machine to the other.

Despite its popularity, it is still not easy for a novice user to adopt virtualization in his or her classroom environment. Creating a virtual machine and installing it on multiple machines are usually straightforward although the level of difficulty varies based on the type of virtualization method one uses. However, the complexity of installation and configuration increases exponentially when the virtualization task involves networking virtual machines and configuring them so that they play certain roles such as routers, firewalls, etc.

Many educators find themselves frustrated when they are faced with dealing with all the details required to setting up virtual machines to perform specific functions required in a certain laboratory exercise.

This type of frustration can be spared significantly if a well-known set of virtualization practices can be shared through standardization. For instance, one can standardize the process of creating and configuring virtual machines for a hands-on exercise to install a digital certificate on a server. Accomplishing this entails the use of certificate authority (CA), registration authority (RA), client and server machines connected through a network, and root certificate installation on the client machines. Without a standard, each new user has to go through the similar process of figuring out how every component of the virtualization effort needs to be installed and configured from scratch. By establishing a standard, reusable components such as configuration files and installation scripts can be created once and shared multiple times among many users, which greatly saves time and effort.

The importance of standardization in virtualization has already been recognized by many researchers who have been building communities of the users of virtualization in educational settings. We look into these efforts in the following subsections.

A. Xen Worlds Project ()

Xen as noted in the Xen Worlds Project [2], is an open source implementation of virtualization. Unlike its commercial counterparts such as VMware and Microsoft Windows Virtual PC, Xen is often difficult to install and configure, as is the case with the many open source software products. Xen Worlds Project has been created to facilitate the use of Xen in a classroom environment by automating the deployment of virtual machines particularly in specific networking scenarios. The Xen Worlds Project makes this automation possible by providing the Xen World configuration file and the VM image file. The Xen World configuration file is mainly used to deploy the virtual machines in specific network settings based on a predefined scenario while the VM image file is the actual file that contains a virtualized operating system. One of the biggest strengths of the Xen worlds project lies in the sheer number of virtual machines it can support. With a mid-level enterprise-grade server, it can support hundreds of virtual machines running independently. Another strength of Xen Worlds Projects is its ability to support various network settings necessary for realistic laboratory exercises.

B. V-NetLab Framework

V-NetLab as noted in [3], uses VMware to implement virtualization. It uses Linux as its host operating system. The open source nature of the Linux operating system allows the developers of the V-NetLab to modify its kernel so that packets used for virtual machines can be filtered and not mixed with production packets. This is done at the data link layer. Therefore, packet-filtering is invisible to the users of both virtual machines and physical machines, accomplishing the transparent segregation of production packets and packets generated

by virtual machines, which may contain potentially harmful code snippets.

Considering that many computer laboratory exercises (particularly computer security-related ones) include untrustworthy software such as botnets, Trojan horses, viruses, and worms, the use of this framework makes sense since it either reduces or eliminates the chance of the production network accidentally getting infected with malicious software.

Although it is a great tool to protect the production network from packets generated by the student-run networks for experiments, V-NetLab framework does not provide the prescribed network settings for a certain learning module, made available by the Xen Worlds Project. It simply provides a means to separate the network traffic based on its nature. Although somewhat limiting in its ability to accommodate all the needs of computer science instruction, V-NetLab framework provides an important element required for fulfilling the goal of standardizing the virtualization efforts in computer science education.

C. SOFT ICE Project

The SOFT ICE project provides an inexpensive way to implement virtualization using recycled personal computers. Unlike the other project described so far, the SOFT ICE project does not require expensive server hardware to run multiple virtual machines. All one needs is dual network interface cards (NICs) to allow regular PCs to act as both regular a client machine and a server running virtual machines.

The SOFT ICE project introduces a load balancing application to manage the processing burden imposed on the personal computers hosting virtualization to avoid potential degradation in performance. The learning modules offered by the SOFT ICE project are limited to Operating System and Computer Networking laboratory exercises.

Virtualization of computer education, as noted in [4], [5] [6], especially in security education, has become the order of the day. The current surge in the need for information assurance and cyber security graduates in workforce has caused the severe need for the redesigning of various computer science course curriculums. There is a huge demand towards the design of computer science graduates and undergraduate, specializing in the area of computer security.

In order to address this need the courses and curricula have been redesigned in various community colleges, four year degree universities and graduate schools, and new focused degree programs in the area of computer security have been designed.

Unfortunately, the current computing resource limitations and the lack of non-profit academic-industry alliances have made these majors and courses more theoretical than necessary, as required by the information assurance workforce.

Various researchers have tried to address this challenge *via* server and operating system virtualization with some success. In this paper we focus on the communication *or network security virtualization* in the current educational curriculum design, development and dissemination.

Furthermore we also propose to include the simple *"checklist"* model for the standardization of the virtualized security course design and development procedures and policies within and in-between various institutions throughout the nation.

In our analysis of needs and assessment of the current scenario we have failed to notice any significant categorization of the methods and apparatus to provide security education: *primarily information assurance in both undergraduate and graduate curriculum.*

To this effect the key contribution of the proposed research is to address the scope and need for the *virtualization of security education* and the *standardization of methods and practices* to improve and enhance the curriculum.

The current breed of majors in computer security education is usually geared towards part time students who are working full time either to support their education or families. The current increased need for security trained professional especially in information security and assurance has led these graduates to return back to school to pursue an advanced major or degree to enhance their skills and improve their demand towards the current cyber security needs of the nation.

To this effect virtualization provides a major help in their distance education. Furthermore, it enables the educational institutions to keep the cost down and cater towards a wider audience. It is important to note that this proposed study has conducted a very thorough analysis and evaluation of the current policies, procedures and practices governing the design of computer security courses and curriculum.

The aim of this study was to evaluate the effectiveness of the current courses with respect to the current demand in the workforce and the cost of designing such curriculum.

Based on the analysis we conducted, we propose to address the problem via a two step approach of virtualization and standardization of courses and curriculum in information assurance and security education over various community colleges and universities throughout the nation.

Another important contribution of the proposed research is to impart the virtualization and standardization efforts as a hierarchical mechanism over the various entities in the educational curriculum. We will discuss the proposed tiered/hierarchical approach in our pilot course evaluation in section IV.

III. IDS/IPS VIRTUALIZATION MODULES

To understand and design the proposed Network Security Virtualization Modules, we aim to review the current apparatus and methods in this area in the two-year institutions, community colleges, and large universities. These educational institutions vary greatly in their course curricula, student types and computing resources.

The first step in our study is to research and analyze the current trends in information assurance education in terms of courses, material and platforms.

In our study we have come across a very serious concern amongst *Information Assurance* educators in community colleges and universities in the state. There is a very severe lack of uniformity and standardization in the design, development and dissemination of the course curriculum.

This is more critical and specific to the hands-on laboratory component for courses in the security curriculum. The state of the art design does not provide enough methods to provide a good checklist and standardization method to enable the effective design and dissemination of such courses.

Furthermore, such incoherent policies and practices lead to serious duplication in establishing and ensuring security education in various levels of the curriculum. This results in duplicate efforts in various levels of the education hierarchy – from the technical staff, to the laboratory managers, the instructors, the teaching assistant and the students.

In order to address this problem, we have tried to conduct pilot studies on the design of procedures and practices in the security curriculum in our universities. We aim to incorporate it into four core undergraduate and graduate security courses – namely *Network Security*, *Operating Systems Security*, *Application Software Security* and the *Case Studies in Computer Security*. Currently we have piloted the idea in the core undergraduate and graduate Network Security courses in our universities.

In our approach we aim to design a comprehensive standard methodology in order to design, develop and disseminate network security modules in the security curriculum. We also propose a feedback mechanism established at various levels of hierarchy in order to evaluate and test the strength of the proposed approach.

A. Network Security Modules

In our successful pilot course we focused on all relevant aspects of network security course design and dissemination. The details of each are described in the following subsections.

The modules were designed and developed for network intrusion detection/prevention; as noted in system design laboratory in [7]. Windows XP SP2, Linux 2.6 and Free BSD 7.1 platforms run as virtual machines. In all this design we focus on the concept of efficiency, usability, scalability, and dependability of such systems.

The first module is the *traffic or packet generation*. The module is designed to provide various types of packet generation for the virtual platform. The generator allows for both fixed and variable traffic rates and also provides the facility to generate different packet types. The designed generator can modify packet volume and packet rate. The generator was designed to be platform independent.

The next module is the *traffic monitoring* module. This module helps to provide a window of operational observation for packets for all egress and ingress packets into the virtual network system.

This *traffic monitoring module* also has provisions for time-slicing in both the vertical and horizontal axes. The output from the traffic monitoring module is input to the traffic analyzer and then to the traffic optimizer modules as discussed in the next paragraphs.

The third module is the *traffic capture module*, which is aimed to emulate or create a virtual packet capture scenario for the real-time collection of data as it travels over virtual networks. This module includes a graphical user interface with additional functions such as graphing utilities and traffic generators.

The *traffic filtering module* is the basic virtual IDS module which helps to emulate an access-control list of a firewall. The traffic filtering module is very powerful to

address various types of traffic, volume and ranges of service. The key feature of the filtering module is that it can be designed to provide multi-filter rule generation and implementation, which enhances the use of the virtual IDS module.

The *traffic analyzer module* is the most important module which helps to study and analyze the various packet flows in the virtual security network. The *traffic analyzer module* is a parameter driven system with a strong graphical interface to monitor individual flows with various time granularities, and also provides a robust method to analyze different flows for comparative study and understanding.

The next module is part of the virtual IPS system, which is also known as the *traffic optimization module*. This module helps to provide two types of optimizations, namely – proactive and reactive. All this is part of the proposed virtual network security framework. The proactive component is based on the intelligent algorithms and helps to create new filter properties via an online short term and offline long term approach. This enables the creation of an environment for providing prevention against known vulnerabilities. The reactive component takes in input from the traffic monitor module, reorganizer and the packet filters to address the attack/vulnerability.

The final module is the *virtual topology or geographic information module* which is aimed to provide the flexibility of various commonly used topologies in the virtual framework. The module has a graphical user interface and graphical display with a plug and play mechanism to design and distribute virtual network entities over the choice of network topology.

These modules function together to provide the IDS/IPS experience to users of the system. In the next subsection we will discuss the proposed standardization mechanisms for the use of these methods in the security education curriculum at both intra and inter institution levels.

B. Checklist Template

The checklist helps to provide a template design for all laboratory exercises for the pilot network security course to the technical staff, the teaching assistant and the instructor. The students follow the template in order to conduct the laboratory exercises. These templates are designed to provide a "concise" and "comprehensive" checklist for the policies and procedures towards the design, use and operation of the virtual modules.

C. Standardization

We provide the rules and policies to provide the standardization efforts within the security course curriculum. These standards can also be imported to various other institutions providing similar courses in order to maintain the efficacy of the policy design. The virtual network modules will also be shared amongst the various institutions and help provide a uniform, reusable and duplication free platform to impart security education.

IV. EVALUATION

We evaluate the proposed methods and practices via a layered approach. The first layer is at the *injection level*, also known as the *instructor level*. The second layer is at the *ejection level* – also known as the *student level*. We aim to evaluate the effectiveness of the proposed approaches both within and amongst various community colleges and universities offering similar courses in security education.

A. Instructor Level (Injection Level)

At the instructor level, we conduct periodic reviews and solicit feedback on the usability, efficiency, scalability, and robustness of the proposed virtual security modules and platforms. The feedback is also taken with respect to the design of a common, concise, comprehensive platform for the standardization of the design template used in security education. The laboratory manager or technical staff is also part of the injection level of evaluation.

There is also an evaluation of the effectiveness of the different laboratory exercises as part of this evaluation. The entities in this level provide feedback to redesign, modify and improve the course materials to provide a more comprehensive and uniform course content for the information security curriculum.

B. Student Level (Ejection Level)

The next level of evaluation is at the student level which is via period feedback, student monitoring, retention and contribution to the information assurance workforce nationwide.

The effectiveness of the curriculum would be reflected from the selection and performance of the students in the computer security workforce for the region. With time we aim to expand the concepts and modules to other security courses in both undergraduate and graduate curriculum and evaluate the effectiveness of the proposed methods and techniques within security educational institutions nationwide.

V. Impact and Dissemination

We aim to broaden our scope, and test and implement our proposed design, methodology and techniques in universities and community colleges within the state and also nationwide.

Our goal is to work collaboratively to provide opportunities in information security and assurance curriculum development, course and program sharing, sharing articulation models between community colleges and universities, networking with administrators and faculty involved with security training, security training workshops, and grant funding available for training and equipment. We aim to provide equipment, resources, and faculty development opportunities across the state.

This approach will enable us to accomplish more together than we could have done separately. In the following years, we shall build our security consortium, with our current and ongoing projects in information assurance education, and expand the benefits of this approach to information assurance education development.

VI. Conclusion

In conclusion, this paper demonstrates the strength of virtualization and standardization of information security education. Virtualization offers important opportunities for cost savings and efficiency in computing infrastructure, and for centralized administration and management of resources over various institutions. In future years we expect to broaden our concept of virtualization along with standardization to a national and international audience leading to efficient, portable, scalable and cost effective unified educational curriculum design and dissemination.

VII. References

[1] *http://www.vmware.com/*

[2] *http://home.eng.iastate.edu/~hawklan/xw-index.html*

[3] Weiqing Sun, Varun Katta, Kumar Krishna, and R. Sekar, *V-NetLab: an approach for realizing logically isolated networks for security experiments,* In Proceedings of the conference on Cyber security experimentation and test, pages 1-6, San Jose, CA, 2008.

[4] Magued Iskander, *Virtualized computer labs and the software tools to make it so*, In Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education, pages 447-452, Springer Netherlands, 2008.

[5] A. Gaspar, S. Langevin, and W. D. Armitage, *Virtualization technologies in the undergraduate IT curriculum*, IT Professional, 9(4): 10-17, 2007.

[6] A. Gaspar, S. Langevin, W. Arimtiage, R. Sekar, and T. Daniels, *The Role of Virtualization in Computing Education*, In Proceedings of the 39th SIGCSE technical symposium on Computer science education, pages 131-132, Portland, OR, USA, 2008.

[7] *http://www.csl.sri.com/programs/intrusion/*

[8 *http://www.nist.gov/index.html*