A Case Study on the Standardization and Virtualization Efforts for Effective Computer Security Education: A Layered Community Driven Approach

S. Acharya, *Towson University*, and Jungwoo Ryoo, *Pennsylvania State University*

Abstract
In this paper, we present our experiences of the first phase of our proposed two phase project for incorporating standardization and virtualization approaches in the undergraduate and graduate computer security courses and curriculum at pilot universities and various community college institutions. We detail as a case study our experiences by developing a model for generating faculty, laboratory instructor and most importantly student interest in imparting and disseminating computer security education.

We have introduced and assessed the approaches in undergraduate and graduate courses in the computer science security curriculum. The initial results are very encouraging and have demonstrated improvement in both the dissemination of instruction material and student interest and response to computer security education. The proposed methods are currently being emulated in the other university education curriculum. Efforts have also been initiated towards an online dissemination of the proposed approach in the computer security curriculum. Our long term goal is to ensure the standardization and virtualized dissemination of security education over all potential university security curriculum for it to be beneficial to students, faculty and laboratory instructors as a whole.

Index terms
Standardization, Checklist, Security, Virtualization, Curriculum, Education

I. INTRODUCTION

There are numerous subtle stumbling blocks for the first adaptors of different virtualization technologies in their classrooms. Many educators discover that they need to invest a significant amount of their time to find out the best virtualization solution for their unique situations and to troubleshoot problems arising during implementation efforts. Particularly if there are too many technical hurdles to overcome both before and during their class time, they often get frustrated and simply give up. We believe that these unnecessary struggles can be avoided by: (1) identifying well-known solutions to common virtualization problems in computer security education, (2) archiving them in a standardized and hence more sharable manner and finally (3) making these virtualization solutions available in a social-network setting.

Therefore, the objectives of the first phase of this project is (1) collects and analyzes the existing methods of virtualization used by the computer security educators,(2) to systematically define recurring and significant virtualization problems, and (3) to then identifies the best practices to solve these problems (e.g. checklists/templates).

The second phase of this project will focus on the development of online infrastructures to promote the exchange of computer and network security learning modules that use virtualization. More specifically, the online infrastructure will be a social networking portal where educators can freely download/upload virtualization-based computer learning modules. The first phase tasks lay foundation for the second phase since they provide an overall structure for categorizing various learning modules according to the nature of virtualization problems they solve.

We believe that this research will have an immediate direct impact on the undergraduate and graduate education in various universities and community colleges across the nation. The academic and pedagogical findings from this project will also have a positive influence on the more widespread adoption of virtualization in computer security education. Once developed, the online, virtualization-based computer security education portal will be available to faculty members at any institution, who are interested in downloading and uploading learning modules.

Dr. Subrata Acharya is an Assistant Professor in the department of Computer and Information Sciences at Towson University, MD and Dr. Jungwoo Ryoo is an Assistant Professor in the Information Sciences and Technology department at the Pennsylvania State University (Altoona College), PA.

ISBN 1-933510-96-X/$15.00

II. Background And Related Work

Even though there has been a lot of work in the area of client and server system virtualization, as noted in [1], there has been very little focus on computer and network security virtualization. This lack of investment has caused the educational curriculum to be constrained by the physical network resources available to design and impart network security education. Furthermore, with the exponential growth of the Internet and with the equal growth in the nature and scope of attacks over the Internet, the importance of network security in computer security education has increased significantly.

In addition, with the spike in cyber crimes and the focus on secure information exchanges over distributed communication media, there is a tremendous need to provide both the theoretical and hands-on education in this area. In addition to these challenges, the most important impediment to security education is the absence of any standardization in the policies, procedures, guidelines, and daily practices of designing, implementing and maintaining a computer security curriculum that heavily relies on virtualization. As a result, sharing the know-how of an effective adoption of virtualization in computer security education is much more difficult than it should be.

In our survey of the current virtualization scenarios, we have found the lack of significant cataloguing and categorization of the methods and apparatuses to provide computer security education. Therefore, one of the key contributions of this research is to address the scope and need for the virtualization of computer security education and the standardization of methods and practices to improve and enhance their reusability and eventually the curriculum.

In this regard, virtualization provides a major help in the retooling of workers through distance education. Furthermore, it enables the educational institutions to keep the cost down and cater a wider audience. It is important to note that this research incorporates a thorough and comprehensive analysis and evaluation of the current policies, procedures, guidelines, and practices governing the design and implementation of computer security courses and curricula.

III. Approach

To address all the above concerns, needs, and challenges, this research proposes a novel *hierarchical (tiered)* method of standardizing the computer security education curriculum in the various entities of an institution, 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 to enhance the reach of the proposed solution. The proposed standardization is enabled by a *"checklist"* system, which can be included within various courses in a given institution and also help provide uniformity for similar security courses among various institutions (community colleges and universities) nationwide. Currently, we have worked on the first phase and are in the process of starting the second phase of the project.

Phase One: During this phase, we have developed virtualization checklists for different stakeholders (e.g., students, instructors, and lab technicians) to be involved in a learning module. To develop the checklist, we have:

- Collected and analyzed the existing methods of virtualization commonly used by the computer security educators,
- Systematically defined recurring and significant virtualization problems, and
- Identified the best practices to solve these problems.

Phase Two: During this phase, we will develop an online infrastructure that allows the seamless exchanges of computer security learning modules as well as virtualization techniques that underlie them among community colleges and universities throughout the nation. To create the online repository, we will develop standardized methods for documenting:

- The learning objectives,
- The description of the learning processes and its expected outcomes,
- The specification of the virtualization architecture, design, and implementation, and
- The archiving of all the reusable artifacts for the learning module.

The checklist developed in the first phase is also included in this web repository. The checklist provides a template design for all laboratory exercises 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 virtualization-based learning modules. To create a checklist for the different stakeholders working on virtualization-based computer security modules, we have reviewed the current apparatuses and methods in this area in community colleges and universities. These educational institutions vary greatly in their course curricula, student demographics, and computing resources.

The first task during the first phase of our project was to research and analyze the current trends in computer security education in terms of courses, materials, and

ISBN 1-933510-96-X/$15.00

platforms. In our prior study we have come across an alarming trend among computer security education programs in community colleges and universities in the U.S. There is a very severe lack of uniformity and standardization in the design, development and dissemination of the course curricula. This is more critical and specific to the hands-on laboratory component. The state of the art design does not provide anything close to an acceptable checklist for standardization.

Furthermore, such incoherent policies and practices lead to serious duplication in establishing and maintaining the quality of computer security education at the various levels of the curriculum. This results in duplicate efforts in the various levels of the education hierarchy–from the technical staff, to the laboratory managers, the instructors, the teaching assistants, and the students. Thus, after the completion of the first phase, we have piloted the adoption of our checklist concept in the computer security curriculum at our university. We have incorporated the approaches in our core undergraduate and graduate computer security courses such as Network Security, Operating System Security, and Application Software Security.

The final goal of the proposed research is the design of an online infrastructure for the dissemination of the proposed approaches over the various universities nationwide. The design is intended to be a social networking portal specializing in sharing information on the use of virtualization in computer security education.

Standardization is critical to allow the seamless exchange of the learning modules among various parties. For instance, Hyper Text Mark-up Language (HTML) forms will be prepared to document the learning objectives, processes, and expected outcomes. Natural language descriptions of the objectives and outcomes are sufficient, but a more rigorous specification is necessary for the description of the learning processes since this element of the virtualization-based learning module sharing elucidates the role of virtualization in the context of the learning module. The learning process also determines the exact timing and dose of virtualization throughout the instructional cycle and significantly affects the effectiveness of the delivery of the learning module.

The online community using the repository will provide the rules and policies for the standardization efforts within the computer security course curricula. These standards can also be imported to various institutions providing similar courses to maintain the coherence and consistency of the learning modules. The learning modules themselves will also be shared amongst the various institutions and provide a uniform, reusable, and duplication-free platform to impart security education. In the following section, we provide an example scenario of an intrusion detection/prevention system-learning module.

IV. IDS/IPS Modules

The keen interest in network security virtualization has been growing steadily among the networking community in the last few years. Network security virtualization opens up huge opportunities in future secure Internet design by aiding the deployment of varied architectures, protocols, and approaches over a shared physical infrastructure. This in-turn helps in the understanding of such systems and helps the design and development of robust and secure Internet for the future.

To this effect computer security educators must balance the need to protect the stability, availability, and security of computer laboratories with the emphasis on effective hands-on learning objectives for the various courses in the security curriculum. In courses where students need to install, configure, and build secure computer systems, it is imperative that there exists a structured approach to designing, developing and disseminating learning objectives to the students. With this thought in mind, we would like to discuss our pilot Intrusion Detection System (Firewall) assignment scenario in an undergraduate and graduate computer security curriculum.

Typical network security virtualization is composed of two main components – link virtualization and node virtualization. For our assignment the virtualized node component performs both functions – *routing and filtering*. Figure 1 presents the virtualization model for our pilot classroom laboratory exercise. Link virtualization aids in transporting multiple separate virtual links over a shared physical link.

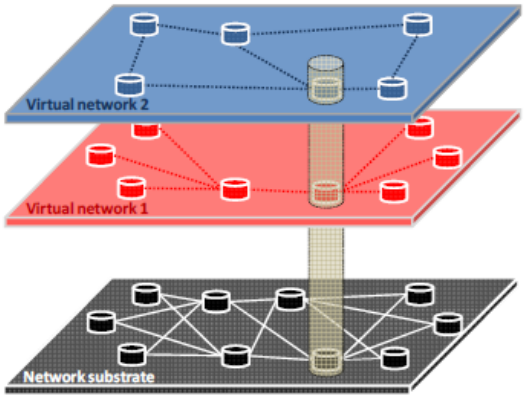

Figure 1: Virtualization Model

Node virtualization is done by partitioning the hardware resources. Physical resources of a node (IDS router) such as CPU, memory, storage capacity, link bandwidth are

partitioned into slices, and each slice is allocated to a virtual node (IDS router). The functionalities of routing and filtering are incorporated as part of the virtual node design.

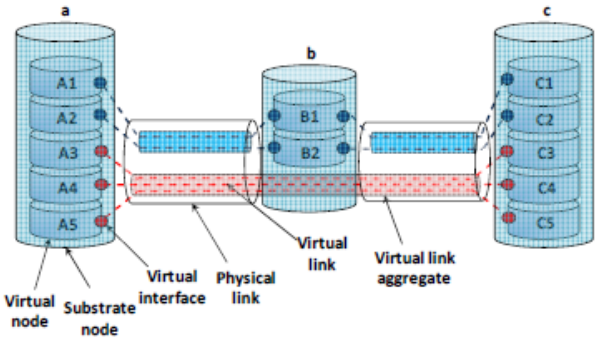

Figure 2: Virtualization Elements – Node (IDS Router) and Link

Figure 2 illustrates the typical virtual node and link elements. Virtual management module provides key features such as performance and fairness in allocation among various virtual elements. This structure in turn creates virtual networks, which is functionally equivalent to a physical network. In the next section we detail the standardization efforts via our proposed checklist approach.

V. Standardization Via Checklist Approach

The concept of *checklists* has been used in many applications – most notably in aviation safety and surgical methods - to reduce the likelihood of human errors. While pre-flight checklists have been considered a key method in improving airline safety, checklists are increasingly used in software assurance. Well-developed checklists serve as a reminder list and help ensure consistency and completeness [2], [3].

A virtualization checklist is a well-defined set of procedures for the standardization of computer security education. Students will be asked to apply the checklists to deliverables as part of a laboratory assignment. By requiring students to complete a concrete series of steps in a specified order, checklists reduce the likelihood of omitting a key feature and provide a quantifiable list of criteria to effectively complete the deliverables and to aid successful student learning [3], [4].

Checklists act as a form of self-assessment; they reinforce security principles [2], [3] and helping students internalize key concepts through critical reflection. The content of our checklists is taken from well-established security mantras that guide software security. The checklist for the design of "*Intrusion Detection System Design*" laboratory that has been piloted in the graduate Network Security (COSC 734) and the undergraduate Operation Systems Security (COSC 440) courses.

VI. Learning Objectives And Outcomes

The goal of the pilot assignment is to design a network intrusion detection system in a virtualized environment using the standardized checklist provided by the instructor. We conducted our preliminary evaluation on the students of the pilot classes in two iterations – *without and with standardization checklists/template.* The objective of the assignment was the design of a virtual IDS system. The students imported the required virtual modules as necessary to enable the functionality of node/IDS router. The students fired up the virtual machine and started building the required intrusion detection system. The students then designed a client and used the traffic-generator module to send packets and check the configuration of their filtering table. The students then write a report of their observation and also provide feedback on the quality of the checklist design.

The learning process and expected outcomes for the IDS learning module are documented in the checklists for students and instructors. There total number of students for the undergraduate and graduate classes was 15 and 20 respectively. In the following we discuss the security checklists. Due ease of understanding and space limitations we provide a very simplified and course level demonstration of the security checklist. Part of the goal for the project is to improve the quality and specifications for the checklists to provide a precise and standardized template for various courses in computer security education.

Student Checklist

- Step 1: Instantiate the Virtual Machine Environment
- Step 2: Import required modules from the Instructor repository
- Step 3: Instantiate the link and node module
- Step 4: Invoke the packet generation and monitoring
- Step 5: Analyze the operation and prepare the report

Instructor/Laboratory Staff Checklist

- Step 1: Establish the hardware and software requirements for the assignment
- Step 2: Import the security modules from the project online repository
- Step 3: Enable the required IDS modules for relevant assignment
- Step 4: Prepare the Student checklist and directions
- Step 5: Monitor the student progress and assess student report

ISBN 1-933510-96-X/$15.00

We conducted a preliminary evaluation of the effectiveness of the proposed approach of security education. We identified the basic metric of effectiveness to student hands-on education by conducting a test after the completion of the class assignment and deliverable from students. Figure 3 illustrates the strength of the proposed approach in the two pilot courses. The Y-axis represents the number of students passing the test with and without the use of the checklist procedure. The results clearly demonstrate the impact of the proposed approach. We aim to conduct similar evaluation studies for both students and instructors for other security assignments and courses amongst all partnering institutions.

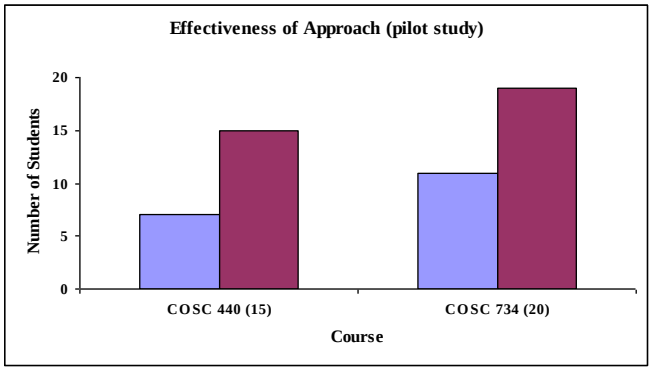

Figure 3: Evaluation Outcome

In this learning scenario, we focused on all relevant aspects of computer security course design and dissemination. The modules were designed and developed for network intrusion detection/prevention (Sebring et al. 1988, SRI International 2010) over virtualization platforms built on Windows XP SP2, Linux 2.6, and Free BSD 7.1. The learning modules focus on the concept of efficiency, usability, scalability, and dependability of such systems. The ***reusable artifacts*** in the IDS learning module include the virtualization modules and the checklists for students and instructors.

VII. Layered (Tiered) Evaluation

The plan is to implement the standardization methods and practices (i.e., the checklists and the online repository) 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. The overall plan is to evaluate the effectiveness of the proposed approaches both within and among various community colleges and universities offering similar courses in computer security education.

Instructor/Laboratory Staff Level (Injection Level)
At the instructor level, we conduct periodic reviews and collect feedback on the usability, efficiency, scalability, and robustness of the proposed checklists and the online repository for the virtualization-based security modules. The feedback is also collected with respect to the design of a common, concise, comprehensive platform for the standardization of the design template used in computer security education. The laboratory manager or technical staff is also part of the injection level of evaluation. The entities at this level provide feedback to redesign, modify and improve our standardization approaches for providing a more comprehensive and uniform course content.

Student Level (Ejection Level)
The next level of evaluation is at the student level and is done by collecting periodic feedback, monitoring students, analyzing retention rates, and measuring contribution to the computer security workforce nationwide. The effectiveness of the virtualization-based curriculum would be reflected from the selection and performance of the students in the computer security workforce for the region.

Thus, the evaluation plan consists of two distinct phases: ***formative and summative***.

Formative evaluation – During project execution
- Evaluate and review materials as they are developed
- After pilot testing, review and assess the effectiveness of the materials
- Develop and evaluate hands-on expertise and ability to apply learned principles among students and involved instructors/laboratory assistants
- Collaborate with the project teams to provide feedback on the development
- Meet periodically to discuss the approaches and methodologies to develop and impart the project goals

Summative evaluation – At specified time periods during project implementation
- Review and assess the effectiveness of the developed materials and instructional approaches at all institutions
- Follow up and provide a comparative effectiveness assessment of the disseminated materials in classrooms and laboratory environments at all institutions
- Evaluate hands-on security expertise and ability to apply learned principles at student and instructor level

In summary, the overall outcomes would be the degree by which we:
- Increase ease of use for instructors/laboratory staff's through standardization and reuse when they try to adopt virtualization to more effectively teach computer security concepts and best practices in a hands-on exercise

ISBN 1-933510-96-X/$15.00

- Improve the student learning experience by increasing the effectiveness and efficiency of how the learning modules are delivered
- Increase the enrollment of students into the computer security curriculum
- Increase the student retention rate in the area of computer security
- Increase the hiring of the students for internships and full-time jobs
- Contribute to security education knowledge base
- Help build the community of computer security educators using virtualization

VIII. Dissemination

The dissemination plan is to ultimately expand the proposed research approaches and outcomes to numerous four-year and two-year institutions nationwide. The end result is a highly-replicable model that has been field-tested across multiple institutions with varying academic and socio-economic populations. All repository material and related training materials, including video recordings of training sessions will be stored on a publicly-available website. This repository will support distribution to all interested national participants. Participants who use the materials will be encouraged to submit original or modified security modules for inclusion in the online repository. An "open source" copyright model (GNU General Public License) will be used to encourage dissemination. All deliverables will be submitted as a collection for inclusion in the National Science Digital Library (NSDL).

Our goal is to work collaboratively to provide opportunities in computer security curricula development, course and program sharing, sharing articulation models between community colleges and universities, and networking between administrators and faculty members involved in computer security education. We aim to provide equipment, resources, and faculty development opportunities across various universities and community colleges nationwide.

IX. References

[1] *http://www.vmware.com/*
[2] M. Bishop and D. Frincke, *Teaching Secure Programming*, IEEE Security and Privacy 3(5) pp. 54-56, Sep, 2005.
[3] M. Bishop, *Teaching Assurance Using Checklists*, Seventh Workshop on Education in Computer Security, Monterey, CA, 2006.
[4] 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.
[5] 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.
[6] 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.
[7] A. Gaspar, S. Langevin, and W. D. Armitage, *Virtualization technologies in the undergraduate IT curriculum*, IT Professional, 9(4): 10-17, 2007.
[8] *http://www.csl.sri.com/programs/intrusion/*
[9] *http://home.eng.iastate.edu/~hawklan/xw-index.html*
[10] *CERT/CC. CERT/CC Statistics*, http://www.cert.org/stats/cert_stats.html , Jan 2008.
[11] J. Davis and M. Dark, *Teaching Students to Design Secure Systems*, *IEE Security and Privacy*, Vol. 1, Num. 2, March 2003.
[12] R. Vaughn, Jr., *Application of security to the computing science classroom*, Proceedings of the thirty-first SIGCSE technical symposium on Computer science education, p.90-94, Austin, TX, March, 2000.
[13] A. Yasinsac and J.T. McDonald, *Foundations for Security Awareness Curriculum*, Proceedings of the 39th Hawaii International Conference in System Sciences, 2006.
[14] B. Olson, *BRAC effort priced in the billions*, Lt. Gov. Brown unveils final 'action plan' focusing on education, transportation. Baltimore Sun, December 18, 2007.
[15] K. L. Kroeker, *The evolution of virtualization*, Communications ACM, 52(3), 18-20, 2009.
[16] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.), Addison-Wesley Professional.
[17] Michael M. Sebring and R. Alan Whitehurst, *Expert Systems in Intrusion Detection: A Case Study*, the 11th National Computer Security Conference, October, 1988.
[18] S. Wippermann and R. Vogel, *Communicating didactic knowledge in university education*, *In L. Cantoni & C. McLoughlin (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, (pp. 3231-3234), Chesapeake, VA, 2004.
[19] http://www.sri.com/news/digest/