Cybersecurity Education: RunLabs Rapidly Create Virtualized Labs Based on a Simple Configuration File

Dr. Connie Justice, Indiana University-Purdue University, Indianapolis

Dr. Connie Justice is a Clinical Associate Professor in Computer and Information Technology (CIT) at the Purdue School of Engineering and Technology at Indiana University Purdue University Indianapolis (IUPUI) and a faculty member of the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University. Professor Justice has over 20 years experience in the computer and systems engineering field. Professor Justice is a Certified Information Systems Security Professional, CISSP. She created the networking option and security option for CIT majors and a Network Security Certificate Program. She has also designed and modified many courses in networking and networking security. Professor Justice is noted for her creation of the Living Lab, an experiential learning environment where students gain real world experience running an IT business.

Dr. Justice takes extreme pride and is a great innovator in the area of experiential learning and service. Experiential learning and service contributes to the integration of theory and application by creating an environment where the students learn by doing or apply their theory in service learning projects, practica, internships, games, and simulations. The Living Lab for CIT was created out of the need to provide a business environment for students to give them a taste of a ”real” IT environment. A secondary purpose is to provide service to internal and external clients. The Living Lab has served many internal and external clients.

Dr. Justice has consulted for and managed IT departments in small and medium sized businesses. Her areas of research include: experiential and service learning, information and security risk assessment, risk management, digital forensics, network security, network and systems engineering, network and systems administration, and networking and security course development.

Rushabh Vyas, Indiana University-Purdue University, Indianapolis

RunLabs: Rapidly Create Virtualized Labs Based on a Simple Configuration File

Rushabh Vyas, Dr. Connie Justice
Purdue School of Engineering and Technology, IUPUI
Indianapolis, IN 46202, USA
rushvyas@iupui.edu, cjustice@iupui.edu

Abstract

The cornerstone in educating the future workforce in cybersecurity in higher education is experiential learning. Cybersecurity competitions are shown to have the potential to increase the workforce and encourage students to pursue the field of cybersecurity. Virtual laboratories allow emulating real life cyber threats and rapid generation of multiple scenarios and infrastructures. The purpose of RunLabs project was to create a lab infrastructure to allow instructors to generate virtualized environments rapidly. Instructors can create virtual lab for students easily, with a simple configuration file. The methods used for RunLabs creation consist of a javascript object notation (JSON) configuration file that creates virtual machines with specified network configuration. In addition, it creates virtual network computing (VNC) service for each virtual machine with a random password, which allows students to be able to access the virtual machines and work on their exercises. RunLabs has a web-based user interface for administration and an application programming interface (API). The API allows additional tools to be written around RunLabs. The administrator can reboot virtual machines, change VNC passwords. If defined in the configuration file, the administrator can create generic routing encapsulation (GRE) tunnel for the virtual machines across multiple hosts. RunLabs project used Python, Flask, SQLite, Minimega, KVM/QEMU, and OpenVSwitch as its backbone software. The analysis showed that the virtual machine host can capture virtual machines network traffic; and by default, any changes made to the virtual machines are not saved to the virtual disk. Due to the way KVM/QEMU work, one virtual disk can be used to spin up multiple virtual machines. Use case scenarios for this project included malware analysis, virtualized penetration testing network, and capture the flag competitions. Future development includes creating a virtual machine repository, bug fixes, and an option to save changes to the virtual disk.

Background and analysis of previous approaches

The National Initiative for Cybersecurity Education (NICE) published a National Institute of Standard and Technology (NIST) special publication 800-181 defining a cybersecurity framework (NCWF) for the workforce. This document outlined the knowledge, skills, and abilities (KSAs) employers expect cybersecurity professional to possess for work roles (Newhouse, 2016). The NCWF consisted of seven categories, with 32 specialty areas, and within each specialty area there were associated KSAs and tasks (Newhouse, 2016). The cornerstone in educating the future workforce in cybersecurity in higher education is experiential learning. Cybersecurity competitions were shown to have the potential to increase the workforce and encourage students to pursue the field of cybersecurity (Conti, Weigand, Skoudis, Cook, &

Arnold, 2014; Ferguson, Tall, & Olsen, 2014; Gavas, Memon, & Britton, 2012; Justice, 2015; Pearce, Zeadally, & Hunt, 2013; Talabis & Martin, 2012).
Hands-on labs were another of the important ways to deliver course content in cybersecurity. Research conducted in 2012 at IUPUI, regarding the Living Lab, shows the benefits of experiential learning (Justice & Do, 2012). Hands-on experience in labs can help with employment and knowledge in work environment (Dinita, Wilson, Winckles, Cirstea, & Jones, 2012; Fanelli & O'connor, 2010; Hoffman, Burley, & Toregas, 2012; Justice & Do, 2012; "The National Initiative for Cybersecurity Education, NICE," 2015; Newhouse,2016). Virtual machines (VM) supported many of the cybersecurity competition and lab operations. Virtual laboratories allowed emulating real cyber threats and rapid generation of multiple scenarios and infrastructures. Creating virtual environments that simulated real world cybersecurity scenarios on the fly can be time consuming and cannot be accomplished in a class period.

Different approaches

Laboratories were accessed by learners in two ways, on-campus, and remote. Approaches to creating on-campus and remote cybersecurity laboratories are discussed.

On-campus

There were two approaches used for cybersecurity lab setups for on-campus students. One approach shown in figure 1 was to install VMs on each workstation the student utilized. This was very simple to do if golden disk images were deployed across the lab. One golden image had the VMs installed and it was pushed out to all the workstations. Additionally, students downloaded VM appliances. In this approach, there was no single point of failure. Student just moved to another workstation, if one machine had issues. The issues to this approach were; 1) if the workstation had limited resources, the student was not able to run multiple VMs; and, 2) it was hard to simulate a real environment with a couple of machines. The second approach was to have a server run multiple target VMs and a student workstation running the attacker VM. Running target VMs on the server allowed students to target machines vulnerable to many different vulnerabilities. There were mainly two problems with this approach. First, the server running the vulnerable VMs was a single point of failure. Second, all the learners had to share the target VMs These problems depended on the resources available to the organization. See Figure 2.

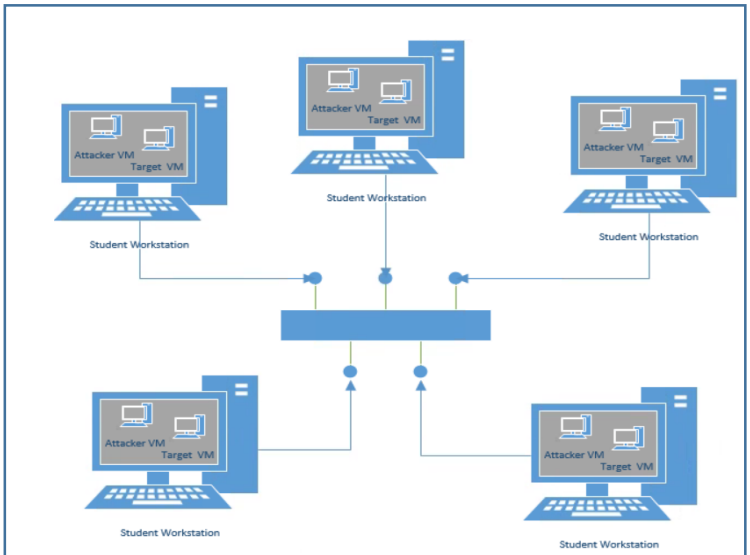

Figure 1: On-campus first lab setup

Remote access

Two approaches were discovered as remote access solutions for cybersecurity laboratories. The first approach was to utilize an attacker server that all the students Secure Shell (SSH) into. The attacker server held all the tools the students would need and updating and managing tools would be in one central area. Due to it being a shared server, resources consumed are medium to low, especially compared to other setups. The configuration is shown in figure 3. The advantages were students familiar with command line wouldn't have too much trouble conducting their exercises and students unfamiliar with command line would be forced to learn it. Disadvantages were not having Graphical User Interface (GUI) tools, lack of GUI may cause issues for some exercises that rely on GUI tools, and having a single point of failure. Having a single SSH server fail can cause issue for all the students, however, it can easily be avoided by having another server. In this approach, the SSH server should be hardened and maintained to protect against attacks from the students.

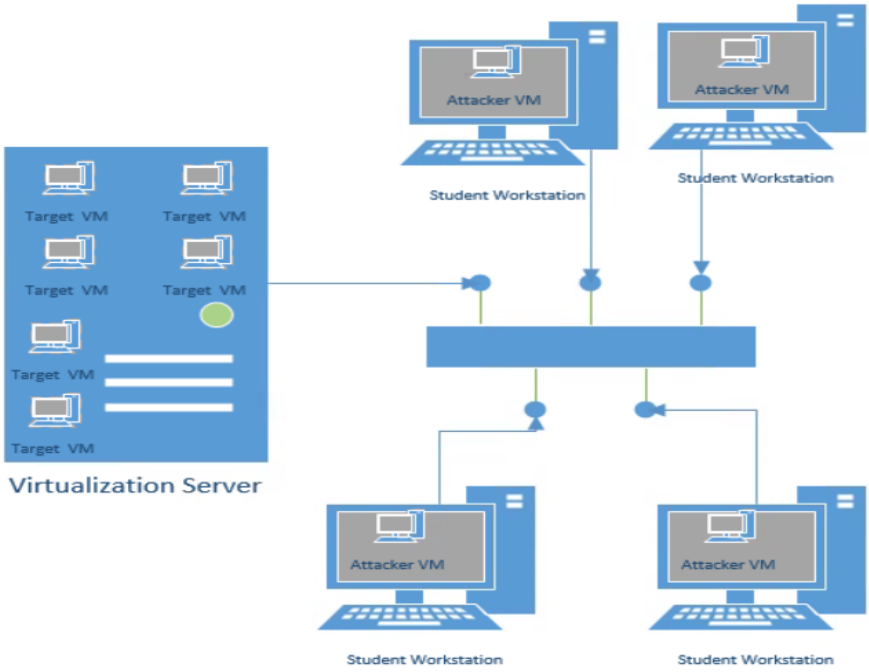

Figure 2: On-campus second lab setup

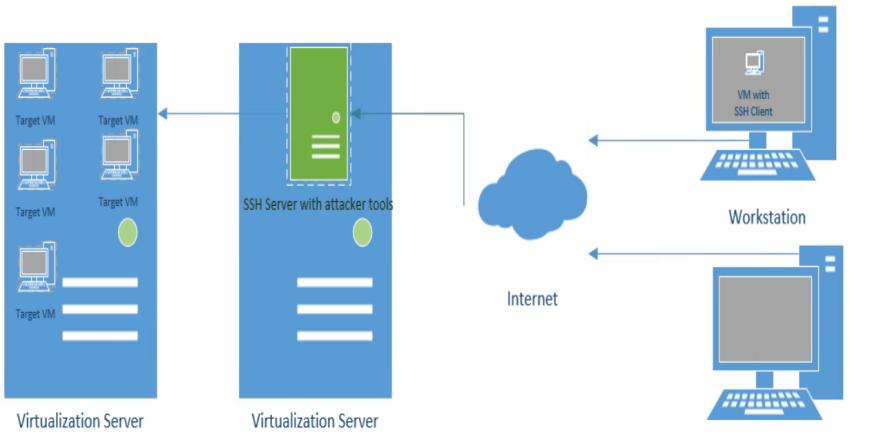

Figure 3: Remote cybersecurity lab using SSH

The second remote approach was Virtual Private Network (VPN) for connection into the attack server. In this case,

the SSH server was replaced with VPN server and the attacker machine ran on the learners' end node computer as you can see in figure 4. The learner VPN'd into the network containing vulnerable VMs. This setup was easy to install and manage. On the organizations side, resources for running attacker tools were not required. The VPN and firewall provided isolation. Just like the SSH server, the VPN server was the single point of failure.

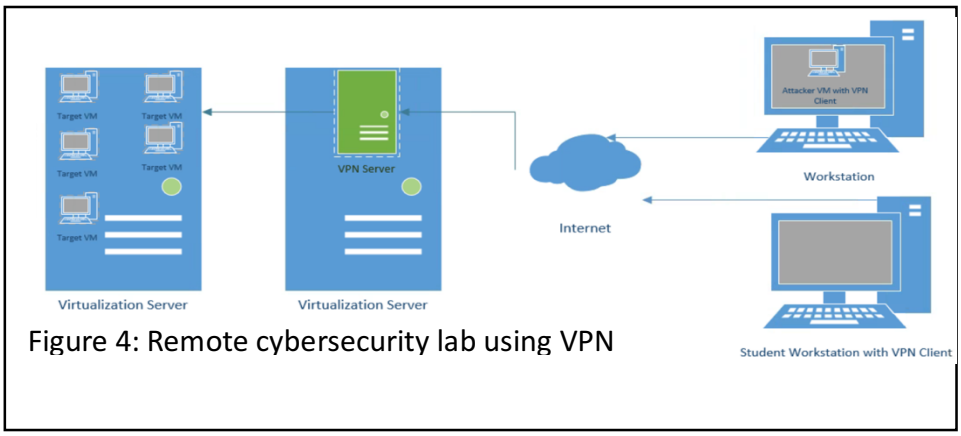

Figure 4: Remote cybersecurity lab using VPN

Problem statement

Creating multiple, diverse and complex virtualized environments for learners to conduct cybersecurity laboratories is desirable for educators. However, it is difficult for educators to create these environments during a class session, much less create them rapidly so the learner can conduct a lab within the time constraints allotted.

Purpose

The focus of research was finding an effective method to deliver cybersecurity labs that could be used on campus or remotely, be spun up rapidly, and be configured by faculty easily so that cybersecurity labs can be conducted within a class session.

Methods

The methods used for rapid RunLabs creation consisted of a JavaScript object notation (JSON) configuration file that created VMs with specified network configuration. In addition, a virtual network computing (VNC) service was created for each VM with a random password, which allowed students access to the VMs individually to work on exercises.

Analysis of lab design approach

The focus of the research was to find an effective method to deliver cybersecurity labs that could be used on campus or remotely, spin them up rapidly, and be configured easily by faculty.
In this section, the design of RunLabs will be discussed.

RunLabs design

RunLabs ran on a hardware platform that was specified for cost effectiveness as well as efficiency. The physical server shown in figures 5 and 6 showed a general-purpose standard server. The

operating system was Ubuntu 14.04 with Quick Emulator (QEMU), an open source machine emulator and virtulalizer (KVM). The QEMU-KVM combination enabled the creation of a virtual environment as exemplified in Figure 6, thus modeling a typical network setting in a work environment. Dnsmasq was used as a Dynamic Host Configuration Protocol (DHCP) and network address translation (NAT) along with Open Switch, an open source virtualized switch designed to manage VM environments (Kelly, 2014). These sub-components along with the Minimega binary, which allows the control of the VMs, will help establish a typical work environment in a cost effective and a portable manner. Each component depicted in figure 6 will be explained in more depth in the Minimega section.

Minimega

Minimega is a tool created at Sandia National Labs that allows for QEMU-KVM VM management. Some of the features Minimega supports are VM creation and management, network creation and management via Open virtual switch (vSwitch), VNC and web user interface (webUI) for VM access, and clustered VM management. Minimega can create, start, and stop VMs. It also lets the user have control over the networking aspects of VMs. For example, the user can put a virtual machine on different virtual local area networks (VLANs) or move them around on different networks. This was done by utilizing Open vSwitch. Minimega can be run with a webUI mode, which allowed the administrator to view the network configuration and VM configuration. The webUI also had HTML5 VNC viewer, which allowed the user to access virtual machines. Finally, Minimega supported clustered mode. In clustered mode, an administrator can manage virtual machines on multiple physical machines. Minimega supported enabling Kernel same-page merging (KSM) feature for QEMU/KVM VM's. KSM simply allows sharing of memory pages for VM's.

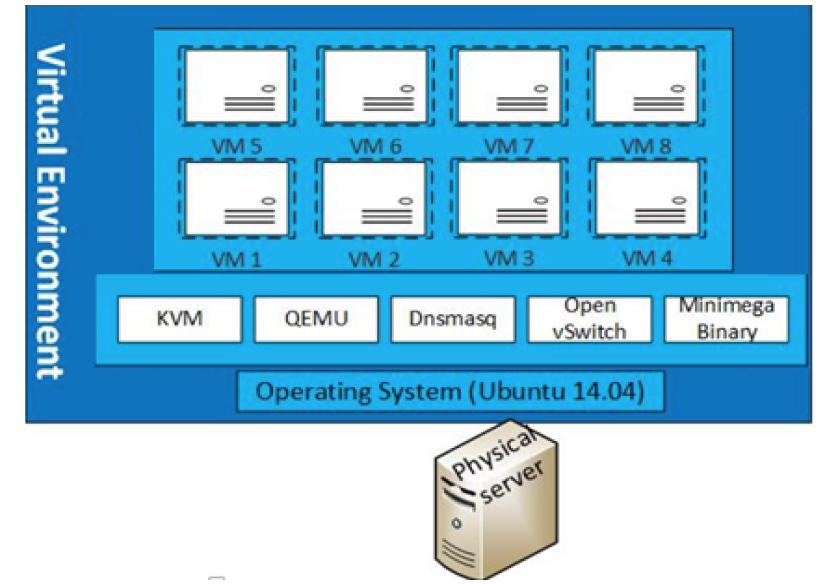

Figure 6: RunLabs virtual environment

Open vSwitch created network interfaces for the VMs that can be monitored from the host. In cases where malware must be monitored, this feature was very useful. Open vSwitch also supports generic routing encapsulation (GRE) tunneling, which can help with scalability of virtualized networks.

RunLabs deployment architecture

RunLabs primary relies on Minimega for virtual machine handling and Python Flask module for creation of virtualized environments. The steps taken to create virtual environment were as follows. A virtual hard drive image was created. QEMU copy on write (QCOW) and ISO (CD specification) were both supported as boot disks. After the virtual hard drive image was created, JSON configuration file was created with networking information and virtual hard drive image information. The JSON configuration file was uploaded via RunLabs webUI. RunLabs parsed the JSON file and created virtual machines accordingly. RunLabs also randomly assigned passwords for VNC interface to the virtual machine. The administrator gained access to the passwords then distributed them to students. The student was then able to then access their VM. Due to the use of Minimega and QEMU-KVM, the resources such as random access memory (RAM) and storage usage was lowered. See figure 7 for the RunLabs deployment architecture.

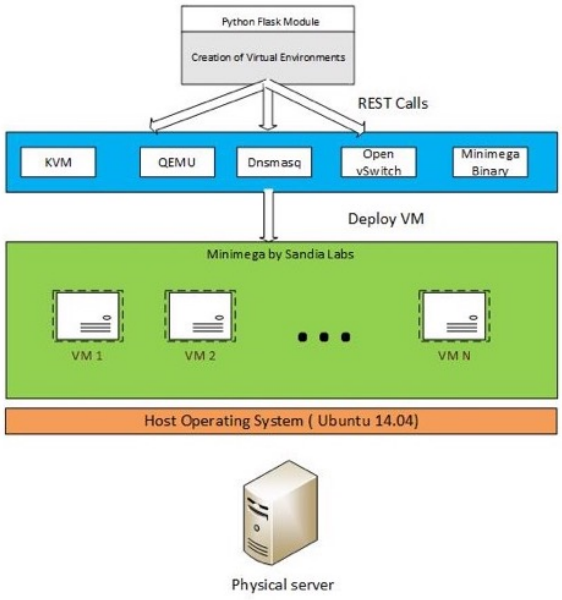

Figure 7: RunLabs deployment architecture

As mentioned before, KSM is utilized to lower the amount of RAM usage and one virtual disk can be used to create multiple virtual machines. Any changes made to the virtual machines by students were not saved to the disk. This was very beneficial because one master image was generated, modified, and used. QCOW format also allowed for linked QCOW files. This meant that there was a base virtual disk and multiple smaller QCOW disks with slight modifications. For example, a Windows 7 installation was saved on 30GB virtual drive and anything installed on top of it can be saved on separate QCOW files linked to the original 30GB virtual drive.

Minimega code was slightly modified to support protecting virtual machines with the use of passwords. RunLabs webUI and API provided the instructor access VNC passwords for each virtual machine, with an option to change the password to something random, in case of compromise. RunLabs webUI also provided the ability to restart the virtual machines quickly, if necessary.

Conclusions

The VNC approach taken with RunLabs had many benefits due to the underlying technologies. VNC access was provided to the students, with password protection. Virtual machines were isolated on virtualized networks created by Open vSwitch. Instead of the instructor generating multiple images for virtual machines, one image could be used to create multiple virtual machines and linked images could be used to create virtual machines with slight configuration differences. Utilizing JSON files for configuring an environment made the job of an administrator easier. JSON configuration files and QCOW were shared amongst multiple RunLabs users to allow for easy virtual environment creation. RunLabs API were utilized to

bridge the RunLabs project with other projects. As discussed previously, the downside of this approach was heavy resource utilization. Since each student receives their own virtual machine, resource use can get high.
Overall, this approach was worth taking for security exercises, cybersecurity competitions, and malware analysis to allow rapid virtual environment creation.

Future work

Future development includes creating a virtual machine repository and bug fixes. Virtual machine repository will allow sharing of virtual machine disk images and RunLabs configuration with predefined challenges. The repository will allow other schools or home users to run their own exercises. Additionally, future work will also include the creation of a noVNC client. A noVNC client uses HTML5, with wss:// encryption.

References

Conti, G., Weigand, M., Skoudis, D. R., Cook, T., & Arnold, T. (2014). Towards a Cyber Leader Course Modeled on Army Ranger School. *Journal Article| Apr, 18*(11), 31am.

Dinita, R., Wilson, G., Winckles, A., Cirstea, M., & Jones, A. (2012). *A cloud-based virtual computing laboratory for teaching computer networks.* Paper presented at the Optimization of Electrical and Electronic Equipment (OPTIM), 2012 13th International Conference on.

Fanelli, R. L., & O'connor, T. J. (2010). *Experiences with practice-focused undergraduate security education.* Paper presented at the Proc. of the 3rd Workshop on Cyber Security Experimentation and Test, Washington, DC.

Ferguson, B., Tall, A., & Olsen, D. (2014). *National Cyber Range Overview.* Paper presented at the Military Communications Conference (MILCOM), 2014 IEEE.

Gavas, E., Memon, N., & Britton, D. (2012). Winning Cybersecurity One Challenge at a Time. *Security & Privacy, IEEE, 10*(4), 75-79.

Hoffman, L., Burley, D., & Toregas, C. (2012). Holistically Building the Cybersecurity Workforce. *Security & Privacy, IEEE, 10*(2), 33-39.

Justice, C. (2015, November 3-5, 2015). *Learner's perceptions of a cybersecurity competition as it relates to knowledge, skills, and abilities (KSA's).* Paper presented at the NICE (National Initiative for Cybersecurity Education) Conference, San Diego, CA.

Justice, C., & Do, L. (2012). *IT experiential learning: The Living Lab.* Paper presented at the Frontiers in Education Conference (FIE), 2012.

Kelly, S. (2014). Dnsmasq. Retrieved from http://www.thekelleys.org.uk/dnsmasq/doc.html

The National Initiative for Cybersecurity Education (NICE). (2015). Retrieved from http://csrc.nist.gov/nice/framework/

Newhouse, B. K.,Scribner, B.,& Witte, G. (2016). NICE Cybersecurity Workforce Framework (NCWF). *Draft NIST Special Publication 800-181*.

Pearce, M., Zeadally, S., & Hunt, R. (2013). Virtualization: Issues, security threats, and solutions. *ACM Computing Surveys (CSUR), 45*(2), 17.

Talabis, M., & Martin, J. (2012). *Information Security Risk Assessment Toolkit: Practical Assessments through Data Collection and Data Analysis*: Newnes.