A Secure Environment to Measure and Manage Cybersecurity Lab Activities

Lee Toderick, Biwu Yang, Te-Shun Chou, and Nicholas Hempenius

East Carolina University
Greenville, North Carolina

Abstract

Implementation and management of a secure environment for online cybersecurity lab experiments is challenging for several reasons. In addition to normal issues associated with hardening a remote lab environment against unauthorized user access and network data transmissions leaking outside the lab environment, the very nature of cybersecurity lab experiments can be inherently dangerous. For example, management network traffic from lab activities may be sniffed using common cybersecurity utilities, the results of lab activity assessment might be hacked and modified, and IT devices used to analyze lab progress may be attacked and compromised. Finally, it is critical for academic integrity that student assessment is secure, assertable, consistent, and repeatable.

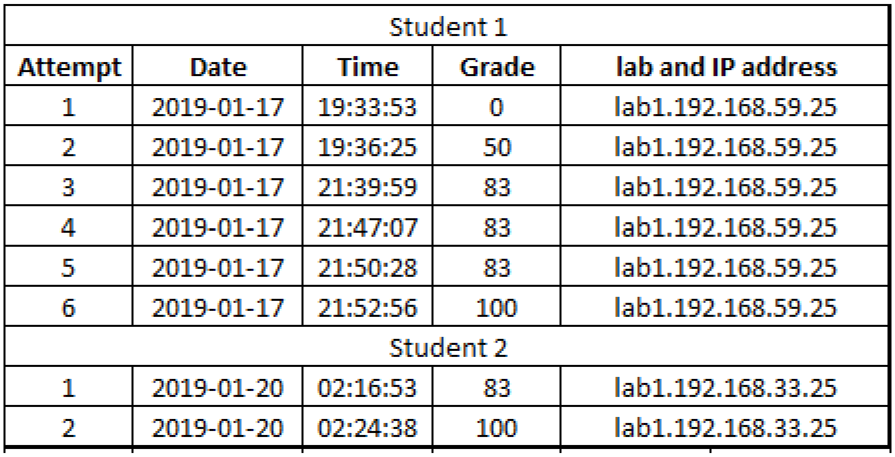
The environment we created provides both educational processes and game based, hands-on exercises, where students can learn the concepts and best practices in cybersecurity defense as well as hands-on skills learning. An initial assessment was taken at a recent train-the-trainer workshop, and several lab learning and effectiveness metrics are provided.

Keywords: Cybersecurity, remote labs, cybersecurity labs, cybersecurity gamification

1. Introduction

The Information and Computer Technology Program, within the Department of Technology Systems, College of Engineering and Technology, East Carolina University (ECU), has successfully integrated secure, remote lab experiments in curriculum courses for more than 14 years [1, 2, 3]. An enhancement to remote labs has been the addition of automated grading lab experiments, which have been employed in Undergraduate curriculum courses for more than 6 years [4]. This combined environment provides students with the ability to work outside the physical University classrooms and labs, at their own pace. Also, in traditional lab grading where student lab assignments are returned typically within one week, the automated grading environment allows the student to receive the results of the lab submission immediately and take corrective action if needed. For example, the following table represents the results of two students who worked separately and completed the same lab experiment, but with a different number of attempts, on different days, and at different times. Student 1 completed the lab on a Thursday evening. Student 2 completed the lab on a Sunday, at 2:16 AM. There are numerous log entries of labs performed throughout the 24-hour period, all performed at the convenience of the student. This particular course supported over 98 active students and has over 100 auto-graded labs.

Table 1. Student performance on lab 1 within the remote lab with automated grading



Student 1				
Attempt	**Date**	**Time**	**Grade**	**lab and IP address**
1	2019-01-17	19:33:53	0	lab1.192.168.59.25
2	2019-01-17	19:36:25	50	lab1.192.168.59.25
3	2019-01-17	21:39:59	83	lab1.192.168.59.25
4	2019-01-17	21:47:07	83	lab1.192.168.59.25
5	2019-01-17	21:50:28	83	lab1.192.168.59.25
6	2019-01-17	21:52:56	100	lab1.192.168.59.25
Student 2				
1	2019-01-20	02:16:53	83	lab1.192.168.33.25
2	2019-01-20	02:24:38	100	lab1.192.168.33.25

The environment for cybersecurity labs is similar to our existing infrastructure that consists of a secure, isolated lab environment with remote access and automated assessment. However, enhancements are needed for additional internal security. For example, a separate, hardened, management network and secure data traffic are added to ensure integrity of the lab results, along with other network security best practices. We call this environment Competitive-Labs-as-a-Service, or CLaaS [5]. In this paper we explore our secure cybersecurity lab environment that provides both educational processes and game based, hands-on exercises where students can learn the concepts and best practices in cybersecurity defense as well as hands-on skills.

In future papers, we will present our research gathered from an upcoming campus workshop where external, visiting community college faculty will have the opportunity to use the system and present feedback about lab experiences, measured success of lab learning objectives, and lab security.

This paper is organized as follows: Section 1 is Introduction. Section 2 discusses related work. Section 3 explores our methodology. In Section 4, we summarize the current work and explore continuing work.

2. Related Work

Other universities and public corporations have created their own cybersecurity laboratories. Some cybersecurity laboratories and centers are research oriented. The Argus Group, in University of South Florida, was founded by Dr. Xinming Ou in 2006. The mission of the group is to “carry out cyber security research with the focus on the defense aspect of the cyber space.” Over the years, many research papers were published [6]. In 2008, a cybersecurity lab was established at the University of Louisville with the aim “to become a premier research center in computer and network security, biometrics, and forensics.” Over the years, faculty members and students published many research papers [7]. University of Maryland, Baltimore County (UMBC) created an Accelerated Cognitive Cybersecurity Laboratory for research in the area of cybersecurity. Many faculty and students are involved in the research efforts using the facility [8].

Similar efforts have been made to provide educational experiences for students who are interested in cybersecurity. The Cybersecurity Engineering Laboratory (CybEL) was established in Embry-Riddle Aeronautical University to support teaching and research in the area of cybersecurity. The lab is equipped with a virtualization infrastructure and uses visualization techniques to analyze attack related data and display information of ongoing attacks. CybEL is constructed within an isolated network environment. This lab primarily supports courses in their M.S. in Cybersecurity program [9]. Montgomery College opened its state-of-the art Cybersecurity Lab at its Germantown campus in 2015. Students learn and practice in the field of cybersecurity in the facility. One important aspect of this facility is that “The lab’s infrastructure runs on a separate network so that students can engage in real-world security exercises without affecting other technology operations at the [c]ollege” [10].

Public Broadcasting Service provides Cybersecurity Lab to the public as a part of its NOVA Labs series. The Cybersecurity Lab is an online game designed to teach game players how to defend against cyber-attacks from multiple aspects [11].

To help spread the effective use of cybersecurity labs, there have been several efforts to provide guidelines and instructions on building cybersecurity labs. Vitaly Ford in the National Cybersecurity Student Association posted instructions and resources (VM images and various tools) for building cybersecurity excises [12]. Alexandru Bardas and Xinming Ou at University of South Florida published a paper about setting up a cyber security lab for educational purpose in 2013 [13].

3. Methodology

The CLaaS project utilizes many technologies to provide an online lab environment for students to learn and practice cybersecurity best practices effectively. As part of the applied research project, many internet based protocols and technologies are evaluated for appropriateness for the implementation of the project. Technologies and tools are carefully selected to meet the project requirements and online lab environment implementation.

3.1. Information Security Protocols and Technologies

Information security protocols and technologies are essential to the CLaaS project. Not only are these protocols and technologies the core contents for students to learn and practice, they are used to make a secure environment as well. Protocols and technologies such as data encryption, access control, logging, and analysis are implemented for the online environment.

3.1.1. Protocols and Technology for Remote Access

Remote access, in its narrow definition, is the communication link from a remote host to access another host in a far distance. However, in this project, all lab activities involve accessing VMs; the remote access technologies are also used.

Guacamole – an open source, clientless remote desktop gateway that provides access for standard connection protocols like VNC, RDP, and SSH. Guacamole acts essentially as a reverse proxy for remote desktop sessions allowing clients to access remote systems within a browser. Written in JavaScript and using standards such as HTML5, Guacamole requires almost no configuration on the client side and minimal configuration on the server side. TLS must be configured on Guacamole to make the connection secure.

Secure Shell (SSH) – an open source, application layer protocol that provides secure access to a host remotely over a network. Using encryption, SSH provides a secure channel over an unsecure network. SSH uses the client-server model, where an SSH client uses the SSH protocol to connect to an SSH server that resides in the destination host. A user can perform any operations that are allowed on the destination host.

Hypertext Transfer Protocol (HTTP) – an open source, application layer protocol that manages data transmission on the World Wide Web. HTTP uses the client-server model in web data communications. A web client uses HTTP to request data from a web server and the web server uses HTTP to manage the delivery of web content to the client. HTTP does not encrypt the data in transit.

Hypertext Transfer Protocol Secure (HTTPS) – an open source, extension of HTTP. HTTPS encrypts data communication of HTTP. The encryption is through the Secure Sockets Layer (SSL) protocol. Nowadays the Transport Layer Security (TLS) protocol replaces SSL.

Firewall – a service that filters incoming network traffic and make a decision to allow or deny the network traffic based on predefined criteria. A firewall can be categorized as host based or network based. A host-based firewall is a service running on a host that protects the host by filtering incoming and outgoing data packets. A network based firewall is a service, typically by a firewall appliance, situated in the network path to protect the network infrastructure.

FreeBSD PFSense- an open source product that is deployed as either a stand-alone physical device or as a virtual machine (VM) appliance. PFSense has multiple uses, but is mostly employed as a stateful/stateless firewall and router, VPN manager, and with add-ons as an intrusion detection/prevention system. It is managed with a robust web interface.

3.1.2. Protocols and Technology for Secure Internal Data

Security considerations must be made in the design and implementation of a cybersecurity learning and practice environment. Examples of security consideration include secure data transmission (data in transit) and secure data storage (data at rest).

File Transfer Protocol (FTP) – an open source, application layer protocol that provides file transmission between hosts in a client-server model. An FTP client uses the **put** method to upload a file to the FTP server and the **get** method to download a file from the FTP server. The data transmission with FTP is not encrypted. If the FTP server is setup to user authentication, an attacker stationed in the data transmission path can capture the packets and user credentials.

FTPS – an open source, set of security extensions added to the FTP protocol to protect FTP data transmission using SSL encryption. FTPS can use FTPS Implicit SSL and FTPS Explicit SSL. FTPS Implicit SSL is now considered obsolete.

SFTP – an open source, file transfer protocol based on SSH encryption. SFTP uses a single channel (default port 22) to transmit both control commands and data packets.

SCP (Secure Copy) – an open source, simple application layer protocol that provides secure file transfer based on the SSH protocol. SCP is mostly used in UNIX/Linux hosts.

Rsyslog and Rsyslog w/TLS-SSL- an open source, mature, extremely flexible system for log processing. The client-server model supports both local and remote log services. Rsyslog used only the UDP transport method, but now supports TCP. Although clear-text, Rsyslog easily supports TLS-SSL secure log encryption and is deployable on Linux and Windows systems.

Security Enhanced Linux (SELinux) – an open source, security enhancement to Linux that provides mechanism for managing access control security policies in Linux.

DMZ network – a network infrastructure design in which a small network is isolated from the main internal corporate backbone typically via a firewall. The servers on this network are facing the outside network such as the internet and they provide information to the public network. Examples of these servers include web server and email server. Firewall policies can determine how corporate internal network and public network access the DMZ network.

3.2. Lab Environment

This section describes the lab environment, including components, topology, and protocols.

3.2.1 Lab Components and Topology

The Student Lab Environment resides within the college data center and is completely isolated from outside physical access through normal access restrictions associated with data center security policy. Logical security is maintained through a Guacamole tunnel that connects the remote student computer to the Student Lab Environment. This connection ensures malicious traffic from the student computer cannot enter the Student Lab Environment, and malicious traffic from the Student Lab Environment cannot enter the student computer. The Student Lab Environment resides in a complete, self-sustaining, cybersecurity ecosystem.

Our lab environment employs the fault-tolerant VMWare ESXi 6.7 virtualization hypervisor, resides on a 1U rack mount server with 512GB RAM, and serviced by tiered, fault-tolerant, secondary storage. Students access the virtualized environment through secure connections with Guacamole remote desktop gateway. Guacamole runs on a Linux or Windows computer, and uses HTML 5 and a clientless web browser. Guacamole supports VNC, RDP and SSH remote connection protocols [14].

Figure 1 depicts an overview of the cybersecurity lab environment. The lab environment accommodates 10 student participants, each with his or her own pod that consists of a Linux host Pod VM, one Attacker VM, and 9 Defender VMs, one for each cybersecurity lab experiment. The student pod is scrubbed to a default state before performing each lab. A secure management environment contains devices used to monitor and display student progress.

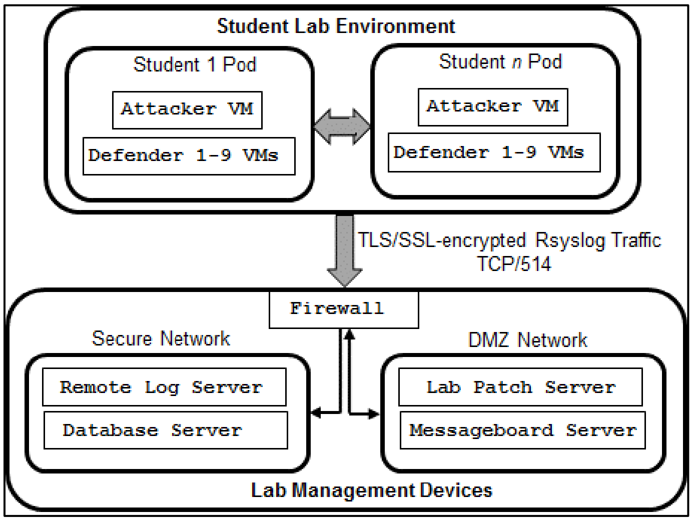


Figure 1. Cybersecurity lab environment

The cybersecurity lab employs one network for student lab activities and another dedicated network for measuring and assessing lab exercise objectives. Because of academic integrity concerns, students only have access to lab experiment devices but not devices used for measuring and assessing lab results. Student network connectivity is between each student pod and DMZ Network. Within the DMZ, the lab patch server supplies application updates to student devices through YUM and HTTP, and messageboard server requests via HTTPS. Data that tracks student activities is encrypted end-to-end using TLS/SSL to a remote log server. The log server is protected by a hardened firewall with minimal open ports and access control lists that pass only specified traffic to specific management devices. Log server data is normalized and securely passed to a database server that stores and supplies student data to a messageboard server for display to students through a web browser using TLS/SSL. Devices in the management network are hardened through SELinux and Linux permissions. No management device login from the Student Lab Environment is permitted.

3.2.2. Student Lab Environment and Assessment Measurement

Refer to Figure 2. Each student pod consists of a Linux VM with sufficient resources to deploy Attacker VM and Defender VMs. The pod host is a Linux Cent OS 7 host. Attacker VM is a Kali Linux host. Depending on the lab, the Defender VM could be either a Linux host or Windows Server. The student lab network IP address range is 10.10.x.0/16. This is the network where student activity occurs. A secure network, 172.20.X.0/24, carries management traffic to the secure management device environment.

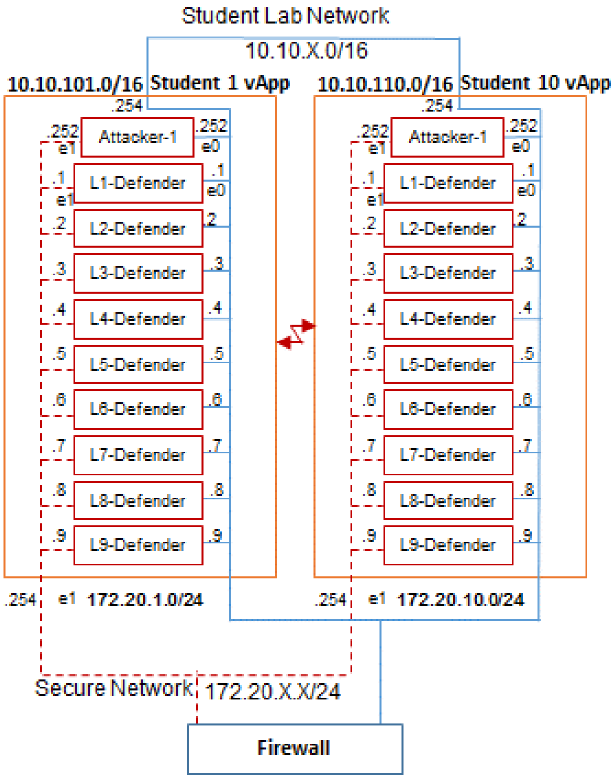
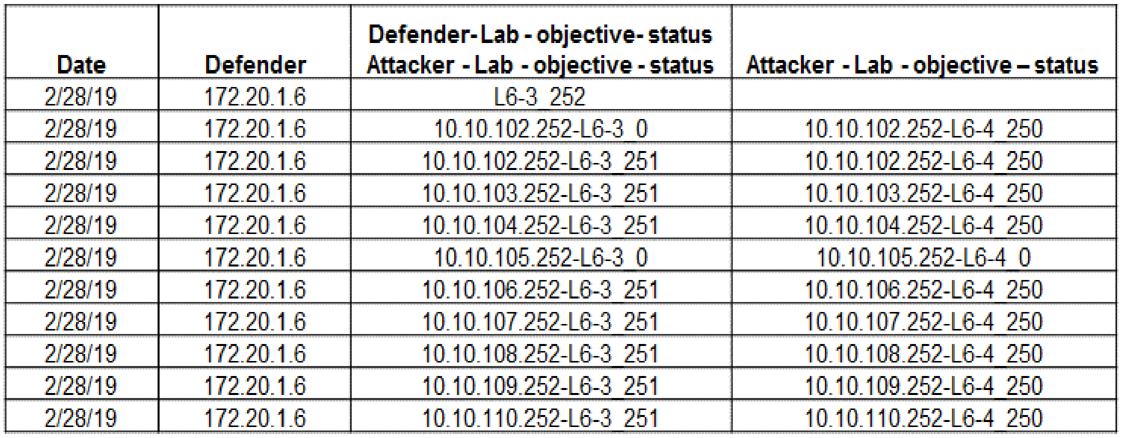


Figure 2. Student lab network

The sequence for students to complete a lab employs a three-stage learning process. First, the student is presented with a lab tutorial that explains the concepts of the lab exercise, followed by a quiz. The quiz can be taken multiple times. Only after the student achieves a passing score of 80 points is the hands-on exercise component available. The second phase of the lab exercise is configuration. During this phase, the student must complete the initial configuration on the Defender virtual machine (VM) within a specific period of time. For example, the student must determine which service on the Defender VM is vulnerable to compromise and update the service with a patch. The student uses the Attacker VM to verify the patch was installed correctly. At the end of the configuration phase, the student uses the Attacker VM against other students' Defender VMs. If the foreign attacker was able to defeat the defender, the attacker would gain points and the defender lose points.

When the lab exercise is active, a management script on each Defender VM monitors the status of all lab specifications. Refer to Table 2 for an example of decrypted log output received by the remote log server from Lab 1 Defender. Every few seconds a log entry is automatically crafted by a resident script on Defender that includes a specification and corresponding status. 0 is used for a specification completed successfully, and a non-zero integer for a specification not completed correctly. For example, in Lab 6, specification three requires the student to patch a vulnerable service. A crafted log entry indicating Defender 1 failure to apply a patch, code 252, would look similar to defender1:lab6:step3-252. If an attacker identified the vulnerable service, such as Attacker 2, the log entry might look similar to defender1:attacker2:lab6:step3-0. Or, if the attacker failed to identify the vulnerable service, the log entry might look similar to defender1:attacker3:lab6:step3-251. If the attacker exploited the vulnerability, the log entry might look similar to defender1:attacker5:lab6:step4-0. If the attacker did not exploit the vulnerability, a log entry might look similar to defender1:attacker2:lab6:step4-250. A log entry for a successful attack from Attacker5 would look similar to defender1:attacker5:lab1:step4-0. Each Defender host sends remote log entries in similar format. Each log entry is routed to the remote log server, where it is processed for a sanity check and evaluated.

Table 2. Decrypted log traffic



Date	Defender	Defender-Lab - objective- status Attacker - Lab - objective - status	Attacker - Lab - objective – status
2/28/19	172.20.1.6	L6-3_252	
2/28/19	172.20.1.6	10.10.102.252-L6-3_0	10.10.102.252-L6-4_250
2/28/19	172.20.1.6	10.10.102.252-L6-3_251	10.10.102.252-L6-4_250
2/28/19	172.20.1.6	10.10.103.252-L6-3_251	10.10.103.252-L6-4_250
2/28/19	172.20.1.6	10.10.104.252-L6-3_251	10.10.104.252-L6-4_250
2/28/19	172.20.1.6	10.10.105.252-L6-3_0	10.10.105.252-L6-4_0
2/28/19	172.20.1.6	10.10.106.252-L6-3_251	10.10.106.252-L6-4_250
2/28/19	172.20.1.6	10.10.107.252-L6-3_251	10.10.107.252-L6-4_250
2/28/19	172.20.1.6	10.10.108.252-L6-3_251	10.10.108.252-L6-4_250
2/28/19	172.20.1.6	10.10.109.252-L6-3_251	10.10.109.252-L6-4_250
2/28/19	172.20.1.6	10.10.110.252-L6-3_251	10.10.110.252-L6-4_250

3.2.3. Lab Management Devices

Firewall is implemented with a FreeBSD PFSense VM that is used to isolate the student lab network from the management devices, but provide connectivity to the DMZ network. It is configured to permit TCP port 514, the well-known syslog TCP port number, from the secure network, and allow student network HTTP and HTTPS traffic to the DMZ network. Within the management devices, Firewall also filters internal traffic between the database server and the messageboard server. No traffic is permitted between the secure network and student lab network.

Remote log server is a hardened Linux VM that only receives remote log message from the management script on student Defender VMs. Terse log messages are processed and passed to the database server VM.

The database server is a Linux VM that hosts a MongoDB database. The database stores parsed log entries from the remote log server, and provides the messageboard server with student activity information such as lab concepts, lab quiz questions, lab quiz scores, and authorized student user accounts. It is also responsible for maintaining scores and messages that are displayed by the messageboard server.

Within the DMZ Network, the lab patch server is a Linux VM which provides software patches and utilities required to complete labs. The messageboard server is a Linux VM that provides lab activity data and status messages and scoring it receives from the database server to each student pod computer.

3.3. Securing the Academic Environment

The most critical component of any remote lab environment is the data center that houses the computers and networking equipment for connectivity with the outside world [16]. Our college data center employs the latest building security features that include a lights-out, windowless environment, raised floor, restricted access with logging to the campus police, single ingress-egress, and compliance with federal and state regulations regarding fire codes and safety.

University in-band computer networking is used to access our physical lab environment. The university allows no packet to flow through the network unless it has been authorized, and normally has been identified with a registered user. Network bandwidth has been analyzed and tested, and is more than adequate and reliable for the number of students who will simultaneously use the lab environment.

3.4. Securing Data in Transit

Several data encryption protocols were considered for transferring data from the Defender VMs to the remote log server. Our focus was on a solution that permits secure data transfer without userid authentication. FTP was dismissed because of manual, clear text, authentication, or anonymous FTP upload security issues. FTPS [17] secures the FTP control channel and optionally the FTP data channel using TLS, but manual authentication to the remote log server was a concern. Issues similar to FTP apply to anonymous FTPS upload. SSH [18] and SFTP, a subsystem of SSH, were considered with key authentication but were eventually dismissed because of the need for continuous authentication.

Finally, we investigated and implemented rsyslog [19, 20]. Rsyslog is a client-server service for log management. Later versions of rsyslog support log transfer over well-known TCP port 514. We employed rsyslog with TLS encryption [21] for several reasons: (1) log data is small and compact, and the source of the log entry is easily identifiable to the log server; (2) no manual authentication is required to upload a log entry; instead, the log entry is crafted by a script and is passed to the rsyslog server; and (3) the communication is secured for confidentiality and integrity through TLS. With the excellent documentation from the rsyslog site, a Windows Server rsyslog client was also configured [22, 23, 24].

The database server receives processed data from the remote log server, and passes it to the messageboard server using TLS.

3.5 Securing Data at Rest

The Remote Log Server rsyslog server receives rsyslog client data from each Defender VM in the Student Lab Environment. Log files on the remote log server are restricted from normal access and viewing through Linux permissions. On remote log server, permissions on the log directory are 700, and 600 for log files. Both the log server remote and database server execute with SELinux set to “enforcing” [25]. SELinux enforces mandatory access control policies on services and the system, and enhances traditional Linux permissions. SELinux “enforces the separation of information based on confidentiality and integrity…" [25]. Parsed log entries are passed to the database server directly for storage.

Both the remote log server and database server have one admin user and a single normal user. During production lab operation, admin commands on the hosts are executed by the normal user through sudo. This provides the added security of logging each admin command to file /var/log/secure.

3.6. Securing Displayed Data

On the DMZ network, the messageboard server communicates with each student's web client on the Pod host computer through a secure TLS connection, HTTPS, and displays data that is sent via TLS from the database server. Each student is able to view the real-time status of their lab activities. In addition to the management devices firewall, there is a firewall on the messageboard server that further restricts access.

3.7. Workshop Survey Results

A one and one-half day, train-the-trainer workshop, was held at ECU in July, 2019. There were 17 faculty from area colleges and universities who attended the workshop and completed an exit survey. The faculty teach a variety of InfoSec courses. Following are several preliminary survey metrics and responses. Additional survey analysis will be shared in future papers.

Table 3. Workshop exit survey results

Metric	**Response Percentage**		
	Strongly Agree or Agree	Neutral	Strongly Disagree or Disagree
The performance of the lab environment was satisfactory.	76	24	
The performance of the virtual machines in the labs was satisfactory.	82	12	6
The simulated lab environment felt realistic, like using real world servers.	100	0	0
I would recommend using this lab environment to other faculty.	100	0	0
The lab environment is user-friendly and felt reliable, secure, and easy to access and operate.	88	12	0
The labs were relevant to current cybersecurity technology and methods.	100	0	0
The introductions were helpful in understanding the theoretical knowledge of cybersecurity topics.	96	6	0
The quizzes were helpful reinforcing information learned from the introduction.	76	24	0
The lab walkthroughs were helpful in getting practical experiences of cybersecurity.	100	0	0

The three-stage learning (introduction, quiz, and lab walkthrough) is a good idea for teaching cybersecurity.	88	12	0
I encountered no difficulties when doing the labs.	65	18	18
The score and Message Board made learning interesting in an interactive environment.	94	6	0

4. Summary and Continuing Work

The purpose of our CLaaS environment is to provide cybersecurity education and skills practice to post-secondary school student and faculty teams, using a secure, isolated, virtualized environment that is accessible from any location with an internet connection. The competitive, game based, hands-on nature of the lab experiments for both the defender configuration steps and attacker compromise steps emphasize to students the effect of vulnerable and compromised computers in real-world situations. Students only access lab experiment devices. Secured devices and a second, secure network, are used to measure and assess student learning. Data that tracks student activities is encrypted end-to-end using TLS/SSL to a remote log server, which is protected by a hardened firewall. Management devices process and store student progress information, and a separate Messageboard displays progress metrics to students through a web browser using TLS/SSL. Devices in the management network are hardened through SELinux and restricted Linux file permissions.

Future ClaaS lab work will continue workshops virtually, and make the software lab environment available to other Colleges and Universities throughout the United States.

Acknowledgments

This research is made possible by the National Science Foundation under grant 1723650. The authors are grateful to the support of the Department of Technology Systems in the College of Engineering and Technology at ECU.

References

[1] L. Toderick, TJ Mohammed, and M. Tabrizi, A consortium of secure remote access labs for information technology education. *Proceedings of the 6th Conference on Information Technology Education*, SIGITE, 2005, Newark, NJ, USA, October 20-22, pp. 295-299. 2005.

[2] P. Li, L.W. Toderick, P.J. Lunsford, Experiencing Virtual Computing Lab in Information Technology Education. In: *Proceedings of the 10th ACM Conference on SIG-Information Technology Education*, SIGITE 2009, Fairfax, Virginia, USA, October 22–24, pp. 55–59. 2009.

[3] P. Li, L.W. Toderick, P. Lunsford, Experiencing virtual computing lab in information technology education. *Proceedings of the 10th ACM Conference on SIG-Information Technology Education*. ACM, pp. 55-59. 2009.

[4] P. Li, L. Toderick, *An Automatic Grading and Feedback System for E-Learning in Information Technology Education* Paper presented at 2015 ASEE Annual Conference & Exposition, Seattle, Washington. 10.18260/p.23518. 2015.

[5] N. Hempenius, Te-S. Chou, and L. Toderick, Cybersecurity Competitive Labs-as-a-Service: Automated Score and Message Board Design. In *Proceedings of the 19th Annual SIG Conference on Information Technology Education* (SIGITE '18). ACM, New York, NY, USA, 148-148. https://doi.org/10.1145/3241815.3241825. 2018.

[6] Argus Cyber Security Lab – University of South Florida, Argus Cyber Security Lab. [Online]. Available: http://www.arguslab.org/. [Accessed January 28, 2019].

[7] University of Louisville, CyberSecurity Lab. [Online]. Available: http://cecs.louisville.edu/security/. [Accessed January 28, 2019].

[8] University of Maryland, Baltimore County (UMBC), Accelerated Cognitive Cybersecurity Laboratory. [Online]. Available: https://accl.umbc.edu/. [Accessed January 28, 2019].

[9] Embry-Riddle Aeronautical University, Cybersecurity Engineering Laboratory. [Online]. Available: https://daytonabeach.erau.edu/about/labs/cybersecurity-engineering-lab/. [Accessed January 28, 2019].

[10] Montgomery College, State-of-the-Art Cybersecurity Lab Opens on Germantown Campus. [Online]. Available: http://mcblogs.montgomerycollege.edu/atmc/state-of-the-art-cybersecurity-lab-opens-on-germantown-campus/. [Accessed January 28, 2019].

[11] Public Broadcasting Service, NOVA Cybersecurity Lab. [Online]. Available: https://scistarter.com/project/1087-NOVA-Cybersecurity-Lab. [Accessed January 28, 2019].

[12] Vitaly Ford, National Cybersecurity Student Association, Build Your Own Lab. [Online]. Available: https://www.cyberstudents.org/blog-post/build-your-own-lab/. [Accessed January 28, 2019].

[13] Bardas, A. & Ou, X., (2013), *Setting Up and Using a Cyber Security Lab for Education Purposes*, Journal of Computing Sciences in Colleges Vol. 28, Issue 5, May 2013, pp 192-294.

[14] Apache Guacamole. [Online]. Available: https://guacamole.apache.org. [Accessed January 22, 2019].

[15] The Four Layers of Data Center Physical Security for a Comprehensive and Integrated Approach. [Online]. Available: https://www.anixter.com/content/dam/Anixter/White%20Papers/12F0010X00-Four-Layers-Data-Center-Security-WP-EN-US.pdf. [Accessed 22 January, 2019].

[16] Data Center Physical Security. [Online]. Available: https://resources.infosecinstitute.com/category/certifications-training/cissp/domains/security-engineering/data-center-physical-security/#gref. [Accessed January 22, 2019].

[17] Securing FTP with TLS. [Online]. Available https://tools.ietf.org/html/rfc4217. [Accessed January 24, 2019].

[18] The Secure Shell (SSH) Transport Layer Protocol. [Online]. Available: https://www.ietf.org/rfc/rfc4253.txt. [Accessed January 24, 2019].

[19] The BSD syslog protocol. [Online]. Available: https://tools.ietf.org/html/rfc3164. [Accessed January 24, 2019].

[20] RSYSLOG is the rocket-fast system for log processing. [Online]. Available: https://www.rsyslog.com/. [Accessed January 24, 2019].

[21] Encrypting Rsyslog Traffic with TLS (SSL). [Online]. Available: https://www.rsyslog.com/doc/v8-stable/tutorials/tls_cert_summary.html. [Accessed January 24, 2019].

[22] Using Rsyslog Windows Agent to forward log files. [Online]. Available: https://www.rsyslog.com/using-rsyslog-windows-agent-to-forward-log-files/. [Accessed 28 Jan, 2019].

[23] Encrypting Syslog Traffic with TLS (SSL). [Online]. Available: https://www.rsyslog.com/doc/tls_cert_client.html. [Accessed 28 Jan, 2019].

[24] NXLog TLS Configuration. [Online]. Available: https://www.loggly.com/docs/nxlog-tls-configuration/. [Accessed 28 Jan, 2019].

[25] Security-Enhance Linux. [Online]. Available: https://www.nsa.gov/what-we-do/research/selinux/. [Accessed January 24].

[26] SElinux Frequently Asked Questions (FAQs). [Online]. Available: https://www.nsa.gov/What-We-Do/Research/SELinux/FAQs/#I1. [Accessed January 24,].

Biographical Information

LEE TODERICK is a full-time teaching instructor within the Department of Technology Systems, College of Engineering and Technology, ECU, since 2001. He earned a BS in Computer Science from ECU (1988), and an MS in Computer Information Systems from Boston University (1994). His teaching course load includes Data Storage Management, Cloud Services, Linux Advanced System and Services Administration, and Information Assurance Technologies. He has authored numerous lab experiments in support of his courses, created several secure lab environments and is a copyrighted author for BroadReach Extended (BRE), a secure, student-centric automated grading system.

BIWU YANG is a professor in the Department of Technology Systems at ECU. He received his PhD from the University of Rhode Island in 1991 and taught at Eastern Kentucky University 1991 through 1995. In August 1995 he joined ECU. From 2005, he served in the Office of Academic Outreach as the coordinator of Platform Research and Development. The office has since been changed to the Office of Academic Initiatives where the focus has been providing virtual exchange opportunities and experiences to students at ECU and global partner institutions worldwide. He is responsible for coordinating technology research, implementation, and maintenance to support various virtual exchange activities. Dr. Yang's research interests include computer networking, information systems, and information security.

TE-SHUN CHOU is an associate professor in the Department of Technology Systems at ECU. He received his bachelor's degree in Electronics Engineering at Feng Chia University and both master's and doctoral degrees in Electrical Engineering at Florida International University. He serves as the program coordinator of the master's program in Network Technology for the Department of Technology Systems and the lead faculty of Digital Communication Systems concentration for the Consortium Universities of the PhD in Technology Management. He is also the point of contact of ECU National Centers of Academic Excellence in Cyber Defense Education (CAE-CDE). Dr. Chou teaches IT related courses, which include network security, network intrusion detection and prevention, wireless communications, and network management. His research interests include machine learning, wireless communications, technology education, and information security, especially in the field of intrusion detection and incident response.

NICHOLAS HEMPENIUS is a graduate student in the Network Technology Information Security Concentration master's degree program at ECU. He received his bachelor's degree in Information and Computer Technology with a Concentration in Information Security at ECU. He is a member of the ECU Colligate Cyber Defense Competition team and works as a graduate research assistant.