Exploring the CELF — An Innovative Cyber Engineering Learning Facility

Jason M. Pittman
Department of Computer Science
High Point University
High Point, North Carolina, USA
jpittman@highpoint.edu

***Abstract*—This innovative practice work-in-progress study presents the technical architecture and pedagogical design from a new type of laboratory environment intended to resolve an experiential disconnect between students and educators or instructional designers. The cybersecurity laboratory exercise is a common feature in computing education. Researchers and instructional designers suggest these laboratory exercises are constructivist by nature and impart an active learning experience. However, students report an overtly objectivist perspective when asked about how they use the same laboratories. More specifically, whereas traditional laboratory exercise designs are *theme parks* in game design terms, the Cyber Engineering Learning Facility (CELF) is designed and implemented as a sandbox. To that end, the CELF uses the pedagogical concept of a *phenomenarium* and first principles of cyber science to engender a full stack Bloom's taxonomy experience. Preliminary results suggest the CELF design and implementation is operating as intended but with room for additional innovation. To that effect, we present analysis of the infrastructure utilization as well as analysis of student perceptions. Further, we discuss limitations of the current CELF design along with several avenues for future innovation.**

Keywords—Cybersecurity; Hands-On Laboratory; Private Cloud; Learning

I. INTRODUCTION

The cybersecurity laboratory exercise is a mainstay in computing education. Existing cybersecurity laboratories [1] aim towards providing realistic cyber subjects, objects, and actions [2]. Researchers and instructional designers assert such laboratory exercises are constructivist in design and implementation [3][4][5]. Typically, students engage such exercises through a mapped-out procedure (e.g. lab exercise). The intended result is a moderated or curated experience mimicking workplace activity. However, a problem exists whereby cybersecurity learning environments are perceived to be objectivist by students [6][7] and are designed as theme parks. More specifically, students view the lab procedure as an expeditious tool they can use to accelerate towards their incentivized goal - the grade. Thus, the procedure or exercise becomes a means to bypass constructivist pedagogy and learning is, at best, in the lower half of Bloom's taxonomy.

Accordingly, this innovative practice study presents a work-in-progress view of the technical architecture and pedagogical design for a new type of laboratory environment intended to resolve the experiential disconnect. To properly situate how the Cyber Engineering Learning Facility (CELF) diverges from traditional architecture and pedagogy, we first carefully examined the advantages and disadvantages of cybersecurity laboratories as evidenced in related work.

II. RELATED WORK

Cybersecurity laboratories (labs) have a rich foundation in the literature [8]. From the beginning [9], the premise of incorporating labs into coursework has been to provide hands-on experience. How a given lab purports to generate such experience is the difference between lab implementations. That said, broadly speaking, there are technical and pedagogical overlaps. Based on such overlap, the related work is divisible into three eras: hardware, virtualization, and post-virtualization. We leveraged the overlap between these eras, along with a synthesized focus for each, to inform our design and implementation of the CELF.

A. Hardware-based Laboratory

The focus of hardware-based labs has been to provide a real experience [8]. This stems from faculty and instructional designer experience with computer science and engineering use of lab exercises as a complement to lecture content [10]. Hardware-based labs hold a clear advantage when an experience calls for direct interaction with hardware. As well, one should not discount that at the time when the first generation of cybersecurity labs came online [9], there were no alternatives to hardware-based setups.

However, hardware-based labs suffer from significant disadvantages [8]. Foremost management overhead is high even when configuration management or image automation is used. Further, resource limitations [11] can present a hard ceiling as there are only so many physical systems which can fit into a given space. Likewise, individual computing systems can only scale so far. Lastly, there is a multiplicative cost as hardware-based labs expand to fit more students or offer additional experiences. Existing literature suggests the multiplier for cost is between 1.5 and 2 times [8].

B. Virtualization-based Laboratory

The focus of virtualization is to consolidate infrastructure [12]. In cybersecurity labs, the consolidation was thought to be an antidote to challenges and limitations in traditional hardware

approaches [8], physical space limitations relative to increasing numbers of students wanting to access these systems. Further, virtualization eases management overhead and can simplify student access to lab systems [12].

While virtualization succeeded in addressing some issues, it did not resolve system resource (e.g., CPU, RAM, disk) limitations [8]. Notably, there are two levels of CPU, memory, and disk concerns in the context of virtualization systems engineering. First, the virtualization host must have enough CPU, memory, and disk to support operational objectives across the total set of users and use cases. Second, an individual guest virtual machine must have sufficient CPU, memory, and disk to support individual user behavior. Consolidation through virtualization falters when meeting either case of demands, particularly when access is widely distributed. As a result, recent efforts have taken researchers towards cloud infrastructure [13] and container-based labs [14].

C. Cloud-based Laboratory

The focus of cloud computing is agility [15]. Chiefly, this agility manifests through high degrees of usability and ease of access [16]. The compute resource limitations present in traditional virtualization are mitigated by abstracting computation across multiple physical hosts (e.g., compute nodes). On one hand, the mitigation comes with a potentially high upfront cost for hardware when the cloud is private although the software is open source such as CloudStack, OpenStack, and OpenNebula. On the other hand, a public cloud such as Amazon Web Services (AWS) exchanges high upfront cost for continual cost with flexible compute resource scalability [16]. Cloud-based labs do require network access whether local for a private cloud or public for commercial solutions as well. Thus, strictly speaking, the exercises are not portable. Additionally, public clouds do not permit some cyber tools or techniques (and understandably so). For these reasons, some [14][17] have begun to research the use of containers as an extension of traditional virtualization and alternative to cloud-based solutions.

D. Container-based Laboratory

Container technology such as Docker or Kubernetes focus on portability [14]. Container-based labs are portable because the host (i.e., student computer) provides the underlying compute power and OS while the container provides the application stack and configuration [18][19]. Students only need a host running a matching OS and a compatible container layer. Thus, labs can be compartmentalized across multiple containers [14] for modularity given their relatively small size and compactness compared to full virtualization. Yet, the very portability imparting an advantage ends up binding the container to a specific host type and configuration [17]. This is desirable when the host needs to be isolated from a lab but is undesirable when students are not using the prescribed host type or configuration.

E. Pedagogy and Game Design

The last segment of related work to inform the design and implementation of the CELF was pedagogy and game design. Fortunately, the advantages and disadvantages of objectivist and constructivist pedagogical modalities are well understood [20]. Further, we assume everyone intends for their labs to be constructivist based on the tone of existing literature but that is not how students report interacting with these exercises [8]. Thus, we sought to design and implement a lab capable of facilitating a different experience.

We speculate that the overarching language encapsulating the traditional design of cybersecurity labs (simulation) drives the disconnect between designers or educators and students. That is, there may be underlying confusion between how cybersecurity labs simulate real-world experience or exist as a simulation of common workplace scenarios and what simulation means in a broader context. To clarify, we turned to the field of game design since researchers there long have worked with construction of artificial experience.

We found that while cybersecurity labs are not game environments, there are conceptual overlaps based on pedagogical intent vis-à-vis simulation. More specifically, whereas traditional laboratory exercise designs are theme parks in game design terms [21], what cybersecurity labs ought to be is more akin to a sandbox [22]. Game design also gives us the concept of a phenomenarium [23]. Interestingly, the underlying concept of a phenomenarium is tightly coupled to simulation. Here, the simulation is constructed as a replication of reality with the goal of immersing the player. Notably, a game phenomenarium is devoid of procedure or script (e.g. theme park). Replacing the procedure and script is open exploration and freedom of action (e.g., sandbox).

III. Design and Implementation

A. Technical Design

Considering the body of related work, we selected a private cloud-based design for the CELF with an emphasis on sandboxed phenomenal engagement. Furthermore, the CELF diverges from the common virtualization model seen in cyber ranges and labs in that we use a horizontally scaling private cloud infrastructure with vertically scaling student desktops. The overall environment scales horizontally because of the cloud layer whereas the student desktops scale vertically due to a nested virtualization layer within their virtual machine sandbox layer. Meanwhile, there are multiple levels of network isolation to prevent unintended harm from adjacent systems.

B. Pedagogy Design

Importantly, the technical design permitted assignments within the CELF to consist of a first principle statement (e.g., least privilege) and a general description of the intended activity as opposed to a prescriptive list of steps. Thus, the underlying pedagogy was *active* and *situated* insofar as successful completion of ten laboratory assignments required students to discover how to construct a virtualized environment simulating a legitimate small business or enterprise architecture to support each of the assignments on their own. As well, the design thematically adhered to a sandbox, phenomenarium approach, allowing students to create and explore what they felt would be necessary to accomplish the assignment purpose. Here, we were careful to design the assignment descriptions to include keywords or phrases which students could operationalize into search terms.

For example, one assignment called for students to implement a file server and three workstations for a small business which allows Alice to share files in read, write, and execute mode with Bob but not Eve; Eve with Alice having read and write only but not Bob. The students must decompose the description into discrete goals themselves, construct an idealized pathway towards each goal, and bring the full sequence into reality.

Notably, students were free to select OS, configurations, and settings they felt best achieved the assignment goal in all the assignments. Thus, the experience was without rails, without predetermined functionality. Further, the students were required to submit evidence of secure configuration of their systems, networks, accounts, and filesystems objects. Assessment of student work occurred through programmed scripts which reported *true*, *false* outcomes for assignment learning outcomes.

C. Implementation

The current hardware architecture consists of seven servers, a core switch, and a firewall. Four of the servers are compute nodes, two are controller nodes, and one is a storage node. The compute nodes are Dell PowerEdge 6515 servers with an AMD 7302P CPU, 128GB RAM, and 2TB RAID 1 disk loadout. The controller nodes are Dell PowerEdge 6515 servers as well but with an AMD 7452 CPU, 256GB RAM, and 1TB RAID 1 disk loadout. The storage node is a Dell PowerEdge with 80TB of disk.

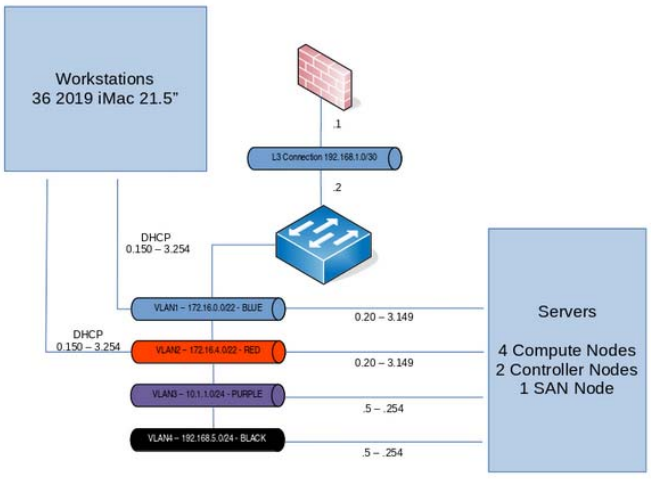

Fig. 1. The CELF architecture design and implementation.

The core network switch is a Cisco 9407R with three 48 port line cards. One port is layer three (routing) enabled and connected to a Pfsense firewall appliance for Internet access. Then, half of the ports are 10GB or higher capable which is used for private storage and management while the other half are 1GB. Further, the ports are segmented into four VLANs to support the isolation design: one Black segment for storage traffic, a Blue segment for cyber defense-oriented experiences, a Red segment for cyber offense-oriented experiences, and Purple for out of band management. The Blue and Red segments interconnect a classroom space consisting of iMac workstations and the private cloud. Netflow between all segments is controlled through access-control lists.

D. Software

All servers run CentOS 7. The controller nodes run OpenNebula and Sunstone for a web-based UI. The compute node servers run OpenNebula on top of a KVM hypervisor. All servers mount two NFS shares (40TB each) from the storage node. Further, each student is supplied with a Kali virtual desktop which includes a nested KVM instance. Each Kali instance is allocated 16GB RAM, 2 virtual CPU, and 1TB of persistent disk space. As well, each Kali instance automatically mounts a NFS share containing a curated set of ISO images. The set includes Ubuntu Server, TinyCore Linux, and intentionally vulnerable images such as Damn Vulnerable, Damn Vulnerable Web App, and so forth.

IV. Results

Preliminary results suggest the CELF is operating as expected in terms of technical implementation and pedagogical phenomenarium. We base this assertion on preliminary data from (a) questionnaire to students and (b) CPU, RAM, and Disk utilization data from the cloud compute systems. The questionnaire was administered at the conclusion of a cybersecurity security course which used the CELF to complete 10 laboratory exercises. The utilization data was collected through the CentOS *atop* software package.

When prompted, students report that the CELF fosters exploration. At first, students found the need for self-guided navigation to be overwhelming. However, by the second exercise, the majority (> 90%) of students began to describe the CELF in a positive tone. Additionally, students now use terms or phrases such as immersive, challenging, and technically demanding.

Moreover, students disagreed unanimously with the idea that CELF hosted exercises were pedagogically equivalent to lecture. Furthermore, students agreed with the assertion that the CELF was *open-ended*, *non-prescriptive*, and allowed for *self-guided exploration* of a subject. As well, students agreed that the phenomenarium approach required more self-motivation and discipline to accomplish laboratory assignment goals. However, once students figured out how to operationalize the keywords in the assignment descriptions, the burden to find relevant information was lessened.

Finally, the technical strategy of distributing overall load horizontally while allowing students to scale vertically within their personal sandbox appears to be effective. For the same lab scenario, the CELF has demonstrated a compute overhead significantly lower than an individual student personal computer. What is more, an unexpected benefit of the CELF design is the reduction of management overhead due to virtual machines being disposable. Anytime a student encounters an error condition, we have found it is more efficient to simply destroy the virtual machine and spin up a new clone. This also seems to limit frustration, thus improving the overall experience.

V. Conclusions

The cybersecurity laboratory exercise is a common instruction and assessment modality. Since its inception, the laboratory exercise has been used to walk students through the processes and procedures associated with securing systems.

While the technical design and implementation has evolved, the pedagogy has remained constant. Unfortunately, existing approaches to the cybersecurity laboratory have limitations and challenged [8].

Consequently, we drew inspiration from game design and reimagined the cybersecurity laboratory as a sandbox or phenomenarium. To achieve our goal, we constructed a learning facility based on vertically scaling cloud computing architecture and developed laboratory assignments to take advantage of the hardware and software. Assignments were open-ended instead of stepwise, thus requiring exploration and active engagement as opposed to rote following of procedure. Overall, students reported positive perceptions of the design and implementation.

To be certain, the CELF is ultimately limited by physical resources just as any other lab design. The existing design is scaled to maximize the phenomenological immersion of 36 students which is the maximum capacity of the physical classroom. Fortunately, the cloud-based architecture facilitates predictable scaling at a reasonable cost (approximately $20,000). Absent procurement of additional systems, there is potential for future work in how to precisely tune the Kali virtual desktops and nested virtualization. What is more, future work is necessary to outline what types of exercises can or cannot best be adapted to a phenomenarium approach.

REFERENCES

[1] Y. Wang, M. McCoey, and H. Zou. “Developing an Undergraduate Course Curriculum on Information Security.” In Proceedings of the 19th Annual SIG Conference on Information Technology Education, pp. 66-71, 2018.

[2] S. Abraham and S. Lifang, “Instructional perspective: towards an integrative learning approach in cybersecurity education.” Inf. Secur. Educ. J, pp. 84-90, 2015.

[3] W. Yurcik and D. Doss. “Different approaches in the teaching of information systems security.” In Proceedings of the Information Systems Education Conference, pp. 32-33. 2001.

[4] A. S. Andreatos, "Designing educational scenarios to teach network security." In 2017 IEEE Global Engineering Education Conference (EDUCON), pp. 1606-1610. IEEE, 2017.

[5] D. Yates, M. Frydenberg, L. Waguespack, I. McDermott, J. OConnell, F. Chen, and J. S. Babb. "Dotting i’s and Crossing T’s: Integrating Breadth and Depth in an Undergraduate Cybersecurity Course." Information Systems Education Journal 17, no. 6 (2019): 41.

[6] J. M. Pittman, and H. Barker. "Are Cybersecurity Laboratory Exercises Constructivist in Use?" In Journal of The Colloquium for Information System Security Education, vol. 2, no. 1, pp. 11-11. 2014.

[7] J. A. Chisholm. "Analysis on the perceived usefulness of hands-on virtual labs in cybersecurity classes." PhD diss., Colorado Technical University, 2015.

[8] J. M. Pittman. "Understanding system utilization as a limitation associated with cybersecurity laboratories–A literature analysis." Journal of Information Technology Education: Research 12, no. 1 (2013): 363-378.

[9] P. C. Clark. “Supporting the education of information assurance with a laboratory environment,” NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 2001.

[10] C. E. Irvine. “Amplifying security education in the laboratory,” NAVAL POSTGRADUATE SCHOOL MONTEREY CA DEPT OF COMPUTER SCIENCE, 1999.

[11] W. I. Bullers Jr, S. Burd, and A. F. Seazzu. "Virtual machines-an idea whose time has returned: application to network, security, and database courses." ACM SIGCSE Bulletin 38, no. 1 pp. 102-106, 2006.

[12] J. L. Duffany and A. Cruz. "Design of a computer security teaching and research laboratory." In Proceedings of the 43rd ACM technical symposium on Computer Science Education, pp. 678-678. 2012.

[13] W. Zhu. "Cloud-based Labs and Programming Assignments in Networking and Cybersecurity Courses." In *2018 IEEE Frontiers in Education Conference (FIE)*, pp. 1-9. IEEE, 2018.

[14] C. E. Irvine, M. F. Thompson, and J. Khosalim. "Labtainers: a framework for parameterized cybersecurity labs using containers," 2017.

[15] Y. Sheng, H. Fan, L. Xiao, and J. Huang. "A virtual experiment platform based on OpenStack." In 12th International Conference on Computer Science and Education (ICCSE), pp. 744-749. IEEE, 2017.

[16] C. Tunc, S. Hariri, F. De La Peña Montero, F. Fargo, and P. Satam. "CLaaS: Cybersecurity Lab as a Service--design, analysis, and evaluation." In 2015 International Conference on Cloud and Autonomic Computing, pp. 224-227. IEEE, 2015.

[17] V. O. Shanmughan. "Lightweight Environment for Cyber Security Education." PhD diss., University of New Orleans, 2017.

[18] L. Tobarra, A. Robles-Gómez, R. Pastor, R. Hernández, A. Duque, and J. Cano. "Students’ Acceptance and Tracking of a New Container-Based Virtual Laboratory." *Applied Sciences* 10, no. 3 (2020): 1091.

[19] J. Sianipar, C. Willems, and C. Meinel. "A container-based virtual laboratory for internet security e-learning." International Journal of Learning and Teaching. IJLT 2, no. 2, pp. 121-128, 2016.

[20] J. Hautamäki, M. Karjalainen, T. Hämäläinen, and P. Häkkinen. "Cyber security exercise: Literature review to pedagogical methodology." In INTED Proceedings, no. 2019. IATED Academy, 2019.

[21] G. Majgaard. "The Playful and Reflective Game Designer." Electronic Journal of E-learning 12, no. 3 (2014): 271-280.

[22] A. Bauer, E. Butler, and Z. Popović. "Dragon architect: open design problems for guided learning in a creative computational thinking sandbox game." In Proceedings of the 12th International Conference on the Foundations of Digital Games, pp. 1-6. 2017.

[23] O'Donnell, Fionnuala, and Brendan Tangney. "Towards' Phenomenaria'in the teaching of distributed systems concepts." In Proceedings 3rd IEEE International Conference on Advanced Technologies, pp. 256-257. IEEE, 2003.