Cyber Arena: An Open-Source Solution for Scalable Cybersecurity Labs in the Cloud

Philip Huff
Department of Computer Science
Emerging Analytics Center
University of Arkansas at Little Rock
Little Rock, AR, USA
pdhuff@ualr.edu

Sandra Leiterman
Emerging Analytics Center
University of Arkansas at Little Rock
Little Rock, AR, USA
saleiterman@ualr.edu

Jan P. Springer
Emerging Analytics Center
Department of Computer Science
University of Arkansas at Little Rock
Little Rock, AR, USA
jpspringer@ualr.edu

ABSTRACT

Numerous institutions are developing cybersecurity education and training programs to supply the considerable global demand for cybersecurity professionals. However, these institutions face barriers in building realistic laboratory environments, commonly referred to as *cyber ranges*, needed for hands-on skills development. Cybersecurity labs differ from traditional computing labs in both size and complexity. They often require multiple distinct components to represent network configurations, adversarial computing, and defense mechanisms.

We present an open-source cloud-based cybersecurity laboratory solution, enabling broader access to cybersecurity education by reducing complexity barriers as well as costs. Our solution already serves over one thousand students each year using the Google Cloud platform in a variety of learning environments. In this paper, we describe our solution's architecture and deployment characteristics. Additionally, we discuss challenges in educational institutions when using cloud-infrastructure platforms and our approaches in addressing these challenges.

CCS CONCEPTS

• **Applied Computing** → **Interactive Learning Environments**.

KEYWORDS

Cyber Range, Cloud Computing, System Security

ACM Reference Format:
Philip Huff, Sandra Leiterman, and Jan P. Springer. 2023. Cyber Arena: An Open-Source Solution for Scalable Cybersecurity Labs in the Cloud. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569828

1 INTRODUCTION

The workforce demand for cybersecurity professionals has remained consistently strong with an expected growth rate of 33% in the U.S. over the decade until 2030 [23] and an expected 3.5 million job openings worldwide [12]. Consequently, academic institutions have been actively engaged in developing programs to meet this demand. Many of these programs require hands-on lab activities, usually supported by cyber ranges [18]. However, the cost to implement and maintain these cybersecurity labs can be prohibitively high.

The first cyber ranges were developed for national defense initiatives to provide realistic cyberattack scenarios [15]. Since then, many cyber-range installations have emerged to meet the needs of cybersecurity research, education, and testing. For example, Ukwandu et al. [22] identified 48 distinct cyber ranges and found 31% of those to be serving an academic purpose. Features of a cyber range include scenario definitions, teaming capabilities as well as administrative functions such as management, monitoring, and scoring [27].

Cyber ranges serve multiple purposes, such as training, testing, exercises, and education. However, a recent survey by Chouliaras et al. [7] finds that most cyber ranges focus primarily on exercises and competitions, which involve building highly complex and realistic environments with Internet emulation and building teams between attacking and defending parties. In contrast, cybersecurity education labs have much simpler architectural requirements. For example, they do not necessarily require the emulation of complex systems when teaching concepts such as cross-site scripting vulnerabilities or asymmetric cryptography. Our design is able to minimize a cybersecurity lab's build complexity in the cloud to meet the needs of specific learning objectives.

We deploy our solution as open-source software to fill the broad gap in access to quality cybersecurity labs. Additionally, open-source cloud solutions align well with the academic environment by requiring only metered computing infrastructure rather than costly on-site infrastructure investments. A few open-source projects have emerged in recent years for developing cybersecurity labs. However, many of these focus on providing infrastructure technology and do not come with scenarios or platform support [14]. In contrast, our solution, further referred to as the Cyber Arena [9], focuses on the learning platform and supports students, instructors, and administrators.

Our contributions in this paper include:

(1) Our approach to enabling a production-scale open-source cyber-range project,
(2) Addressing challenges and discussing solutions to cloud cybersecurity labs in educational institutions, and
(3) Presenting cost figures and run times for cloud labs over various instructional environments.

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada.

ACM ISBN 978-1-4503-9431-4/23/03…$15.00
https://doi.org/10.1145/3545945.3569828

2 RELATED WORK

Cybersecurity laboratory platforms are intended to simplify the process for an instructor specifying and deploying their labs. Several initiatives exist to automate the deployment of educational cybersecurity labs using an *Infrastructure as Code* (IaC) specification language [17, 3]. For example, the *Cyber Range Instantiation Systems* (CyRIS) uses a YAML specification to define a system, deploy software and settings, and create instance clones of a system for multiple learners [19]. CyRIS is a component of the *Cybersecurity Training and Operation Network Environment* (CyTrONE), which includes a front-end application to build predefined labs and integrate these labs into learning management systems [4].

KYPO uses a set of YAML specifications to build competition scenarios [24, 25]. Vykopal et al. [24] refer to the specifications as sandboxes to orchestrate the low-level infrastructure configuration. The specifications abstract cloud-ready build instructions written in Ansible [1].

The *Cyber Range Automated Construction Kit* (CRACK) provides a higher-level scenario-definition language (SDL) on top of an IaC standard [20]. The SDL has two high-level objects, referred to as nodes, and relationships to define the infrastructure build as well as an interface object, which defines actions to perform on the nodes. The SDL is used to produce standard IaC TOSCA instructions [5]. Similarly, the *Austrian Institute of Technology* (AIT) uses the TerraForm IaC system [21] and deploys on the OpenStack cloud infrastructure [13].

More recently, Yamin and Katt [26] used an SDL to build an educational cybersecurity environment. It also allows simulating weaknesses, attacks, and defensive actions. The specification is written in Backus-Naur form (BNF) to allow the use of Datalog [6] for scenario validation. Automating a real-time cybersecurity scenario is more complex than automating the infrastructure build. Thus, the more complex a BNF specification, it is warranted to improve efficiency in validating and testing the scenario.

Our approach provides a simpler specification for use in various learning environments. First, we avoid specifying the software and data configuration of computing objects. Instead, we allow instructors to configure and image their servers before deployment. Secondly, we developed an IaC layer in a Cyber Arena application to interact directly with the cloud provider. We have found that many cybersecurity education scenarios do not warrant the overhead of generic IaC specification languages such as Terraform, Ansible, or TOSCA. Moreover, directly interacting with the cloud provider allows access to cloud features such as timeouts, DNS, and many others not yet accessible in the IaC standard.

3 SYSTEM ARCHITECTURE

3.1 Deployment Prerequisites

The Cyber Arena system initially deploys to a cloud project using a setup script at the root of the code repository. The installation configures the cloud project with the necessary service accounts, application code, and publish/subscribe messaging infrastructure for human interaction and back-end automation. In addition, the following prerequisites are necessary for deployment:

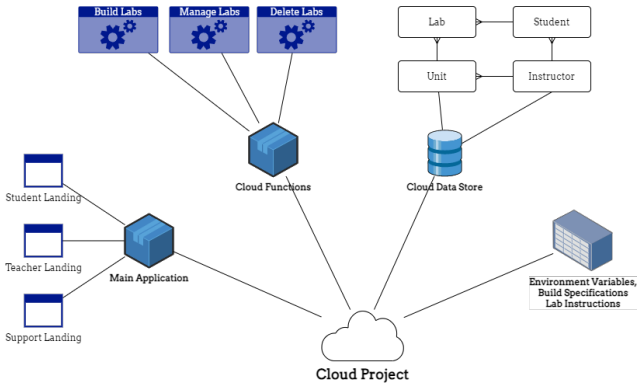

Figure 1: Cyber Arena system architecture. Our architecture consists of the main application as an interface to users (e.g., students, teachers, administrators), cloud functionality to build, manage, and deploy lab instances, a data store containing institutional and administrative information, and general build specifications.

(1) **Cloud Project** - A Cyber Arena project currently works with Google Cloud only. However, the system architecture is generic and portable to other cloud service providers.
(2) **Authentication Provider** - Most deployed web-application services require authentication. Using a cloud authentication provider offloads a significant portion of the support services' overhead in maintaining a cybersecurity lab deployment.
(3) **DNS Service** - Entry points for labs require a transitory IP address each time students start their lab instance. A cloud project requires control of the DNS domain to automatically manage DNS entries for dynamic IP address assignment.
(4) **Quota Evaluation** - Cloud services use quota limits to prevent (severe) cost overruns. A typical cloud project will not require the maximum possible standalone virtual networks. However, quota limits will typically require adjustment to support a maximum concurrent number of students.

Figure 1 shows the primary cloud components involved in the operation of the Cyber Arena. The *main application* provides the interface for students, teachers, and support administrators and runs as a serverless container. The user initiates cloud build operations in the main application and passes information to *cloud functions* using asynchronous publish/subscribe protocols. Offloading operations to cloud functions avoid long wait times and allows parallel build operations. The cloud data store is a simple NoSQL database service. We chose a data store over a traditional relational database to reduce cloud costs associated with an always-on back-end service.

Each institution's instructor designates their students' email addresses as records in the data store. When an instructor builds a set of labs, the build specification is stored as a classroom unit. Additionally, each student lab in the unit has its lab specification record stored. These four types of records provide all the data necessary to build, present, and maintain cybersecurity education labs. Private *cloud functions* are used as services for building, managing, and deleting the labs. When an instructor builds a lab, the *main application* stores the specification in the *cloud-data store* and passes the data-store key to the cloud function for processing. The

cloud function then asynchronously pulls the specification from the data store and builds the lab environment. The parallel building of each student lab allows an instructor to immediately distribute labs while the cloud functions manage the allocation of new computing resources.

Students can also interact with their labs by initiating cloud functions through their landing page on the main web application. Once students start their lab, a user-defined timer is initiated, which automatically triggers the shut down of computing resources after a specified period. Additionally, a scheduled cloud-service task is executed several times per hour to invoke cloud functions to search for and stop idling labs. During the build process, instructors specify how long they expect labs to be available and an expiration timestamp is stored for each lab unit. A scheduled task executes once per hour to process any expired units. Expired units will have all computing resources deleted from the cloud project, saving costs and preventing expired resources from counting against cloud quotas.

Environmental data defining the institution and the labs persist in the data store and cloud-storage buckets. When the Cyber Arena deploys, information about the institution, such as instance name, administrators, or design preferences, are stored in an environment data store. The specifications and instructions for all labs are uploaded to a storage bucket used in the main application to present the labs to instructors and students. The server images required to run each lab are also uploaded to the cloud project.

3.2 Build Categories

The Cyber Arena system supports four specification types for automatic cybersecurity lab deployment:

(1) **Compute** - Requires allocation of virtual resources for networking and server interaction. An instructor configures a set of virtual servers with all the necessary tools and data sets to complete the exercise. The server is imaged once and deployed in a predefined network configuration for each student.
(2) **Serverless** - Docker [16] containers are deployed as a web application. Serverless labs have drastic cost savings over those requiring allocated compute resources and work well for cryptography or web-security learning objectives.
(3) **Classroom** - A combination of fixed resources and ephemeral student resources to allow in-class exercises where all students have servers in the same network. The classroom build allows an instructor to synchronously walk through examples with all students.
(4) **Competition** - Instructors are able to deploy competitions for special events, practice, or capstone projects using compute and serverless resources. Students synchronously participate in the same network and receive points for completing specific objectives in these lab instances.

The Cyber Arena's existing lab specifications are categorized in Table 1 by their build types along with additional information (e.g., number of available specific labs, basic requirements).

Category	Lab Build Type	# of Labs	Description
Access Control	Compute	6	Requires installed software to perform access-management activities
Adversarial Thinking	Compute	4	Requires compute resources for both the attacker and defender
Cryptography	Serverless / Compute	5	Most labs can be serverless; computing resources are helpful when students work with configuration and key management
Digital Forensics	Compute	4	Requires special software and preloaded data sets to perform forensic analysis
IoT Security	Serverless	2	IoT devices interact through publish/subscribe protocols, which tie into serverless web applications
Network Security	Compute	7	Requires traffic generation and analysis involving multiple virtual devices or networks
Log Analysis	Compute	2	Specialized log analysis and indexing software
Vulnerability Management	Serverless / Compute	4	Requires specialized software for vulnerability scanning; some public data sets may be analyzed using web applications
Web Security	Serverless	6	Web-based attacks (e.g., cross-site scripting, SQL injection)

Table 1: Mapping of Cyber Arena lab categories to build categories with an indication of how many actual lab units per Cyber Arena lab category are available as well as basic requirements connected to each of the Cyber Arena lab category.

3.3 Infrastructure as Code

Existing Infrastructure as Code (IaC) languages can be complex and require substantial effort to develop and maintain, which presents a barrier to broader adoption and development of new cybersecurity labs. Instead, in the Cyber Arena, we translate the human-readable specification directly into cloud resources using the cloud provider's API. Many cybersecurity labs have a typical architecture with one or two network segments, a few network servers, and usually a gateway for the students to access lab resources. Therefore, we can abstract the IaC to require a minimal specification for new labs. An example YAML specification and the instructor build form is shown in Figure 2.

Build specifications are maintained in private cloud storage. When an instructor builds a new lab, the *main application* pulls a human-readable specification, expands the specification to include student information, and passes the specification through a schema checker. The *main application* stores the IaC output in the *cloud-data store* and triggers the build operation in the back-end *cloud function* through a publish/subscribe message, which allows the asynchronous build of any number of student labs.

Back-end *cloud functions* allow rapid deployment by recursively calling itself through publish/subscribe messages to avoid waiting

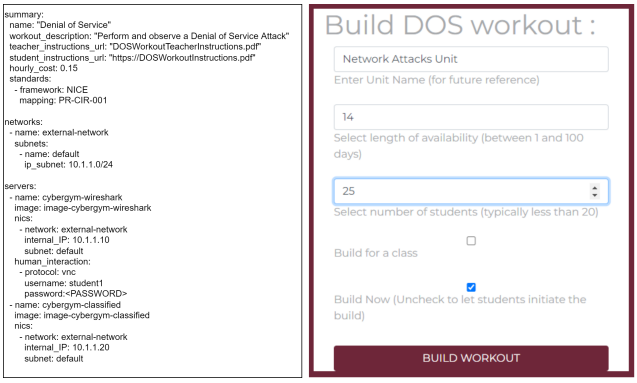

Figure 2: The YAML specification on the left describes the minimal elements necessary to build and replicate a cybersecurity lab for student use. Instructors create new labs using the web interface shown on the right.

on any cloud API calls. Consequently, instructors can build large deployments of cloud labs for their students during their time of instruction with the trade-off of increased complexity in the build operations. We can manage the complexity using a state model in which every lab record in the *cloud-data store* has a recorded state.

Figure 3 shows the abstract state model describing the lifetime of a cybersecurity lab in the Cyber Arena. Several states are defined to build the cloud resources according to their dependencies, and, after a successful build, a lab moves into a *READY* state. The maintenance routine will look for expired labs and automatically move those to *DELETING* state. If a failure is detected, the lab is moved into *BROKEN* state.

Developing the IaC directly on the cloud service layer allows tighter control over the lab resources and a better alignment with the instructional environment. For example, instructors can perform just-in-time deployments for their classes regardless of class sizes, while students can monitor and control labs assigned to them. The

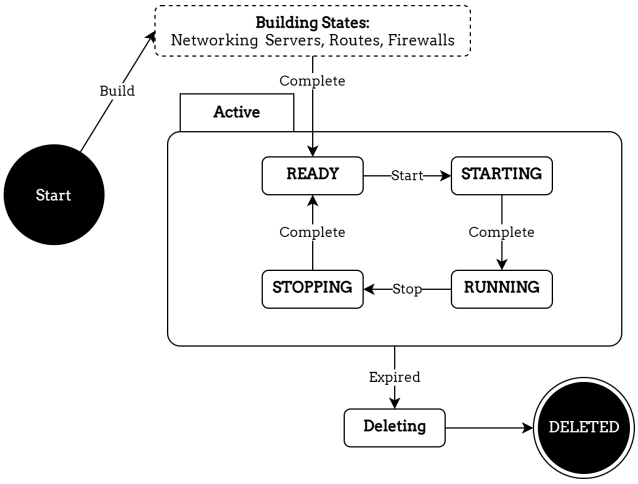

Figure 3: To allow for asynchronous deployment and automated troubleshooting, each lab follows a state model from start to expiration and deletion.

state model also allows for faster troubleshooting when failures occur, benefiting the instructional environment to ensure labs are available for student assignments and assessments.

3.4 Customized Deployments

Instructors can quickly develop labs using the build specification and customized virtual machine images. Most of the deployment work involves setting up a virtual machine with required software, data, user settings, and any automated scripts for lab interaction. New virtual machines can be created in the cloud project's default network and imaged using a dedicated script at [8]. Alternatively, existing virtual disk images can be uploaded to the cloud as described in [11]. Then, instructors are able to use the images' names to specify the deployment. Instructors will also find step-by-step instructions for specifying and deploying labs in the main `README` file of the open-source project page at the [9].

3.5 Cost Optimization

We reduce cost barriers in providing cybersecurity labs by aligning computing resources to instructional time. In our experience, server resources used in compute labs account for most of these costs. Likewise, several *cloud functions* are employed to automatically shut down idling or unused resources. The functions include shutting down servers immediately after they are built and allowing students to turn them on when they begin their work. Furthermore, in the Cyber Arena, students have a default instruction period of two hours. Therefore, we employ a *cloud function*, which is invoked every 15 minutes to shutdown down servers exceeding the instructional time allotment. Each cybersecurity lab in the *cloud-data store* also contains an expiration timestamp set by the instructor. A *cloud function* is invoked once per hour to deallocate any cloud resources associated with expired labs. The automated deletion of resources eliminates the lesser costs of storage and network allocation while deletion also frees up resources counting against a cloud project's quota.

Figure 4 shows the cost of a cybersecurity course for approximately five hundred students in the academic year spanning 2020 through 2021. Compute time and use of related data store dominates most of the costs. Spikes throughout the year represent portions of the cybersecurity curriculum requiring more computing resources for lab work by students. In contrast, serverless computing resources such as the *main application*, *cloud functions*, or serverless labs scarcely show in the graph. The overall cloud costs for the academic year on display were approximately $4,000 with serverless computing accounting for less than $15 during that period.

One downside of metered cloud billing is the risk of cost overruns. Many educational institutions cannot afford or may not even have the allowance to exceed their budget. However, most cloud-service providers only send notifications when costs exceed a specific budget and a cloud project may rapidly incur costs over short periods of time. Consequently, the Cyber Arena continuously monitors cost accumulation and automatically restricts the use of cloud resources when costs exceed a specified budget amount to avoid overspending.

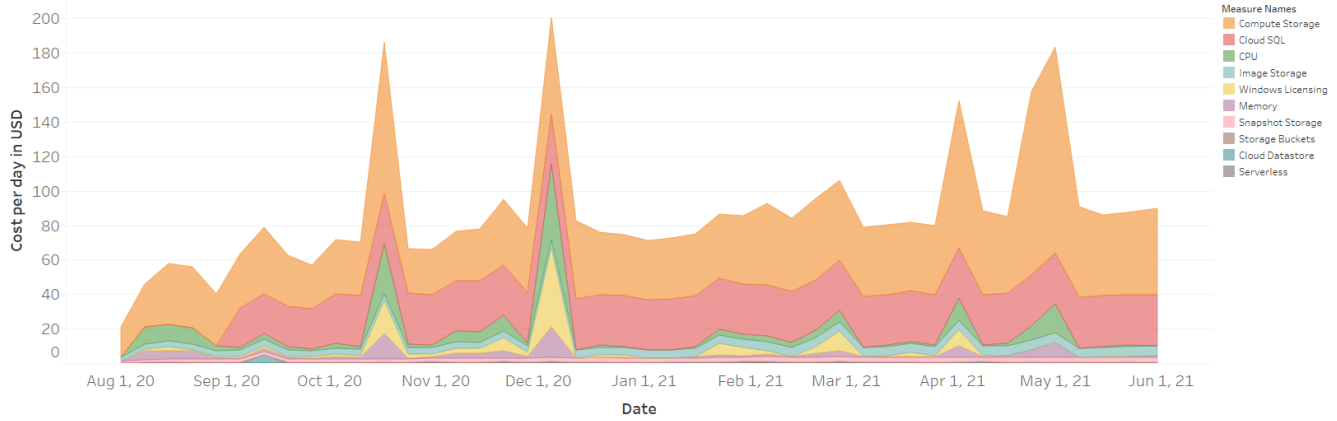

Figure 4: Cyber Arena cloud costs by category for the academic year 2020 – 2021. Compute time and related data store is the most significant contributor to the costs compared to serverless compute resources, which barely register in the graph.

3.6 Assessment

Each Cyber Arena lab allows the inclusion of student assessment in its specification. Most existing labs use the assessment to increase difficulty levels to engage multiple skill levels. This way, the assessments motivate students to complete objectives rather than test subject-matter knowledge. Labs have assessment information placed in the *cloud-data store* under the key for the student's lab. An assessment may be performed automatically or require the student to provide input such as a challenge flag. For automated assessments, the *main application* provides a restricted API to mark the completion of a lab assessment. Designated computing elements in a student's lab periodically test the state of the lab and set the completion status when the student has finished a given mission. Although it is possible to preload scripts onto server images for automated assessment, it requires re-imaging the server setup each time a new assessment is added. Instead, the Cyber Arena maintains assessment scripts in private cloud storage and designated servers load the startup scripts during the build process. For example, the denial of service lab assessment scripts (cf. [10]) will be dynamically loaded from startup scripts residing in the cloud-storage environment. Students then use denial of service tools with various traffic generation options. The script on the targeted server runs once per minute to check the server state. Once the CPU utilization of the targeted server exceeds 40% the script marks the challenge as completed.

4 IMPLEMENTATION

The Cyber Arena labs were used by over 1,000 students in various learning environments between July 1, 2021, and June 30, 2022 using a multitude of separate autonomous cloud projects. Each cloud project is categorized according to its learning environment, as shown in Figure 5. The *high school* learning environment included 538 labs for students in grades 9–12. These labs are aligned with the state-published curriculum. Additionally, learning environments in the *testing and event* category consisted of 1,670 labs. They included special events such as summer camps, hackathon competitions, STEM outreach, and cybersecurity-awareness events. This category also includes the required testing to ensure all labs run in their respective environments as intended. The *workforce development* category includes 782 labs used for *upskilling* and *reskilling* of adult students. Finally, the *higher education* category includes 242 labs from students enrolled in undergraduate and graduate cybersecurity degree programs in the Department of Computer Science at UA Little Rock.

Students' average time in a lab was around two to five hours and varied depending on the learning environment. For example, *high school* and *testing and events* labs, having more basic learning and assessment objectives, tend to be shorter than the average lab duration. In contrast, *higher education* students often spend between a few weeks and an entire semester in a single lab! *Workforce development* students will also spend considerable time in their labs. However, training programs are often limited to a few weeks, limiting the maximum time of lab availability.

Overall, the average cloud cost per lab was approximately $0.59 per hour. Costs in the *higher education* category were lower, reflecting the relatively inexpensive labs used for a semester-long course. Both the *high school* and *testing and events* categories exhibited higher costs for their labs relative to the time spent in them by

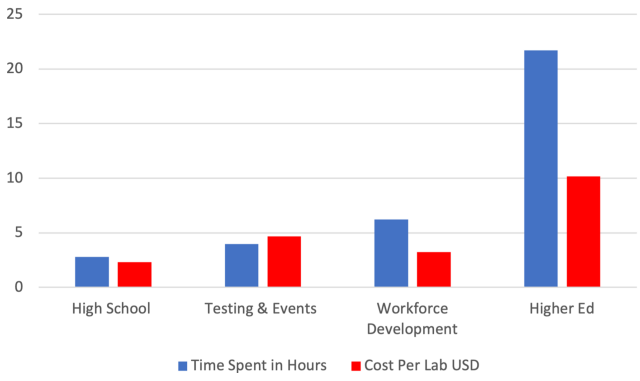

Figure 5: Average time spent per student and average cost per lab grouped by category of learning environment during the time period July 1, 2021 and June 30, 2022.

students. For the *high school* category, students heavily used a competition lab, which contained several servers with an approximate cost of $0.76 per hour. Labs in the *testing and events* category often used more expensive computing platforms to ensure faster booting and response times.

5 CHALLENGES

5.1 Distribution

Providing access to cybersecurity-relevant labs to students is challenging because institutional hardware or software changes often involve lengthy procurement and installation procedures. While cloud infrastructure avoids such costly lab setups, students must still access their labs using personal or institution-issued computers. Thus, similar barriers exist when laptops require new software or changes in network configuration occur. Consequently, the Cyber Arena provides a platform-independent gateway to each lab using [2], a browser-based web application. A Guacamole server is installed with a public IP address for each student in their cloud-based lab. A startup script is constructed during the lab's build process to provide proxied access to either a remote desktop protocol (RDP) connection for Windows servers or a virtual network computing (VNC) connection for Linux servers. The build process also creates a pseudo-randomly generated password for each student, which is then displayed on a student's landing page in the *main application*. Ephemeral IP addresses are used to avoid the cost of reserved static IP addresses. Thus, *cloud functions* must also manage DNS records for the Guacamole server gateway. Once the server starts, a *cloud function* samples the public IP address and modifies the DNS record, which has a short expiration period for fast propagation to the student. This way, a student only needs to navigate to their lab URL when the server starts and the entire lab becomes accessible through a single Guacamole server.

5.2 Scalability

One of the principal tenets of cloud computing is rapid elasticity in which resources scale horizontally to meet demand [15]. However, cloud providers in practice must set quotas to prevent enormous computing costs from unrestricted growth. While this constraints costs, it also restricts scalability for many academic institutions, which usually lack the contractual agreements with cloud providers to scale beyond entry-level quota restrictions. Cloud providers use quotas for planning purposes in a given data-center location. Hard quota limits are set to avoid resource exhaustion at the data center or for a geographic region. Although these limits are not published, we experienced limiting quotas for scaling the Cyber Arena to about 300 virtual private cloud (VPC) networks.

We were able to work around this hard quota limit by creating child-cloud projects derived from a parent-cloud project. Cloud projects are a collection of computing resources for individuals or organizations that the cloud provider, in our case Google, uses for regional planning. However, other cloud providers have similar concepts involving geographic regions. Cloud projects are linked through account permissions using a deployment script. For a given institutional application, the *main application* and *cloud-data store* reside solely in the parent project.

In contrast, each child project only requires installation of *cloud functions* and *environmental data*. When an instructor requests a build for their class, a resource manager in the *main application* allocates labs to itself and available child projects according to the availability of cloud resources. Thus, the complexity of resource scaling across cloud projects is abstracted from the instructor and other users.

6 CONCLUSIONS

Our work on the Cyber Arena provides an open-source solution for educational and other institutions offering hands-on cybersecurity training. Educators may share their Cyber Arena lab specifications as an open-source solution for repeatable classroom experiences across institutional boundaries. Specific contributions of our system include a serverless infrastructure for managing cybersecurity labs, an abstracted *Infrastructure as Code* specification, and a scalable solution to manage cloud quotas. Moreover, our solution provides cost optimization in the cloud to encourage broader adoption.

The Cyber Arena has been successfully deployed to over 1,000 students and, to date, the deployment primarily aligns lab experience with the instructional curriculum. However, the broad adoption will allow researchers to study how the characteristics of lab environments not only align with the curriculum but improve cybersecurity understanding. In future research, we plan to analyze collected information about the engagement and effectiveness of different types of cybersecurity labs to support the analysis of skill development. Future work will also include specifying more advanced network configurations and adversarial functions to better emulate real-world systems.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under award 1623628.

REFERENCES

[1] Ansible Core Documentation. URL: https://docs.ansible.com/ansible-core/devel/index.html (visited on December 2022).

[2] Apache Guacamole. URL: https://guacamole.apache.org/ (visited on December 2022).

[3] M. Artac, T. Borovssak, E. Di Nitto, M. Guerriero, and Damian A. T. "Devops: Introducing Infrastructure-as-Code." In: *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. IEEE Computer Society. 2017, pp. 497–498. DOI: 10.1109/ICSE-C.2017.162.

[4] R. Beuran, D. Tang, C. Pham, K. Chinen, Y. Tan, and Y. Shinoda. "Integrated Framework for Hands-On Cybersecurity Training: CyTrONE." *Computers & Security* 78 (2018), pp. 43–59. DOI: 10.1016/j.cose.2018.06.001.

[5] T. Binz, U. Breitenbücher, O. Kopp, and F. Leymann. "TOSCA: Portable Automated Deployment and Management of Cloud Applications." In: *Advanced Web Services*. Ed. by A. Bouguettaya, Q. Z. Sheng, and F. Daniel. Springer, 2014, pp. 527–549. DOI: 10.1007/978-1-4614-7535-4_22.

[6] S. Ceri, G. Gottlob, and L. Tanca. "What You Always Wanted to Know About Datalog (And Never Dared to Ask)." *IEEE*

Transactions on Knowledge and Data Engineering 1 (1) (1989), pp. 146–166. DOI: 10.1109/69.43410.

[7] N. Chouliaras, G. Kittes, I. Kantzavelou, L. Maglaras, G. Pantziou, and M. A. Ferrag. "Cyber Ranges and Testbeds for Education, Training, and Research." *Applied Sciences* 11 (4) (2021), p. 1809. DOI: 10.3390/app11041809.

[8] Cyber Arena Admin Scripts. *Create Production Image*. URL: https://github.com/emerginganalytics/cyberarena/blob/master/admin_scripts/create_production_image.py (visited on December 2022).

[9] Cyber Arena Github Project. URL: https://github.com/emerginganalytics/cyberarena (visited on December 2022).

[10] Cyber Arena Startup Scripts. *DoS Completion Q1*. URL: https://github.com/emerginganalytics/cyberarena/blob/master/build-files/startup-scripts/dos-completion-q1.py (visited on December 2022).

[11] Google Compute Engine. *Import Virtual Disks*. URL: https://cloud.google.com/compute/docs/import/importing-virtual-disks (visited on December 2022).

[12] Herjavec Group. *The 2019 Official Annual Cybersecurity Jobs Report*. URL: https://www.herjavecgroup.com/2019-cybersecurity-jobs-report-cybersecurity-ventures (visited on December 2022).

[13] M. Leitner, M. Frank, W. Hotwagner, G. Langner, O. Maurhart, T. Pahi, L. Reuter, F. Skopik, P. Smith, and M. Warum. "AIT Cyber Range: Flexible Cyber Security Environment for Exercises, Training and Research." In: *EICC 2020: Proceedings of the European Interdisciplinary Cybersecurity Conference*. ACM. 2020, pp. 1–6. DOI: 10.1145/3424954.3424959.

[14] T. Lieskovan and J. Hajný. "Building Open Source Cyber Range To Teach Cyber Security." In: *The 16th International Conference on Availability, Reliability and Security*. ACM. 2021, pp. 1–11. DOI: 10.1145/3465481.3469188.

[15] P. Mell and T. Grance. "The NIST Definition of Cloud Computing" (2011). NIST Special Publication 800-145.

[16] D. Merkel. "Docker: Lightweight Linux Containers for Consistent Development and Deployment." *Linux Journal* 2014 (239) (2014), #2.

[17] K. Morris. *Infrastructure as Code: Managing Servers in the Cloud*. O'Reilly Media, 2016.

[18] NICE Cyber Range Project Team. *The Cyber Range: A Guide*. 2020. URL: https://www.nist.gov/document/cyber-range-guide (visited on December 2022).

[19] C. Pham, D. Tang, K. Chinen, and R. Beuran. "CYRIS: A Cyber Range Instantiation System for Facilitating Security Training." In: *SoICT '16: Proceedings of the Seventh Symposium on Information and Communication Technology*. ACM. 2016, pp. 251–258. DOI: 10.1145/3011077.3011087.

[20] E. Russo, G. Costa, and A. Armando. "Building Next Generation Cyber Ranges with CRACK." *Computers & Security* 95 (2020), p. 101837. DOI: 10.1016/j.cose.2020.101837.

[21] Terraform Language Documentation. URL: https://www.terraform.io/language (visited on December 2022).

[22] E. Ukwandu, M. A. B. Farah, H. Hindy, D. Brosset, D. Kavallieros, R. Atkinson, C. Tachtatzis, M. Bures, I. Andonovic, and X. Bellekens. "A Review of Cyber-Ranges and Test-Beds: Current and Future Trends." *Sensors* 20 (24) (2020), p. 7148. DOI: 10.3390/s20247148.

[23] US Bureau of Labor Statistics. *Occupational Outlook Handbook: Information Security Analysts*. 2021. URL: https://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm (visited on December 2022).

[24] J. Vykopal, R. Ošlejšek, P. Čeleda, M. Vizváry, and D. Tovarňák. "KYPO Cyber Range: Design and Use Cases." In: *ICSOFT '17: Proceedings of the 12th International Conference on Software Technologies*. SciTePress, 2017, pp. 310–321. DOI: 10.5220/0006428203100321.

[25] J. Vykopal, M. Vizváry, R. Ošlejšek, P. Čeleda, and D. Tovarňák. "Lessons Learned From Complex Hands-on Defence Exercises in a Cyber Range." In: *2017 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2017, pp. 1–8. DOI: 10.1109/FIE.2017.8190713.

[26] M. M. Yamin and B. Katt. "Modeling and Executing Cyber Security Exercise Scenarios in Cyber Ranges." *Computers & Security* 116 (2022), p. 102635. DOI: 10.1016/j.cose.2022.102635.

[27] M. M. Yamin, B. Katt, and V. Gkioulos. "Cyber Ranges and Security Testbeds: Scenarios, Functions, Tools and Architecture." *Computers & Security* 88 (2020), p. 101638. DOI: 10.1016/j.cose.2019.101636.