# CHEESE: Cyber Human Ecosystem of Engaged Security Education

Rajesh Kalyanam
*Purdue University*
West Lafayette, USA
rkalyana@purdue.edu

Baijian Yang
*Purdue University*
West Lafayette, USA
byang@purdue.edu

Craig Willis
*University of Illinois at Urbana Champaign*
Champaign, USA
willis8@illinois.edu

Mike Lambert
*University of Illinois at Urbana Champaign*
Champaign, USA
lambert8@illinois.edu

Christine Kirkpatrick
*San Diego Supercomputer Center*
San Diego, USA
christine@sdsc.edu

***Abstract*—This Innovative Practice Full Paper presents CHEESE, a platform for cybersecurity education that complements formal classroom instruction with hands-on experience. With the ubiquitous use of computing devices and applications today, the protection of personal and privileged information is a persistent challenge. Modern software applications are typically complex pieces of code that borrow from various preexisting software libraries. Consequently, a flaw in one piece of software can have far-reaching and often unintended security implications that malicious actors can exploit. Thus, cybersecurity education needs to be transformed from a purely academic enterprise for cybersecurity researchers into a necessary skill that is imparted to the current and future IT workforce at large. CHEESE aims to impart such skills.**

**CHEESE is composed of CHEESEHub, a public web-platform hosting demonstrations of cybersecurity concepts, a set of lessons complementing the demonstrations, and a community-driven approach to the contribution of new demonstrations and lessons. CHEESE is intended to supplement and enhance traditional cybersecurity education with hands-on training that has been shown to improve concept retention and understanding. Instructors can incorporate CHEESE into their teaching in several ways: by utilizing one or more of the demonstrations hosted on the publicly-accessible CHEESEHub in conjunction with the web-accessible lessons; by deploying their own version of CHEESEHub with a custom set of demonstrations and lessons; or by developing their own lesson plan which borrows from and combines one or more demonstrations on CHEESEHub. The use of CHEESEHub only requires a web-browser and can hence be employed in a wide variety of educational and training settings from K-12 schools through university.**

***Index Terms*—cybersecurity, education, cloud computing, web platform, community**

## I. Introduction

The widespread adoption of digital devices and applications has permeated almost all aspects of our lives from communication, media consumption, banking, and travel to work. They have brought with them unprecedented advantages of instant access to information and services, increased productivity, and enhanced global connectivity and collaboration. This has further spurred the meteoric rise of new applications and services being developed by a constantly growing information technology (IT) workforce. At the same time, a significant amount of personal and privileged information collected by such applications is at increased risk of exploitation by unscrupulous actors who are constantly uncovering weaknesses in the underlying software. Thus cybersecurity has to be transformed from an academic specialization into a necessary skill that is imparted from the ground up through training and education to better prepare the entire IT workforce. Current initiatives and programs in cybersecurity education while on the rise, often fail to make the transition from theory to practice. This is not to say that hands-on training in cybersecurity does not exist; several such training platforms have been developed and demonstrated impact in the form of improved understanding of cybersecurity concepts by students. However, there is a high barrier to entry to using and adopting these platforms by a broader audience due to a lack of infrastructure and the technical know-how to manage and operate such infrastructure.

Our project – Cyber Human Ecosystem of Engaged Security Education (CHEESE) – seeks to address these shortcomings by providing a dynamic, open, web-based learning platform that lowers the barrier to access to a training environment while adopting a contribution process that enables rapid response to emerging trends and issues. Keeping pace with quickly developing trends in cybersecurity requires an active, community-driven approach whereby users can contribute demonstrations of, and where available, solutions to emerging and recently discovered exploits. CHEESE hopes to catalyze a community of researchers, educators, practitioners, and students around an open, web platform to accelerate this contribution process.

The rest of the paper is organized as follows: in section II we describe other existing platforms for cybersecurity education, comparing and contrasting them with CHEESE, in sections III and IV we describe the components of CHEESE, their design, deployment, and implementation status, in section V we de-

This work is funded by the U.S. National Science Foundation Award #1820573 and #1820608

scribe the innovative aspects of CHEESE, ending with planned future work in section VI.

## II. Background and Related Work

We describe a few key examples of prior efforts in hands-on cybersecurity education and training, illustrating how CHEESE either complements or improves upon these prior platforms.

### *A. TryCybSI*

TryCybSI [1] is a direct precursor to CHEESE and utilized the Amazon Web Services container orchestration framework, EC2 Container Service (ECS) to deploy 12 containerized demonstrations of cybersecurity concepts from network security, secure coding, cryptography, and other research projects. However, this is a static, curated platform with the project team contributing all applications, and provides no pathway to community contributions. Furthermore, the integral role of ECS implies that the platform cannot be easily ported to any other infrastructure (cloud or otherwise). CHEESE is designed with portability in mind and can be hosted on a wide range of target environments ranging from a laptop to commercial cloud. CHEESE also lays emphasis on a design that enables and encourages community contributions.

### *B. SEED Labs*

The SEED Labs [2] project has been a successful effort in imparting practical skills in cybersecurity concepts. Since its inception in 2002, the SEED project has developed over 30 labs that cover a range of fundamental topics in cybersecurity or well-known attacks. The low cost and ease of deployment has led to over a thousand institutions adopting the virtual labs developed by SEED. In addition to the labs themselves, a collection of textbooks on cybersecurity are also available that supplement the information in the labs. The non-trivial lab setup process (involving the download and setup of a virtual machine on a personal computer) however represents a non-trivial barrier to entry that still precludes all but tech-savvy users and instructors from utilizing these labs.

### *C. Labtainers*

Labtainers [3] is a set of over 40 containerized lab exercises that are designed primarily as student assignments. The platform also includes developers tools that can be used to develop new exercises. The framework is designed to support student evaluation through automatic collection of artifacts during a student's work on a container. The development environment for new exercises is feature-rich, providing a set of useful base container images and configuration files for setting up container networking and executing optional post-setup steps in a container. This custom development environment may, however, prove a steep barrier to entry for contributors who are more familiar with the networking and execution hooks available natively in containerization and deployment frameworks such as Docker and Kubernetes.

### *D. DETER*

The DETER project [4], [5] extends virtual machine-based labs to the cloud. A testbed with over 400 computers and resources are allocated free-of-cost to institutions based on the need of various security projects. DETER is primarily focused on providing resources for cybersecurity research rather than the development of cybersecurity curriculum and applications.

### *E. Virginia Cyber Range*

The Virginia Cyber Range [6] project is perhaps the closest to CHEESE in terms of goals and capabilities. It is a scalable, cloud-based infrastructure that provides students with virtual environments for hands-on training in cybersecurity. The Cyber Range provides a highly curated set of labs and exercises organized into course modules that can be used as is, or adapted by faculty for their needs. While Cyber Range does allow faculty contributions, the preference is for contributions of complete modules that include a set of lessons and associated labs. This may deter all but cybersecurity instructors from contributing to the platform. Furthermore, Cyber Range has thus far been restricted to institutions in Virginia, although a recent U.S. Cyber Range effort is working towards expansion to the entire country.

## III. CHEESE

We conceive CHEESE as an ecosystem comprised of three equally important and inter-related elements:

1) A web **platform**, CHEESEHub for hosting cybersecurity demonstrations that can be easily replicated, extended, and ported to a variety of hosting infrastructures.
2) An open set of online **lessons** that complement each hosted demonstration, describing the cybersecurity concept being demonstrated, instructions for following along, and associated learning objectives.
3) A **community** of users and contributors developed through **outreach** efforts to help grow the platform and obtain valuable feedback on usability and technical accuracy.

We describe the key (non-)technical objectives and challenges associated with each of these elements followed by our design and implementation addressing these objectives and challenges.

### *A. Platform*

The driving principle behind CHEESEHub is to provide a low-barrier platform for the hosting and usage of hands-on learning scenarios in different settings such as semester-long classroom instruction or short training events at workshops and conferences. Cybersecurity instruction spans a wide variety of concepts ranging from network security, database exploits, memory and buffer overflow attacks, to malware and phishing. Demonstrating such concepts requires a correspondingly wide range of infrastructures and software environments. Virtualization techniques such as virtual machines and containers are ideal for this purpose due to their minimal impact on the hosting infrastructure and the ability to create a wide variety of software environments, including operating system versions with known vulnerabilities. CHEESEHub is also intended for use by the broader IT community, K-12 educators, and general public. High profile exploits like the HeartBleed bug [7] have been featured heavily in the news and a demonstration of this bug would have more impact if it depicted the real-world implications of the exploit. For instance, the demonstration of HeartBleed on CHEESEHub illustrates how hackers can easily gain access to another user's authenticated session on a web application by extracting session cookies from the memory of an unsecure webserver.

The CHEESEHub platform utilizes containers due to their advantages over virtual machines: they are more resource efficient with the ability to host several containers on a single machine; are easier to develop, manage, and extend through a declarative template; and can provide a wide range of operating system versions without the concern of automatic updates patching software libraries with known exploits. However, containers have their own pitfalls. For instance, since the container shares the kernel of the host machine, kernel-level exploits cannot be demonstrated via a container; container networking simplifies connections between different containers running on the same host machine and may not allow for the simulation of certain real world scenarios that require non-trivial network topologies; and experiments that require real networking infrastructure such as routers or switches cannot be simulated via software in containers. One of the goals of CHEESE is to identify and evaluate the limits of container-based solutions in simulating various cybersecurity concepts.

### B. Lessons

We have developed and adhere to several guiding principles when designing lessons complementing the CHEESEHub applications: the lessons need to be self-contained so any user can learn both the underlying concept and replicate the demonstration; the lessons are categorized in a way that enables easy integration of various pieces into a curriculum; and contributors have a well-defined template for designing a new lesson to maintain consistency across lessons while simplifying the contribution process. A different concern when designing the lessons is the level of detail that is appropriate for a particular learning environment. When designing a lesson for classroom instruction, instructors may often prefer that lessons leave out certain information that the student is either expected to learn or recollect from the class. When designing a lesson for the general public, the converse is true; a higher level of detail helps retain interest and improve engagement with the material.

### C. Community and Outreach

CHEESE is intended to be a community-driven and sustained effort that provides training on not just existing cybersecurity concepts, but also emerging trends from active research. Furthermore, in order to ensure broad adoption of the platform by various institutions it is necessary that CHEESE provide a diverse set of applications that can be incorporated into the individual curricula at these institutions. Our goal is to create a broad ecosystem comprising instructors who see the value of the platform and contribute to it; researchers who use the platform to host reproducible cybersecurity applications to supplement publications; professionals who use the platform to facilitate quicker and broader dissemination of information and reproducible demonstrations of emerging exploits; and students who develop and contribute their own cybersecurity demonstrations to improve their understanding of the underlying concepts.

In order to ensure such community buy-in, it is necessary to reduce the barriers to contributions and conduct outreach efforts to various communities of researchers, instructors, and university IT security professionals to coalesce a broad community of stakeholders around CHEESE.

## IV. Design and Deployment

CHEESEHub[1] is intended to be a portable, scalable, and extensible open-source platform combining containerized training scenarios with a set of open lessons. This includes support for the following:

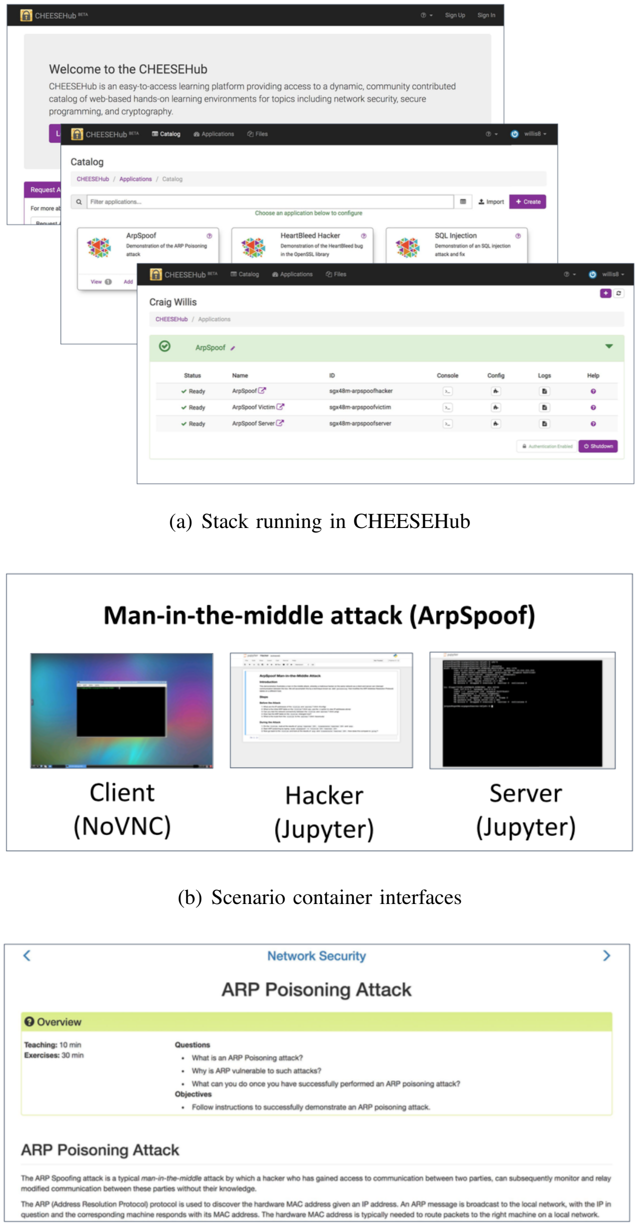
1) Ability to easily install and scale up/down the platform on academic and commercial cloud services
2) Ability to customize the catalog including the addition of community-contributed containerized environments and lessons
3) Ability to track usage and encourage contribution

Figure 1 demonstrates the CHEESEHub user experience for the ARP Poisoning Attack scenario (ArpSpoof). ArpSpoof consists of 1) a GitHub repository containing the container image definitions and the necessary demonstration software and user interface; 2) a set of JSON specifications in the CHEESEHub catalog that specify Docker container image references, resource limits, and networking specifications; and 3) an ”episode” in the Network Security lesson. A student following the lesson would launch the ArpSpoof scenario containers via CHEESEHub and be provided with access to the three separate environments: client (or victim) via NoVNC, server via Jupyter notebook, and the hacker also via Jupyter.

### A. Container-based Learning Environments in CHEESE

In training environments such as CHEESEHub, container-based solutions have proven effective for providing consistent environments with lower overhead than virtual machine-based approaches. Container orchestration frameworks such as Docker Swarm and Kubernetes implemented by cloud hosting providers have simplified the process of scaling computational resources to meet the demands of large classroom or user community settings. This, combined with easy-to-use, web-based interactive tools such as the Jupyter or RStudio, make it feasible to provide entry-level training scenarios at scale for computational materials [8]. For more advanced users, containers are more portable and easier to develop, test, and use locally. Furthermore, once developed and tested, a

[1] https://www.hub.cheesehub.org



Welcome to the CHEESEHub

CHEESEHub is an easy-to-access learning platform providing access to a dynamic, community contributed catalog of web-based hands-on learning environments for topics including network security, secure programming, and cryptography.

ArpSpoof — Demonstration of the ARP Poisoning attack

HeartBleed Hacker — Demonstration of the HeartBleed bug in the OpenSSL library

SQL Injection — Demonstration of an SQL injection attack and fix

| Status | Name | ID | Console | Config | Logs | Help |
|---|---|---|---|---|---|---|
| ✓ Ready | ArpSpoof | sgx48m-arpspoofhacker | | | | |
| ✓ Ready | ArpSpoof Victim | sgx48m-arpspoofvictim | | | | |
| ✓ Ready | ArpSpoof Server | sgx48m-arpspoofserver | | | | |

(a) Stack running in CHEESEHub

(b) Scenario container interfaces

ARP Poisoning Attack

Overview

**Teaching:** 10 min
**Exercises:** 30 min

**Questions**
- What is an ARP Poisoning attack?
- Why is ARP vulnerable to such attacks?
- What can you do once you have successfully performed an ARP poisoning attack?

**Objectives**
- Follow instructions to successfully demonstrate an ARP poisoning attack.

ARP Poisoning Attack

The ARP Spoofing attack is a typical *man-in-the-middle* attack by which a hacker who has gained access to communication between two parties, can subsequently monitor and relay modified communication between these parties without their knowledge.

The ARP (Address Resolution Protocol) protocol is used to discover the hardware MAC address given an IP address. An ARP message is broadcast to the local network, with the IP in question and the corresponding machine responds with its MAC address. The hardware MAC address is typically needed to route packets to the right machine on a local network.

(c) Lesson

Fig. 1. ArpSpoof scenario overview. Through the CHEESEHub system, users can easily start a 3-container hands-on scenario to actively demonstrate the client, server and hacker in a man-in-the-middle attack.

container can be immediately pushed to the central hosting repository, DockerHub with immediate access and guaranteed reproducibility for any users of the container.

Since CHEESEHub is a completely web-based platform, all applications hosted on CHEESEHub need to provide a web interface for their use. While this may seem restrictive, the availability of base container images for interactive computation platforms such as Jupyter notebook and complete web-accessible Linux desktop environments through VNC (virtual networking computing) allows for the hosting of the whole gamut of applications ranging from command line tools to complex JVM (Java Virtual Machine)-based desktop tools.

### *B. Orchestrating CHEESE Containers*

CHEESEHub leverages an existing container orchestration platform to help manage and orchestrate containers in response to user requests from the web interface. Labs Workbench [9] is an open-source platform designed to support turn-key deployment of containerized tools related to data management, analysis, and visualization in service of research data access, education, and training. It has been used to provide low-barrier access to large-scale research datasets using specialized software environments; as a scalable service for tutorials, workshops, and hackathons; and as an education platform to provide hands-on experience with computational tools and concepts.

Labs Workbench provides a thin abstraction over the Kubernetes orchestration framework with a Javascript web-accessible user interface, REST API, and service catalog. The catalog contains a set of customizable JSON specifications that define multi-component application ”stacks” – each referencing one or more published Docker images – that are translated into Kubernetes objects at runtime. Workbench also provides basic user registration, authentication, and authorization services. It is installable via Helm chart requiring only access to a Kubernetes cluster, wildcard DNS, and TLS certificates. Because of the ease-of-installation and scalability, the platform has been effectively used to support a variety workshop, training, and hackathon environments. CHEESEHub is currently deployed on Jetstream cloud [10] resources obtained through a research allocation with by the U.S XSEDE program [11].

The CHEESE project adapted the Labs Workbench platform for the CHEESEHub web-accessible interface and to support turn-key deployment of cybersecurity education scenarios. The CHEESEHub catalog contains a set of JSON specifications defining the various scenario application stacks that are launchable from the web interface. For example, the ArpSpoof scenario consists of three containers (hacker, victim, and server) all sharing a common network. When launched via CHEESEHub, each user has access to a private set of instances to complete lesson exercises. Privacy is enforced via the use of role based access control and Kubernetes namespaces.

The CHEESE project required several enhancements to Labs Workbench. Workbench was previously designed to work only on Kubernetes clusters deployed via the OpenStack cloud provider widely used in academic environments, but CHEESE required support also for commercial cloud such as Amazon Web Services (AWS) and Google Cloud for wider adoption. CHEESEHub scenarios required finer-grained control over privileges given to a container instance and the ability to co-locate containers on a host. Co-located containers are required for example in ArpSpoof where it is assumed that the victim, server, and hacker are on the same local network. While these capabilities were already available in Kubernetes, they had not been exposed via the Workbench API. Active development efforts are being undertaken to also extend Workbench to

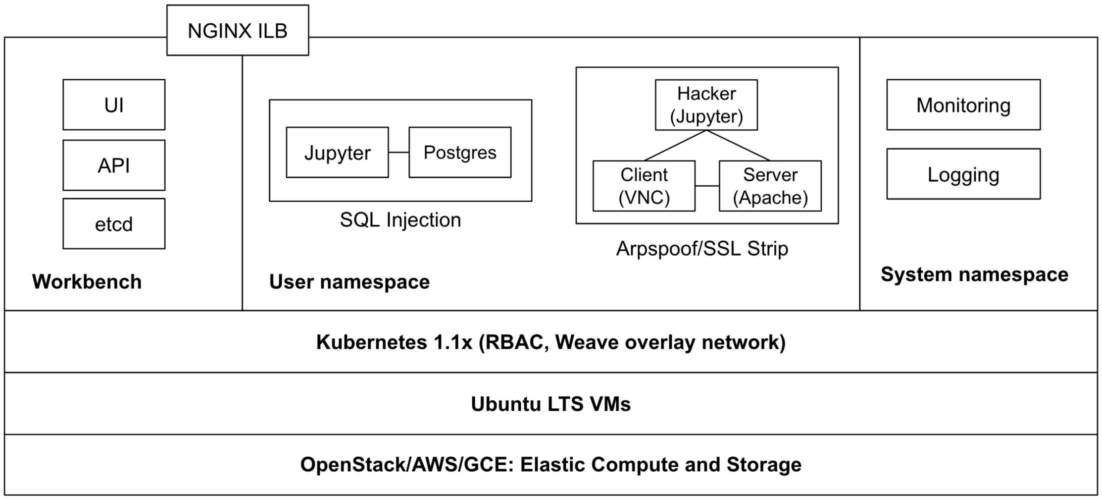


Fig. 2. CHEESEHub system architecture

facilitate improved usage tracking and reporting. Additionally, the Labs Workbench platform did not include any facility for integration with online learning materials.

### *C. Lessons*

Early in the project, we faced the question of how best to integrate lesson materials with the containerized scenarios available in CHEESEHub. The Labs Workbench platform had no specific support for integration with courseware, although it had been used in Carpentry-style workshops in the past. We considered the idea of developing an online textbook and lessons modeled after the successful ”Computational and Inferential Thinking” textbook used in the ”Data 8: Foundations of Data Science” developed at UC Berkeley[2]. However, the idea of developing a textbook seemed counter to the flexible and open-lesson goals of CHEESE. After some discussion, we decided to model our lessons after the Carpentries initiative [12]. The Carpentries develop open lesson materials for hands-on learning environments and have been adopted both in structured workshops and training events as well as a la carte into various instruction environments, including academic classrooms. Wilson (2016) reports several factors that contributed to the success and continued growth of the Carpentries initiative: the use of hands-on lessons with live coding, transparency, and open lesson materials.

Following the Carpentries model, we have developed a set of CHEESE lessons based on the Carpentries template maintained in GitHub and published as web-accessible GitHub pages to encourage open contribution, re-use, and customization. The lessons are designed using Markdown syntax which allows for rich text elements, callouts for emphasizing key points, and embedded images for application screenshots.

### *D. Building a Community around CHEESE*

To enable sustainability beyond the project period, one of the primary goals of the CHEESE project is to encourage community engagement, adoption, and contribution. While the original project plan included various social extensions to the Workbench platform to support user application requests, ratings, contributions, and reviews; it was later decided that such features were already available and widely used by community members in GitHub. GitHub is not just a source control management service, but also a social and collaborative platform. It was decided to utilize the existing features of this popular tool to reduce the barriers to community contribution. The contribution process for new demonstrations and lessons for CHEESEHub follows the standard GitHub *fork*, *pull request* and *issue* workflows to manage, review, and integrate community generated contributions. This pipeline has been utilized by both the project developers and student interns during the first year of the project.

Various outreach activities have also been conducted to publicize the CHEESE project through presentations at the Women in Cybersecurity conference and a training event at the annual ACM SIGITE, IT educators conference. Future outreach activities will target faculty involved in IT and cybersecurity education and the cybersecurity and information assurance staff at various university IT organizations.

### *E. Current Status*

CHEESEHub currently hosts nine applications and associated lessons demonstrating various cybersecurity concepts ranging across three categories: network security, secure coding principles, and cryptography. Some of the cryptography and secure coding applications are drawn from existing assignments in SEED Labs [2]. In these demonstrations, a key design goal is to also make the content accessible to lay-people who may not be familiar with the underlying IT

[2] http://data8.org/

concepts. For instance, in both the HeartBleed and ArpSpoof demonstrations, the real-world implications of the exploit are demonstrated by "hacking" privileged login information from a web browser session.

Through this project, we have also trained and mentored undergraduate students as part of summer internship programs at the authors' institutions. Two undergraduate summer interns were involved in the development of the containers and lessons for four applications ported from SEED Labs.

We also recently utilized the Science Gateway Community Institute (SGCI) [13] UX consulting service to conduct usability testing and analysis of the CHEESEHub platform. This process involved six undergraduate student participants and covered a wide range of usability and cognitive factors. Feedback obtained from the analysis will be utilized to make improvements to the CHEESEHub platform and lessons.

## V. Innovation and Impact

CHEESE embodies innovative practice in several key aspects of design and development, and education.

### A. Design and Development

CHEESEHub is to the authors' knowledge the only publicly accessible platform hosting demonstrations of cybersecurity concepts. While other similar cybersecurity education platforms have made their lesson material and containers or virtual machines publicly accessible, it is still the instructor/student/user's responsibility to install them on their own computing infrastructure. CHEESEHub also provides a user-friendly web interface through its use of Labs Workbench which signficantly lowers the barrier to creating a new user account and launching applications. The use of containers and an existing orchestration platform like Labs Workbench allows the project team to focus on developing the actual cybersecurity demonstrations rather than the deployment infrastructure and user interfaces.

By tying the contribution of lessons and applications together, CHEESE ensures that each application has associated instructions for reproduction and key learning objectives, while also encouraging contributors to consider the clarity and usability of their instructions. The use of the Carpentries model for lessons further lowers the barrier to contribution and lesson preparation. The Markdown syntax utilized in this model has a very low barrier to entry while also providing rich UI elements for incorporation in lessons. Furthermore, by using GitHub to manage lesson contributions, automated continuous integration/deployment (CI/CD) pipelines can be used to immediately publish the updated lessons to GitHub pages on new contributions or updates.

Labs Workbench leverages Kubernetes and Helm for container orchestration and can hence be deployed on a wide range of computing infrastructures ranging from a single laptop to academic and cloud computing resources. Furthermore, individual CHEESEHub installations are highly customizable. The application catalog for a CHEESEHub installation is specified using a configuration value in the Helm chart and can hence be modified to point to any GitHub repository of choice. Consequently instructors can quickly customize their CHEESEHub instance for a particular semester or course by simply modifying this configuration value and redeploying their installation.

### B. Education

The use of hands-on exercises for cybersecurity education has been shown to improve concept understanding and retention. In prior experiments with the *TryCybSI* [1] platform, we have observed better testing performance (measured as percentage of students answering correctly) on Bloom category 1,2,3 and 4 questions about the ARP Poisoning attack.

The CHEESEHub platform provides a collection of community driven learning scenarios that indulge learners with real-world experiential learning. For example, the HeartBleed attack container prepares learners with the knowledge that is vital for their future careers. From the pedagogical perspective, such learning experiences promote the learners to comprehend *what*, *when*, *how*, and *why* behind each learning module. Our pilot results show that it can greatly intrigue students' curiosities. More importantly, challenges can be brought into each lesson to cultivate students' adversarial thinking skills and system thinking skills.

## VI. Future Work

With the CHEESEHub platform and lesson structure in place, our next steps are to conduct various evaluation and outreach activities to foster a community of instructors and users around the platform and project. Following the usability testing and analysis, we have received feedback in the form of suggestions for user interface improvements that we will implement in the CHEESEHub platform and lesson structure. The authors plan to utilize CHEESEHub in classroom instruction activities in upcoming courses, workshop events, and summer training activities for K-12 students.

CHEESE is intended to be a community-driven project with the broader user community utilizing, evaluating, and contributing applications to the platform. As part of the community engagement efforts, we are working with faculty at various colleges and universities to solicit new applications for CHEESEHub, while providing them with a publicly accessible platform for use in their own instruction and training activities. We are also working to develop and improve documentation and related materials to encourage outside contributions.

## Acknowledgment

This work was funded by the U.S. National Science Foundation (NSF) collaborative research grants 1820573 and 1820608. We thank Noah Oller Smith and Takahide Iwai who developed cybersecurity demonstration applications for CHEESEHub as part of the Purdue University Discovery Undergraduate Research Initiative (DURI) summer internship program.

# REFERENCES

[1] B. J. Yang and B. Kirk, "Try-cybsi: A platform for trying out cybersecurity," *IEEE Security & Privacy*, vol. 14, no. 4, pp. 74–75, 2016.

[2] W. Du and R. Wang, "Seed: A suite of instructional laboratories for computer security education," *Journal on Educational Resources in Computing (JERIC)*, vol. 8, no. 1, p. 3:1–3:24, Mar 2008.

[3] M. F. Thompson and C. E. Irvine, "Individualizing cybersecurity lab exercises with labtainers," *IEEE Security Privacy*, vol. 16, no. 2, p. 91–95, Mar 2018.

[4] J. Mirkovic, T. V. Benzel, T. Faber, R. Braden, J. T. Wroclawski, and S. Schwab, "The deter project: Advancing the science of cyber security experimentation and test," in *2010 IEEE International Conference on Technologies for Homeland Security (HST)*. IEEE, Nov 2010, p. 1–7.

[5] J. Mirkovic and T. Benzel, "Teaching cybersecurity with deterlab," *IEEE Security Privacy*, vol. 10, no. 1, p. 73–76, Jan 2012.

[6] "Virginia cyber range," 2020, accessed 02/03/20. [Online]. Available: https://www.virginiacyberrange.org/

[7] "Heartbleed bug," accessed 04/11/20. [Online]. Available: https://heartbleed.com/

[8] C. Holdgraf, A. Culich, A. Rokem, F. Deniz, M. Alegro, and D. Ushizima, "Portable learning environments for hands-on computational instruction: Using container- and cloud-based technology to teach data science," in *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*, ser. PEARC17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3093338.3093370

[9] C. Willis, M. Lambert, K. McHenry, and C. Kirkpatrick, "Container-based analysis environments for low-barrier access to research data," in *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*, ser. PEARC17. Association for Computing Machinery, Jul 2017, p. 1–4. [Online]. Available: http://doi.org/10.1145/3093338.3104164

[10] C. A. Stewart, T. M. Cockerill, I. Foster, D. Hancock, N. Merchant, E. Skidmore, D. Stanzione, J. Taylor, S. Tuecke, G. Turner *et al.*, "Jetstream: a self-provisioned, scalable science and engineering cloud environment," in *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, 2015, pp. 1–8.

[11] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, "Xsede: Accelerating scientific discovery," *Computing in Science & Engineering*, vol. 16, no. 5, pp. 62–74, Sept.-Oct. 2014. [Online]. Available: doi.ieeecomputersociety.org/10.1109/MCSE.2014.80

[12] G. Wilson, "Software carpentry: lessons learned," *F1000Research*, vol. 3, p. 62, Jan 2016.

[13] K. A. Lawrence, M. Zentner, N. Wilkins-Diehr, J. A. Wernert, M. Pierce, S. Marru, and S. Michael, "Science gateways today and tomorrow: positive perspectives of nearly 5000 members of the research community," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 16, pp. 4252–4268, 2015.