CYBERSECURITY PEDAGOGY & PRACTICE JOURNAL

Volume 2, No. 2
September 2023
ISSN: 2832-1006

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Cybersecurity Pedagogy and Practice Journal** (**CPPJ**) is a double-blind peer-reviewed academic journal published by **ISCAP** (Information Systems and Computing Academic Professionals). Publishing frequency is two times per year. The first year of publication was 2022.

CPPJ is published online (https://cppj.info). Our sister publication, the proceedings of the ISCAP Conference (https://proc.iscap.info) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the ISCAP conference. At that point, papers are divided into award papers (top 15%), and other accepted proceedings papers. The other accepted proceedings papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the CPPJ journal.

While the primary path to journal publication is through the ISCAP conference, CPPJ does accept direct submissions at https://iscap.us/papers. Direct submissions are subjected to a double-blind peer review process, where reviewers do not know the names and affiliations of paper authors, and paper authors do not know the names and affiliations of reviewers. All submissions (articles, teaching tips, and teaching cases & notes) to the journal will be refereed by a rigorous evaluation process involving at least three blind reviews by qualified academic, industrial, or governmental computing professionals. Submissions will be judged not only on the suitability of the content but also on the readability and clarity of the prose.

Currently, the acceptance rate for the journal is under 35%.

Questions should be addressed to the editor at editorcppj@iscap.us or the publisher at publisher@iscap.us. Special thanks to members of ISCAP who perform the editorial and review processes for CPPJ.

2023 ISCAP Board of Directors

Jeff Cummings Univ of NC Wilmington President	Anthony Serapiglia Saint Vincent College Vice President	Eric Breimer Siena College Past President
Jennifer Breese Penn State University Director	Amy Connolly James Madison University Director	RJ Podeschi Millikin University Director/Treasurer
Michael Smith Georgia Institute of Technology Director/Secretary	David Woods Miami University (Ohio) Director	Jeffry Babb West Texas A&M University Director/Curricular Items Chair
Tom Janicki Univ of NC Wilmington Director/Meeting Facilitator	Paul Witman California Lutheran University Director/2023 Conf Chair	Xihui "Paul" Zhang University of North Alabama Director/JISE Editor

Copyright © 2023 by Information Systems and Computing Academic Professionals (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to editorcppg@iscap.us.

CYBERSECURITY PEDAGOGY AND PRACTICE JOURNAL

Editors

Anthony Serapiglia
Co-Editor
Saint Vincent College

Jeffrey Cummings
Co-Editor
University of North Carolina Wilmington

Thomas Janicki
Publisher
University of North Carolina Wilmington

2023 Review Board

Etezady Nooredin
Nova Southern University

Li-Jen Lester
Sam Houston State University

Jamie Pinchot
Robert Morris University

Samuel Sambasivam
Woodbury University

Kevin Slonka
Saint Francis University

Geoff Stoker
University of North Carolina Wilmington

Paul Wagner
University of Arizona

Paul Witman
California Lutheran University

Johnathan Yerby
Mercer University

Recommendations for Developing More Usable and Effective Hands-on Cybersecurity Education Materials Based on Critical Evaluation Criteria

Ahmed Ibrahim
aibrahim@pitt.edu
University of Pittsburgh
Pittsburgh, PA

Vitaly Ford
fordv@arcadia.edu
Arcadia University
Glenside, PA

Abstract

Effective cybersecurity education requires offering hands-on exercises in addition to lecture-based learning. The study provides insights into the challenges of cybersecurity hands-on education and offers a pathway for developing better cybersecurity educational materials. The fast-paced nature of the cybersecurity field makes it difficult for educators to keep up and create realistic exercises. Hence, there is a need for a common framework that would enable educators to produce more usable and effective cybersecurity hands-on educational resources, avoiding the reinvention of the proverbial wheel. In this work, we developed criteria to categorize and evaluate existing cybersecurity education resources based on their technical and educational characteristics. We analyze and evaluate four existing cybersecurity resources and provide critical remarks on their usability and effectiveness. Finally, we propose recommendations for developing new cybersecurity labs or exercises as well as designing cybersecurity platforms.

Keywords: cybersecurity, hands-on education, usability, effectiveness, evaluation, criteria.

1. INTRODUCTION

Providing effective cybersecurity education is challenging since it requires a lot of hands-on exercises, not just lecture-based learning. Additionally, developing students' competitive skills in cybersecurity puts educators in a position of creating (and often re-creating) a large variety of hands-on materials to keep up with the fast pace of this ever-growing field, which takes time and requires significant computational resources.

Many of the existing hands-on cybersecurity education resources exemplify high-quality learning materials and tools. However, there is no unified structure nor defined order allowing educators to use them seamlessly and coherently in their courses. Having a cybersecurity education framework for such resources would make it simpler to find and use what is needed in specific courses as well as determine the support structure required for efficiently and successfully deploying those resources in the classroom. As a result, we believe that educators should unite their efforts to design usable and effective cybersecurity education materials based on robust standardized criteria.

Developing and deploying real-life cybersecurity scenarios in academia take more time than regular assignments/projects for other types of courses. As educators design their assignments and labs, one element they should ponder about is how to minimize frustration and maximize

©2023 ISCAP (Information Systems and Computing Academic Professionals)

active learning. For instance, a considerable amount of time is dedicated to exercise setup and troubleshooting rather than spending time on actual learning. An educator may spend 30 minutes setting up a 5-minute attack demonstration. At the same time, if a lab requires a lot of setup or is somewhat vague, it may deter students from learning because it does not work as it should or does not work at all. Additionally, the fast-paced cybersecurity and technological landscape makes it infeasible for educators to keep up and develop realistic exercises for their students as it often requires them to revamp all exercises every semester. As a result, hands-on cybersecurity education is behind a lot of other STEM education areas.

In this study, we investigate known available resources/environments, assess them, share our experience based on their educational and technical pros/cons, provide recommendations for everyone, and then share our vision on what criteria need to be there for exercises and platforms to be effective. The primary goal of this project is to make cybersecurity exercises consistent and better. Critiquing the projects is part of our analysis that is not meant to downplay the importance and usefulness of the projects but is rather meant to help the reader understand the existing challenges with such projects.

2. OBJECTIVE AND METHODOLOGY

Our objective is to develop a framework for producing more *usable* and *effective* cybersecurity hands-on educational resources. In the context of this work, *usability* is mainly related to technical characteristics (e.g., customizable labs and lab access method) and *effectiveness* is mainly related to educational characteristics (e.g., clear instructional materials, learning objectives, and progress tracking).

To achieve our objective, we developed criteria (Appendix A) to categorize and evaluate (determining the instructional value) existing cybersecurity education resources. At the beginning of the criteria development process, we compiled a list of questions to ask before considering a new resource:

- Does the education resource provide enough support such that the instructor finds it easy to use in their class?
- Does it clearly state learning objectives?
- Does it provide supplemental material (e.g., network map)?
- Does it contain relevant content?
- Does it come with an instructor's manual?
- Does it have a grading rubric?
- Does it include an instructor answer key?
- Is it simple to deploy, run, and administrate exercises?
- Is it modular, allowing for mix and match / plug and play / not-sequential exercise completion? Does it allow instructors with different special areas to find what they need? Is each exercise/lab an independent unit and it does not necessarily depend on finishing the one before it? Does the project allow users to put the available exercises/labs in the order they need them to be rather than enforcing a particular order for using the labs?
- Does it challenge students to complete the exercise or merely hand-holds them to follow the step-by-step instructions?
- Is there a way to assess the student's learning?
- Does it provide the instructors with assessment tools/measures?

Based on these questions, we identified specific criteria metrics described in section 3. We then conducted a critical review of each of the resources' usability and effectiveness characteristics based on our experience and following the developed criteria. We used the collective knowledge from our experience to design the pathway for more usable and effective materials.

The rest of this paper is outlined as follows. In section 3, we list the developed evaluation criteria. Section 4 critically evaluates several existing environments according to the developed evaluation criteria. Section 5 discusses some of the other known cybersecurity projects that are not included in this study. Section 6 introduces the proposed recommendations. Section 7 presents the conclusion and the future direction of cybersecurity education resource development.

3. EVALUATION CRITERIA

For evaluation to be useful, it must be based on well-developed criteria. To accomplish that, we first introduce the following two categories: usability and effectiveness. Under each category, we list several criteria that are either quantitative or have a simple answer (e.g., yes/no). Each criterion will have a number, a label, possible answers, and a description. Due to the difficulty of objectively measuring qualitative criteria without introducing an opinionated bias, we minimized the number of qualitative criteria. However, more qualitative criteria will be

introduced in a later work after polling the broader community as will be mentioned in the Future Directions section.

Usability Criteria

In this subsection, we list four criteria developed under the usability category: (C1) type of labs, (C2) customization possibility, (C3) access method (in the form of 3 sub-criterion: C3-A, C3-B, and C3-C), and (C4) level of support.

Number: C1
Label: Type of labs
Possible Answers: Stand-alone / Connected machines / Both
Description: This criterion lists the type of labs available on a provided platform. The "Stand-alone" type describes the exercises/labs in the platform that can be executed on a single operating system without any need to connect to another OS using network communication. The lab may be done on a virtual machine or may represent a set of instructions that a student can perform on their own computer. The "Connected-machines" type implies that labs require the involvement of at least two computer/virtual machines and a computer network. The machines required can be either hosted on the cloud and available online or downloadable virtual machines with setup instructions. The "Both" type implies that the project offers "Stand-alone" and "Connected-machines" labs.

Number: C2
Label: Customizable Labs
Possible Answers: Yes / No
Description: This criterion lists whether a project enables instructors to customize the lab environment according to the instructor's needs or not. In the case of "Stand-alone" labs, this may be adding, editing, or removing services (either provided by the project or the instructor) in any given system. In the case of "Connected-machines" labs, this may be adding, editing, or removing virtual machines (either provided by the project or the instructor) on the network.

Number: C3-A
Label: Cloud-based (the whole education resource is accessible through a web browser)
Possible Answers: Yes / No
Description: This criterion lists whether a project enables instructors to access and use the education resource through a web browser or not. "Yes" means using the resource does not require downloading any virtual machines or software by instructors or students. "No" means the project requires some sort of downloading, configuring, or installing software that is dependent on some specified requirements (e.g., hardware or operating system).

Number: C3-B
Label: Lab access (for cloud-based projects)
Possible Answers: SSH Only / Web Interface Only / Both
Description: This criterion lists whether the labs are accessible via SSH (terminal), a web browser, or both.

Number: C3-C
Label: Setup guidelines (for downloadable material)
Possible Answers: Yes / No
Description: This criterion lists whether projects providing the downloadable material include directions and guidelines pertaining to the setup process.

Number: C4
Label: Level of support
Possible Answers: Institutional / Individual / None
Description: This criterion lists the level of support provided to instructors and students by the project. An "institutional" support implies that the project has some form of a ticketing/helpdesk system to report problems, ask questions, and get support (e.g., in case of network failure, environment not being available, or account problems). An "individual" support implies that the project is supported by a single individual via email or a form fill-out. None implies that there is no clear way for instructors and students to ask questions or get support.

Effectiveness Criteria

In this subsection, we list eight criteria developed under the effectiveness category: (C5) instructor's manual availability, (C6) student instructions availability, (C7) includes learning objectives, (C8) mapping to frameworks, (C9) limitations, (C10) progress tracking, (C11) time tracking, and (C12) accessibility level.

Number: C5
Label: Instructor's Manual Availability
Possible Answers: Yes / No / Partial
Description: This criterion identifies if a project provides an instructor's manual to help instructors understand and prepare for the material (i.e., exercises or labs). "Yes" means that all exercises have instructor's manuals. "No" means that none of the materials have instructor's manuals. "Partial" means that some, but not all, of the material has instructor's manuals or that instructor's manuals are partially incomplete.

Number: C6
Label: Student Instructions Availability
Possible Answers: Yes / No / Partial
Description: This criterion lists if a project provides instructions for students on how to walk through the material. The instructions can be either step-by-step or general guidelines on how to complete the labs. "Yes" means that all labs have student instructions. "No" means that none of the materials have student instructions. "Partial" means that some, but not all, of the materials have student instructions.

Number: C7
Label: Includes Learning Objectives
Possible Answers: Yes / No / Partial
Description: This criteria lists if the project materials include clear learning objectives. "Yes" means that all materials have learning objectives. "No" means that none of the materials have learning objectives. "Partial" means that some, but not all, of the material have learning objectives.

Number: C8
Label: Mapping to Frameworks
Possible Answers: NICE KSAs / CAE KUs / Both / None
Description: This criterion lists the cybersecurity educational frameworks which the project uses for mapping its materials. At the time of writing this article, the NICE KSAs (Petersen et al., 2020) and CAE KUs (CAE Documents Library, n.d.) are the two widely adopted cybersecurity education frameworks.

Number: C9
Label: Limitations
Possible Answers: Resources / Time / Both / None
Description: This criterion identifies the project's limitations. The "Resources" limitation can be a limit on the number of students running an exercise at any given time, a capacity per account/course (e.g., a lab may state that no more than 10 users can have access at the same time or only 16 machines are available for provisioning), or the necessity to have students download and install/import one or more virtual machines. The "Time" limitation can be a limit on the period of usage (e.g., resources are only available for two days).

Number: C10
Label: Progress Tracking
Possible Answers: Yes / No
Description: This criterion identifies projects that track students' progress on the assignment and allows instructors to view it.

Number: C11
Label: Time Tracking
Possible Answers: System / Lab / Both / None
Description: This criterion lists projects that track students' time and allow instructors to view it. The "System" time tracking implies that the project tracks the total time students have spent on the platform. The "Lab" time tracking implies that the project tracks the time students have spent on a specific lab.

Number: C12
Label: Accessibility Level
Possible Answers: Nationwide / Limited / Paid
Description: This criterion identifies the accessibility level of a project. The "Nationwide" level means it is accessible (free of charge) to any educational institution in the United States. The "Limited" level means it is accessible (free of charge) to a certain population (e.g., only Virginia State institutions). The "Paid" level means it is accessible for anyone who pays a fee (varying by the material provider).

4. APPLYING THE CRITERIA THROUGH CRITICAL EVALUATION

One of the struggles that educators face is developing labs and exercises that would not hand-hold students but instead would promote discovery and self-learning and open the opportunity to make mistakes without affecting grades. Cybersecurity is a field where technical knowledge is closely interleaved with theoretical foundations. Thus, it is challenging to determine a balance between how much information is enough and how much information is too little or too much for students to complete an exercise and facilitate learning.

In this section, we analyze and evaluate four existing cybersecurity resources (based on the evaluation criteria defined in the previous section) providing critical remarks on their usability and effectiveness. We realize that it is challenging to measure the materials' value, hence, we follow up with a discussion and recommendations section about educational and technical pros/cons according to our experience. In Appendix A, we include a table that shows which criteria are included in each of the four major hands-on educational resources evaluated in this work.

DETERLab

DETERLab (DETER Project, n.d.) (Mirkovic & Benzel, 2012) is a cluster environment focusing on allowing researchers and instructors to deploy cybersecurity experiments with custom network configurations to investigate cyber attacks and

defenses. DETERLab aims to provide an active learning scalable platform, a large number of computing resources, exercise setup automation, as well as access to reusable and modular experiments. In 2012, DETERLab was used by over 47 universities and colleges and had more than 400 general-purpose computing nodes (Mirkovic & Benzel, 2012). As of December 2016, DETERLab users have created 192 projects for their classes and DETERLab has served 13,000 students (DETERLab, n.d.). We decided to evaluate DETERLab's usability and effectiveness in educational settings, providing practical recommendations for improvement, because we believe that the platform has the potential and clear direction if it is made more adoptable.

Personal Experience
The first major struggle we faced was related to a confusing, outdated user interface (UI). It was challenging to navigate the platform, find where and how to start, and figure out how to add experiments to our project/class. Tabs on the homepage were not consistent. We did not see an "Experiments" tab until we figured out how to add our first experiment -- and only then did a new tab called "Experiments" appear on the instructor's page. It was difficult to find an answer to our questions using the site's Wiki since the search feature was not giving the correct results. Eventually, we had to use an external search engine to actually look things up on the Wiki. Overall, the Wiki did not seem to be written with the end-user in mind.

Labs are accessible using nested SSH connections to connect to the remote hosts which makes navigation between the hosts on the network confusing. The upper right corner of the DETERLab website showed the number of available PCs out of 691 PCs in total. The number of PCs freely available for deployment has been very low on a daily basis during the spring of 2020 (under 100) which poses a serious scalability challenge. During the spring of 2022 we found out that the total number of available PCs went down from 691 to 360 due to the retirement of the Berkley DETERLab site administrator (Figure 1).

Figure 1 DETERLab Berkeley nodes are down

The DETERLab project allows for lab customization (DETERLab's custom NS syntax) but there is a significant learning curve associated with the customization process. The DETERLab project has a simple FAQ page, a wiki, and a ticketing system. Based on their publicly available ticket information (DETERLab Ticket System, n.d.), some tickets are addressed within days while others may require follow-ups and take months.

More information about the DETERLab experience, the sign-up process, steps required, etc. can be found here (Ibrahim & Ford, 2021).

Recommendations
We recommend the DETERLab project to restructure and redo the Wiki, make the website usable and user-friendly, make homework and teacher manuals consistent, and provide training videos/classes for instructors. In addition, we think that offering an online training module (e.g., tutorial videos) to use DETERLab for all new users and publishing user reviews provides transparency and makes the material reliable, hence improving the material over time. And for those interested in using DETERLab we recommend expecting to put in significant effort and spend a considerable amount of time familiarizing yourself with how things work.

NICE Challenge
The NICE Challenge Project (NICE Challenge, n.d.) allows educators to use real-world virtualized business environments to teach cybersecurity. It contains over 100 different labs including defense, offense, server administration, configuration, setup, auditing, logs, malware, and other topics. At the time of writing this work, the NICE Challenge served more than 550 institutions and 1,000 faculty with 3,000 virtual machines/day and more than 150,000 total workspaces deployed. All NICE challenges are mapped against the NICE Framework's KSAs (Petersen et al., 2020) and CAE KUs (CAE Documents Library, n.d.). The project is available at no cost to educational institutions, provides training for new educators, has a support portal, requires only a web browser, and has a ticketing system to provide feedback and request support. The NICE Challenge Project evaluated according to the Usability and Effectiveness Criteria can be found in Appendix A.

The NICE Challenge project has a strategy guide that only includes a short explanation of the objectives and does not include any instructions on how to perform the tasks or reach the goal. From an instructional perspective, it does not provide the instructors with any useful information to help them be prepared for the challenge. The NICE Challenge project provides students with a narrative-driven scenario, a workspace, and a set of technical objectives

and/or a written deliverable, but it does not tell students how to complete the challenge and reach the technical objectives and/or a written deliverable.

The NICE Challenge project does not allow challenges to be available immediately. Instructors (a.k.a. curators) have to reserve pods for their students. Each reservation is limited to two consecutive days. And, each instructor has a limited number of seats to use for the reservations. Reservations must be requested by the instructor at least one day prior to the beginning of the required reservation day. In some cases, some days are not available due to insufficient workstation availability as shown in Figure 2.

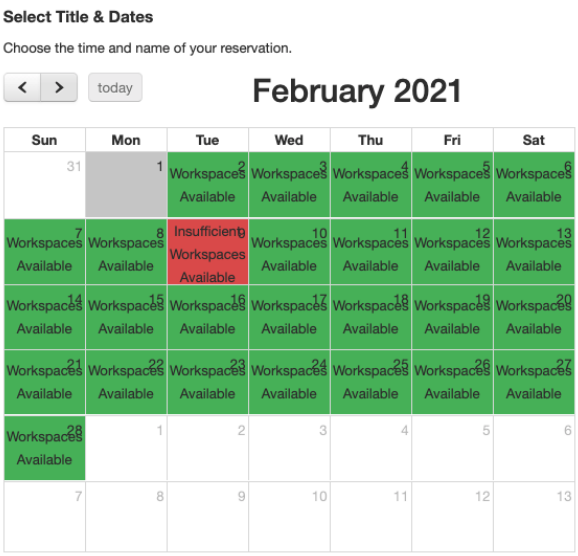

Figure 2 NICE Challenge workspace availability

The NICE Challenge project has an FAQ section and a helpdesk. Based on our experience, they respond very quickly (within 24 hours) to new tickets submitted through the helpdesk site.

The NICE Challenge project does not track the students' progress per challenge. However, it divides challenges into multiple checkpoints and it has automatic triggers to know whether a student was successful to reach the end goal for each checkpoint or not. It does not track the students' actions from when they start working on the checkpoint until they encounter an issue or complete the challenge. The NICE Challenge project provides information about the amount of time it took the student to complete each challenge checkpoint (Figure 3).

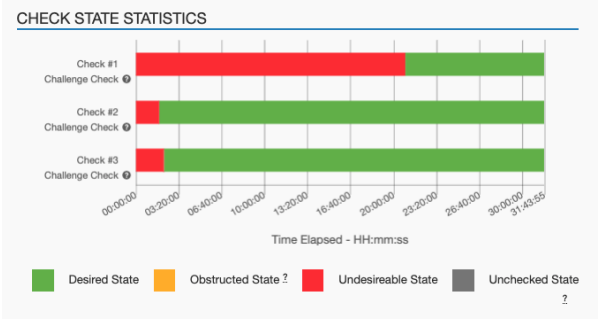

Figure 3 NICE Challenge student statistics

Personal Experience

The account application process is straightforward. The application is available on the main webpage and requires the instructor's name, EDU email, and their plan to use the challenges. When we applied to use the NICE Challenge platform, our application got reviewed and approved within a couple of days. It was simple to find out how to create a reservation. Once a reservation was active, it was easy to begin a lab and try it out.

Before using the challenges in the classroom, we had to go through the labs ourselves as there are no instructor's manuals for any lab. Most of the time, we experienced roadblocks and we had to reach out to the support team for clarifications. The support team promptly responded and our issues were resolved in a timely manner, even during weekends. After our confusion was resolved, it was straightforward to adopt the labs and provide hints and instructions to our students. We used it in the classroom and to host a cybersecurity awareness month competition. Students who participated in the competition expressed that they especially enjoyed the real-world, scenario-based experience.

There are many advantages to using the NICE Challenge project. It has a large variety of labs that helps educators find the right fit on a specific topic for their students to experience the practical application of theoretical knowledge that is often covered in cybersecurity courses. The labs are simple to deploy, provide an engaging scenario for students while they are waiting for the virtual environment deployment, and can be used by all students at the same time since the reservation is already made. A demo challenge is available that helps students understand how the platform operates and how to submit a challenge. In addition, the instructor can access students' deployments (GUI in the browser) and look at their machines which is especially beneficial when troubleshooting or providing support to students. It can be used for developing skills within a course

and to train/prepare students for attack and defense competitions. Instructors can also submit a "Challenge/Feature Request" through the project's ticketing system.

The NICE Challenge project also has some limitations that instructors should be aware of. Instructors cannot add students to a reservation if it has already started. A student cannot deploy more than one challenge at a time. The lab reservations are limited to two consecutive days, and the instructors have a limited number of seat credits available to use for reservations. An instructor's manual is not available, making it challenging for instructors to support their students when they encounter difficulties. Also, some challenges may be harder than they seem to be and some may not be working correctly. Thus, instructors must go through the challenges and ensure the "checks" (objectives) work correctly before assigning any of the challenges to their students. Finally, if an instructor has a teaching assistant (TA) added to the portal as an overseer to grade submissions, the TA must be the one who creates the reservations in order to be able to view student submissions. The instructor cannot allow an overseer (e.g., TA) to view submissions made for reservations that were created by the instructor.

Recommendations

We hope that in the future the NICE Challenge would allow educators to reserve available pods for more than just 2 consecutive days. Also, it would be beneficial to have a mechanism to generate a custom environment. A repository of challenge instructions would help educators in adopting the challenges and assisting students when they get stuck. Consequently, hints would be a useful feature to add. Instructors should have the option to allow TAs to view any reservation created at any point as well as submissions for reservations created before the TA was assigned to the class.

SEED Labs

The SEED project (SEED Labs, n.d.) started in 2002 by Kevin Du and has been growing since then. The SEED project's objective is to develop hands-on laboratory exercises (called SEED labs) for computer and information security education and help instructors adopt these labs in their curricula. As of 2021, the project has been funded by a total of 1.3 million dollars from NSF, and is now used by over a thousand educational institutes worldwide.

The SEED project consists of four main elements: Labs, Books, Lectures, and Workshops which cover topics such as computer and information security, cryptography, software security, network security, web security, operating system security, and mobile app security.

The *labs* are hosted as downloadable virtual machines that instructors and students would deploy themselves on their local computers. Lately, there has been an effort to make virtual machines available on cloud platforms (e.g., AWS, Google Cloud). However, the instructors would be responsible for any associated fees to host and operate the virtual machines or they would need to join a cloud-based Educate Program (e.g., AWS Educate) providing free credits for students and instructors. Additionally, instructors can receive lab manuals via email after providing evidence that they are the instructors of the course where the labs are going to be incorporated.

The *books* cover computer/Internet security topics and include problem sets associated with the hands-on labs allowing students to practice and learn both theoretical and practical cybersecurity paradigms. Slides and solution manuals are freely available in an electronic format for instructors upon request.

The *lectures* are recorded on the Udemy platform as two separate courses, namely Computer Security and Internet Security, and can be accessed for a fee. As of the time of writing this article, the SEED website provides a Udemy coupon to receive discounted access to the recorded lectures.

The *workshops* provide training to instructors who are interested in using SEED labs in their courses. They have been offered annually every summer since 2015, free of charge to accepted instructors. The SEED labs project according to the Usability and Effectiveness Criteria can be found in Appendix A.

Personal Experience

Downloading the SEED virtual machines and adopting the labs is simple and available for anyone to use under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. The project provides detailed setup instructions on how to get started, download, import, and configure the SEED virtual machine. It is possible to customize the labs but instructors would need to do it on their own without the involvement of anyone from the SEED Project team. The lab manuals are made with the end-user in mind, assuming zero previous knowledge. The labs' documentation is very

illustrative, including screenshots and code snippets describing every configuration step in detail, a troubleshooting section, and a guide on related topics. Additionally, we found it beneficial that each lab includes suggested supervised and unsupervised times that students should typically spend doing the lab.

SEED labs (version 2.0) provide 9 software security, 10 network security, 3 web security, 2 hardware security, 8 cryptography, and 2 mobile security labs. Some of the labs (e.g., web security) can be done using only one copy of the SEED virtual machine, whereas other labs require cloning the VM to one or two additional copies in order to go through the lab. The instructors should ensure beforehand that students' hardware supports virtualization and has enough hard disk space (at least 10 GB per VM) and RAM (at least 2 GB of RAM available per VM). In regards to support, we received timely answers to our questions from the PI of the SEED project.

Recommendations
The SEED labs are very stable and are being used by many institutions all over the world. One major topic we recommend everyone investigate is finding a simple way to host the SEED VM on the cloud to minimize the overhead of requiring one VM per student. For instance, the SQL injection lab can be done by hosting a centralized version of the VM and all students need to only use the browser to do the lab rather than having all students install and download the VM to do that lab. Lately, the SEED project started providing instructions on how to deploy the VMs in commercial cloud systems.

Another important topic is creating a web-portal (forum) for instructors and users to connect and share knowledge. Since the SEED project is individually supported, developing an online forum/blog can be helpful to connect all those who use the SEED labs. Instructors who adopt the SEED labs may need to restructure and clarify the submission guidelines and deliverables that students need to follow as well as include a grading rubric, if necessary. Additionally, we believe that mapping the labs to the major cybersecurity education frameworks would be of high benefit to the instructors.

EDU Range
EDURange (Boesen et al., 2014) is an NSF-funded project "providing hands-on exercises, a student-staffed help-desk, and webinars". Initially, EDURange was a cloud-based platform hosted on AWS. But currently, EDURange is no longer a cloud-based project and the code is provided for instructors to host it on their own cloud or servers. The exercises' goal is to allow faculty with little prior background to teach security and increase the number of schools teaching cybersecurity concepts. The gamified exercises (called *scenarios*) are open-source and available on GitHub. Based on their project's website, there are 8 scenarios currently available on the platform, namely: getting started, file wrangler, SSH inception, total recon, strace, ELF infection, treasure hunt, and metasploitable. These scenarios go over learning about the basics of using the Linux command line, permission loopholes in Linux, nested SSH, executable file examination using strace, SQL and XSS injections on a web app, nmap scanning, files and directories in Linux, basics of metasploit on a widely used image of Metasploitable2 (Linux-based), and infected binaries.

Personal Experience
When we started investigating how to use the EDURange project, we found that a server is required to host the scenarios but there are no server specifications provided. Thus, we created an Ubuntu 20.04 Virtual Machine with 6 GB RAM, 2 CPU cores, and 16 GB of storage on an ESXi virtualization server. To deploy the environment on the server we used the commands in the EDURange GitHub repo which went well with a few hiccups. After our first failed deployment attempt, we contacted EDURange support about our deployment process and they updated the repo with commands that worked the second time we attempted the deployment process. Afterward, we were able to create one administrative account and two student accounts on the platform for testing. However, we encountered issues with the lack of hardware specifications. The first scenario we chose to deploy was Metasploitable which constantly failed to deploy. After tracing the terminal output, we found that the scenario was not progressing. Given that we knew that Metasploitable requires a large storage space, we checked the remaining storage space to find out that we ran out of space on the server. We had to extend the disk storage from 16 GB to 40 GB and then redeploy the scenario.

It is of major importance to set up the EDURange *.env* file correctly from the first time. That file will include the hosting server's hostname or IP address which will be used in every scenario to be deployed. When we set up the *.env* file incorrectly during our first deployment, there was no available option to update the values after the server was started. To resolve the error resulting from a misconfiguration in the *.env* file, we had

to completely destroy the first deployment and start a totally new one with the correct configuration.

Another issue we encountered is that student accounts cannot access a scenario if the scenario was created prior to the student registrations. Hence, students must already be in the system before the scenario is assigned to the students' group and later deployed. Otherwise, the students would see an error message if they try to join that scenario even after rebooting the server.

An additional challenge we encountered is that a scenario may not deploy correctly if the name includes a space in it. Some scenarios (e.g., Elf, strace, and WebFu) did not deploy at all or were deployed with errors at the time of writing this work.

Almost every scenario has some inconsistency in it. For example, in the File Wrangler scenario, the task numbering in the student guide does not align with the actual numbering of the questions that the students should answer (e.g., the guide for task #4 will cover question #4 and #5, then task #5 will cover question #6) which can be confusing. In addition, all the steps in the task guide were numbered as "1" instead of having a sequential numbering. We experienced many inconsistent or wrong formatting in different scenarios. Also, student guidelines are not complete and do not map properly to the task(s) required to finish a scenario.

Another example of an inconsistency is that question #6 in the Getting Started scenario asks for six file names of image files. However, there is only one textbox to enter the answers. The question mentions entering each filename separately. We found this confusing and it would have been better if the environment included an unambiguous way to accept the answers.

In addition, the order of the deployed scenarios displayed in a table format (which includes the buttons to start/stop/destroy a scenario) on the administrative dashboard keeps changing in real-time, making it difficult for instructors to be sure they click the correct button. It has happened a couple of times that when we meant to click "stop" for a particular scenario, it actually stopped another scenario.

In an attempt to offer the SSH Inception scenario to students, we developed our own set of questions in an exercise using our Learning Management System. In the exercise instructions, we included the local IP addresses that students need to use according to the SSH Inception scenario. When we destroyed and created a new SSH Inception scenario for a new group of students, the IP addresses in the new scenario were different from the first one. This means that we had to take new screenshots and update the IP addresses we used in our exercise to match the newly deployed scenario. Instructors should expect that every new deployment of an exercise will have a different set of IP addresses associated with the scenario.

Recommendations

The first recommendation is to have a clear starting point for how to set up the EDURange server since the "Guides" tab on the EDURange website (visited on 04/05/2023) points to outdated instructions. For example, when clicking on the "An instructor EDURange installation guide" link (in Figure 4), it navigates to a deprecated GitHub repo (Figure 5). Also, there are two websites that host the EDURange information, guides, etc. (edurange.org and edurange.github.io), and the latter is an outdated version.

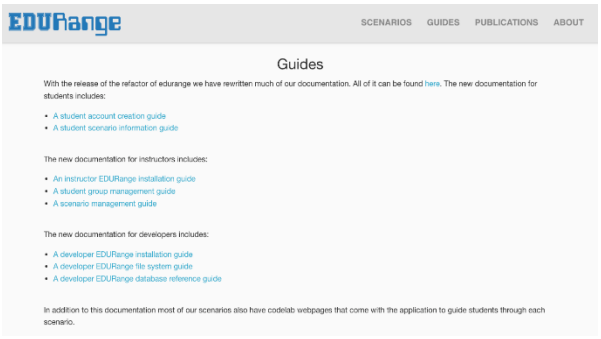

Figure 4 EDURange guides

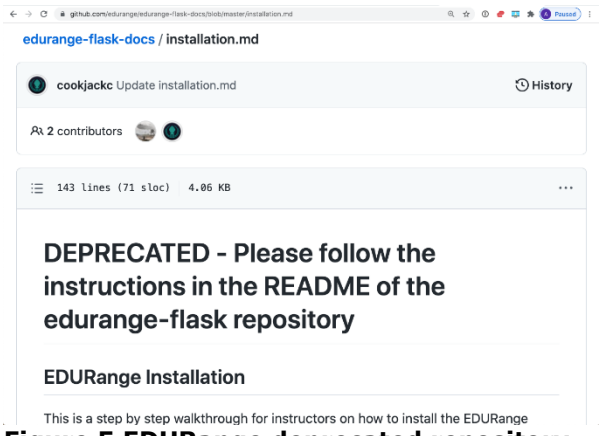

Figure 5 EDURange deprecated repository

After navigating the EDURange's main GitHub, we found the correct repository which is the one we

used for setting up and running the EDURange server. We highly recommend pointing to that GitHub repository directly from the EDURange website to eliminate any confusion or trouble on the instructors' end.

We also recommend adding a "reset scenario" button for the whole class as well as separately for every student to address cases when students accidentally make irreversible changes to the environment preventing further progression through the assignment.

For those interested in using the EDURange project, keep in mind that you need to plan your hardware requirements according to the number of scenarios you will deploy and the number of students you will engage. If EDURange developers include example estimates for hardware requirements needed to support students in classrooms, that can help instructors plan accordingly.

There are two ways to create student accounts. Admins can create a "Group" which generates a "Registration Code" to share with students. Each student then has to create their own account with that Registration code. Note that if a student registers after the admin builds a scenario to the "Group", the student will not be able to access the scenario (an error will be shown when the student tries to open the scenario). Therefore, the instructors need to ensure that all students are registered before assigning a scenario to the "Group".

Alternatively, the admin can pre-populate the Group with a number of Temporary Members. This option gives the admin a standardized usernames and random passwords list which the admin can pass on to their students. However, if students use these usernames and passwords, they cannot change their usernames to their real names which can become a challenge during grading. There should be a way to import/export group members from an Excel/CSV file. The export function could provide the details of students' performance in the labs. The import function can allow the automated creation of user accounts, using students' real names/email addresses.

When an administrator creates a scenario, the only way to know which group it was created for is to go to the "Command History" tab inside the scenario and look for the player names and then find out the group they belong to from the administrator dashboard. For the users of EDURange, when creating a scenario, we recommend typing the GROUP NAME as part of the Scenario name because there's no easy way to know which group a scenario belongs to.

When building a scenario, the terminal will show information about the time elapsed for some scenarios (e.g., Metasploitable), but the web interface will not display any information about the progress of building a scenario. We recommend displaying information about the progress of building a scenario on the instructor dashboard. Also, indicating the disk space used by a scenario is of vital benefit in managing resources. Thus, we recommend adding the disk space used by each scenario on the instructor dashboard (e.g., Metasploitable requires about 7GB of disk space). The information about the required storage for all scenarios would allow the instructors to ensure that they have allocated enough disk space for the machine. Thus, it would also be useful to see how much disk space is left in total on the instructor dashboard. We recommend that the platform does not allow the instructor to create a scenario unless there is enough disk space available, displaying an appropriate notification message.

The dashboard could also benefit from an option for the instructors, notifying them when there is an update of the platform available on GitHub. As of now, there is no way to check for updates on new scenarios or incomplete scenarios.

It would be beneficial to provide an instructor Answer Key. Currently, instructors can find the correct answers to the questions by creating a test student account, going through the scenario tasks, and finding the answers themselves. Or, the test student account can be used to type any kind of an answer; then the instructor can go to the admin dashboard to see the correct answers for the test student user account trials. A separate answer key will also eliminate the highly unlikely event of a student running an EDURange server and finding the embedded answers themselves.

The answers to all scenarios are publicly available on the GitHub repository (meaning that students could potentially be able to find it), for example, the source for the SSH Inception scenario can be found here:
https://github.com/edurange/edurange-flask/blob/master/scenarios/prod/ssh_inception/questions.yml

The EDURange playground should allow instructors to specify the IP addresses range they need to use when deploying a scenario. If that IP range is not already in use by another scenario,

the IP range should be accepted to be deployed. Otherwise, instructors should be notified that the IP range they chose is currently in use and should be able to change their choice.

5. OTHER PROJECTS NOT INCLUDED IN THIS STUDY

In this section we list other projects we hoped to assess but could not do so for different reasons.

GENI CyberPaths (Mountrouidou, 2019) included a variety of network-related exercises, such as DoS attacks, covert storage channels, and intrusion detection systems. However, the project is no longer maintained and most of the exercises are not possible to run. The SecKnitKit project (Siraj, Ghafoor, Tower, & Haynes, 2014), funded by the NSF, offers a VirtualBox standalone virtual machine to cover four different security areas: Network, Software Engineering, Operating Systems, and Database Management. It is important to note that this project is of a smaller size than the other projects included in this study. At the same time, the topics that the exercises cover could be introduced in non-security-related courses. The project is no longer maintained and is accessible for downloading from the CLARK platform (Taylor, Kaza, & Zaleppa, 2021).

There are some state-sponsored cyber ranges established in the US like the Michigan Cyber Range, Florida Cyber Range, and Virginia Cyber Range (Priyadarshini, 2018). For instance, the Virginia Cyber Range is a cloud-hosted infrastructure with hands-on cybersecurity labs, modules, and courseware repository that maps to the NICE Framework KSAs (Petersen, Santos, Smith, Wetzel, & Witte, 2020) and CAE KUs (CAE Documents Library, n.d.). The materials are freely available for Virginia State high schools and colleges that meet eligibility criteria. The same material is also available nationwide but under the name of the US Cyber Range. The US Cyber Range has a pricing model that is dependent upon the class enrollment and the number of months that the students plan to use the cyber range for.

There are several other paid platforms similar to the US CyberRange that we have not included here. Additionally, there are cyber ranges that have a limited availability scope, such as Cyber.org Range (n.d.) which is only accessible for K-12 schools.

6. RECOMMENDATIONS

Hands-on cybersecurity learning depends on two main elements: *effective exercises* and *usable platforms* hosting such exercises. In this section, we propose recommendations for developing new cybersecurity labs or exercises (following an evidence-based learning approach) as well as designing cybersecurity platforms.

According to our experience, we noticed that some students would follow all the steps in an exercise but would not be able to put all the exercise pieces together, therefore they would not fully understand the purpose of what they were doing. They would complete the exercises and pass the class but it would become a waste of time as no effective knowledge transfer had occurred. Thus, the exercise development process (as recommended in the following Exercise Development subsection) is key in making sure that knowledge transfer is effective for students. Additionally, without doubt, the usability characteristics of cybersecurity platforms (as recommended in the following Platform Development subsection) directly impact the use of effective exercises.

Exercise Development Recommendations

At the beginning of the exercise development, educators should clearly define what they want students to learn (learning objectives) so that they can evaluate it based on known frameworks such as the CAE Knowledge Units (KUs). Exercises should state what students need to know (prerequisite knowledge) and what they should be able to do using that knowledge (e.g., actionable outcomes). Educators should incorporate reflection questions at different stages (checkpoints) of the exercise to verify that students have grasped the individual KUs correctly. An additional benefit of such an approach is that other educators would be able to quickly understand which KUs are covered by the exercise, thus facilitating the continued development of new exercises and labs that address missing KUs of the existing resources.

We believe that each exercise should include the following sections:

- Learning objectives
- A mapping of learning objectives to NICE KSAs and/or CAE KUs
- Prerequisite knowledge
- Network map (or other necessary diagrams pertaining to the exercise)
- Glossary of major terms
- Complete and clear setup directions (for instructors), if applicable
- Scenario-based guided directions for students
- A comprehensive walk-through directions, such as instructional videos (for instructors)

- Hints and references to helpful resources at the end of every stage of the exercise that students could look into before proceeding (a system of checkpoints rather than step-through instructions)
- Submission guidelines (for students)
- An exercise answer key (for instructors)
- A sample evaluation rubric/methodology (for instructors)
- Knowledge base where students can comment, engage in discussions, and ask questions (e.g., Piazza)
- Optional: for extra complex projects, an FAQ section could be beneficial
- Optional: a final challenge scenario (e.g., a capstone project - for students)

Platform Development Recommendations

Based on the platforms evaluated in this work, we developed usability and feature recommendations pertaining to cybersecurity education platforms. The essential platform usability requirements include having a user-friendly intuitive UI/UX design, informative text about progress, and easy-to-access support pages.

Our recommendations for the platform features are:

- A "getting started" demo exercise for students and instructors.
- Setup guidelines (including required resources) to deploy the platform if self-hosted by instructors.
- A process (with examples) for instructors to publish exercises on the platform.
- An ability to add new students to the exercise at any time (some platforms do not allow adding new students if the exercise has been deployed).
- Students' time tracking spent on working on exercises.
- Students' exercise progress tracking.
- An ability to rate, review, and provide feedback for each exercise on the platform.
- An ability to add co-instructors or TAs to assignments.
- A list of platform limitations (e.g., number of students working at the same time, a list of required computing resources, etc.).
- Support mechanisms for students and instructors.
- FAQ for students and instructors.

Additionally, cybersecurity platform developers should keep in mind that some educators are new to this field (especially at the K-12 level) and need easy access to material and instructions. We also believe that funding agencies (e.g., NSF, NSA) should provide opportunities for projects with promising initiatives (e.g., NICE Challenge, US Cyber Range) to give gifts/funds to schools that want to start using these environments for a predefined period of time (like a year) and require these newcomers to provide effective feedback on how to enhance the project.

7. CONCLUSION AND FUTURE DIRECTION

Cybersecurity resources have come to a state of spaghetti code: unstructured and difficult to maintain. In this work, we developed evaluation criteria for cybersecurity exercises and platforms and used them to evaluate existing cybersecurity education resources, determining the instructional value for each of them. We shared our personal experiences using the platforms and provided recommendations to developers and users. These recommendations are not meant to contain an exhaustive list of best practices but rather be a starting point for cybersecurity educators to use and improve upon. Finally, we listed recommendations for exercises and platform development which can enhance the existing cybersecurity posture in education.

Our next steps in this research include developing a survey to poll the community (NICE, NSA CAEs, SIGCSE, etc.) about the evaluation criteria, their feedback on their experiences, and recommendations. In addition, we plan to ask them what resources they use and what approaches they follow to share their experiments, if any. We will compile the results of our future work in the form of a publicly available white paper to the community. Additionally, we plan to work with the community at large to improve the existing cybersecurity posture in education.

8. REFERENCES

Boesen, S., Weiss, R. S., Sullivan, J. F., Locasto, M. E., Mache, J., & Nilsen, E. (2014, August). EDURange: Meeting the Pedagogical Challenges of Student Participation in Cybertraining Environments. In *CSET*.

CAE Documents Library. (n.d.). Documents library. Retrieved May 8, 2023, from https://public.cyber.mil/ncae-c/documents-library

Cyber.org Range. (n.d.). Home. Retrieved May 8, 2023, from https://cyber.org/range

DETERLab. (n.d.). Home. Retrieved May 8, 2023, from https://www.isi.deterlab.net

DETERLab Ticket System. (n.d.). New ticket (login required). Retrieved May 8, 2023, from https://trac.deterlab.net/newticket

DETER Project. (n.d.). Home. Retrieved May 8, 2023, from https://deter-project.org

Ibrahim, A., & Ford, V. (2021). Observations, Evaluations, and Recommendations for DETERLab from an Educational Perspective. *Journal of Cybersecurity Education, Research and Practice*, *2021*(1).

Mirkovic, J., & Benzel, T. (2012). Teaching cybersecurity with DeterLab. *IEEE Security & Privacy*, *10*(1), 73-76.

Mountrouidou, X. (2019). CyberPaths. *Journal of Computing Sciences in Colleges*, *34*(3), 16-16.

NICE Challenge. (n.d.). Home. Retrieved May 8, 2023, from https://nice-challenge.com

Petersen, R., Santos, D., Smith, M. C., Wetzel, K. A., & Witte, G. (2020). Workforce Framework for Cybersecurity (NICE Framework) NIST Special Publication 800-181 Revision 1. *National Institute of Standards and Technology*.

Priyadarshini, I. (2018). *Features and architecture of the modern cyber range: a qualitative analysis and survey*. The University of Delaware.

SEED Labs. (n.d.). Home. Retrieved May 8, 2023, from https://seedsecuritylabs.org

Siraj, A., Ghafoor, S., Tower, J., & Haynes, A. (2014, June). Empowering faculty to embed security topics into computer science courses. In *Proceedings of the 2014 Conference on Innovation & Technology in computer science education* (pp. 99-104).

Taylor, B., Kaza, S., & Zaleppa, P. A. (2021). CLARK: A Design Science Research Project for Building and Sharing High-Quality Cybersecurity Curricula. *IEEE Security & Privacy*, *19*(5), 72-76.

APPENDIX A

Resource Name	Usability Criteria					
	Type of labs	Customizable Labs	Cloud-based	Lab access	Setup guidelines	Level of support
	C1	C2	C3-A	C3-B	C3-C	C4
DETERLab	Both	Yes	Yes	SSH only	N/A	Institutional
NICE Challenge	Both	No	Yes	Web Interface Only	N/A	Institutional
SEED Labs	Both	Yes	No	N/A	Yes	Individual
EDURange	Both	Yes	Yes	SSH	Yes	Individual

Resource Name	Effectiveness Criteria							
	Instructor's Manual Available	Student Instructions Available	Includes Learning Objectives	Mapping to Frameworks	Limitations	Progress Tracking	Time Tracking	Accessibility Level
	C5	C6	C7	C8	C9	C10	C11	C12
DETERLab	Partial	No	No	No	Resources	No	No	Nationwide
NICE Challenge	No	No	No	NICE TKSAs + CAE KUs	Both	No	Lab	Nationwide
SEED Labs	Yes	Yes	Yes	No	Resources	No	No	Nationwide
EDURange	No	Yes	Yes	No	Resources	Yes	No	Nationwide