2021

DeapSECURE Computational Training for Cybersecurity Students: Improvements, Mid-Stage Evaluation, and Lessons Learned

Wirawan Purwanto
Old Dominion University

Yuming He
Old Dominion University

Jewel Ossom
Old Dominion University

Qiao Zhang
Old Dominion University

Liuwan Zhu
Old Dominion University

See next page for additional authors

Original Publication Citation

Purwanto, W., He, Y., Ossom, J., Zhang, Q., Zhu, L., Arcaute, K., Sosonkina, M., & Wu, H. (2021). DeapSECURE computational training for cybersecurity students: Improvements, mid-stage evaluation, and lessons learned. *Journal of Computational Science Education, 12*(2), 3-10. https://doi.org/10.22369/issn.2153-4136/12/2/1

Authors

Wirawan Purwanto, Yuming He, Jewel Ossom, Qiao Zhang, Liuwan Zhu, Karina Arcaute, Masha Sosonkina, and Hongyi Wu

DeapSECURE Computational Training for Cybersecurity Students: Improvements, Mid-Stage Evaluation, and Lessons Learned

Wirawan Purwanto
Old Dominion University
Norfolk, VA
wpurwant@odu.edu

Yuming He
Old Dominion University
Norfolk, VA
yhe004@odu.edu

Jewel Ossom
Old Dominion University
Norfolk, VA
josso001@odu.edu

Qiao Zhang
Old Dominion University
Norfolk, VA
qzhan002@odu.edu

Liuwan Zhu
Old Dominion University
Norfolk, VA
lzhu001@odu.edu

Karina Arcaute
Old Dominion University
Norfolk, VA
karcaute@odu.edu

Masha Sosonkina
Old Dominion University
Norfolk, VA
msosonki@odu.edu

Hongyi Wu
Old Dominion University
Norfolk, VA
h1wu@odu.edu

ABSTRACT

DeapSECURE is a non-degree computational training program that provides a solid high-performance computing (HPC) and big-data foundation for cybersecurity students. DeapSECURE consists of six modules covering a broad spectrum of topics such as HPC platforms, big-data analytics, machine learning, privacy-preserving methods, and parallel programming. In the second year of this program, to improve the learning experience, we implemented a number of changes, such as grouping modules into two broad categories, "big-data" and "HPC"; creating a single cybersecurity storyline across the modules; and introducing post-workshop (optional) "hackshops". Two major goals of these changes are, firstly, to effectively engage students to maintain high interest and attendance in such a non-degree program, and, secondly, to increase knowledge and skill acquisition. To assess the program, and in particular the changes made in the second year, we evaluated and compared the execution and outcomes of the training in Year 1 and Year 2. The assessment data shows that the implemented changes have partially achieved our goals, while simultaneously providing indications where we can further improve. The development of a fully on-line training mode is planned for the next year, along with a reproducibility pilot study to broaden the subject domain from cybersecurity to other areas, such as computations with sensitive data.

KEYWORDS

Parallel computing, Big data, Machine learning, Cybersecurity, Non-degree training, Hands-on

https://doi.org/10.22369/issn.2153-4136/12/2/1

1 INTRODUCTION

State-of-the-art cybersecurity research is increasingly reliant upon advanced computing, also known as advanced cyberinfrastructure (CI) to strengthen cyber systems against attacks. This includes research areas such as penetration testing, intelligent intrusion detection, run-time malware detection, and secure and privacy-preserving machine learning. DeapSECURE (Data-Enabled Advanced Training Program for Cybersecurity Research and Education) is a non-degree computational training program that provides solid foundations in high-performance computing (HPC) and big data for cybersecurity students. DeapSECURE aims to complement the degree programs in cybersecurity, considering the ever-increasing scale of cybersecurity challenges. The goals, approach, and philosophy of the training program have been elaborated in our previous publication [18].

DeapSECURE consists of six modules covering a broad spectrum of topics such as the HPC platform ("HPC"), big-data analytics (BD), machine learning (ML) including neural networks (NN), privacy-preserving methods (CRYPT), and parallel programming (PAR) [18]. Each module is delivered as a three-hour workshop, combining a presentation on current cybersecurity research topics and basic introduction to the CI methods. DeapSECURE emphasizes hands-on experience in CI tools and frameworks as *applied* to solving cybersecurity research problems. Currently, the six modules consider topics such as: spam/phishing analysis, mobile device security, encryption (privacy protection), and hardware security. We built the detailed content and activities for the modules and delivered them as six workshops during the 2018–2019 academic year and a week-long summer institute in June 2019.

This paper is focused on the changes made in the second academic year (2019–2020) of the DeapSECURE *workshop series* program in order to improve the learning experience. This paper is organized as follows: In Section 2, we recap our experience of the first year of the workshop series (2018–2019) as well as the lessons learned. Then, we detail in Section 3 the improvements to the training program implemented in the second year. Section 4 covers the assessment results and lessons learned from the second year of the training. In Section 5, we briefly cover the pilot online workshop conducted in Summer 2020 as an online virtual event during the time when all the educational activities were held virtually nationwide. We briefly outline our future direction in Section 6, then conclude in Section 7.

2 FIRST-YEAR RECAP

The DeapSECURE's six lesson modules were delivered as a series of six workshops during the 2018–2019 academic year (three workshops per semester). They were all offered again as a summer institute in June 2019. In this paper, we will focus primarily on our workshop series experience. Each workshop began with a 30-minute cybersecurity research presentation by an Old Dominion University (ODU) faculty, followed by an introduction of a CI technique, such as big data or machine learning, featuring rather extensive hands-on activities on ODU's Turing HPC cluster. Because the workshops would run during the school semesters, we decided to limit the length of each workshop to three hours. This time duration would give an opportunity for students to do the exercises during the workshop, while preventing a long-drawn session, which may discourage participation.

Starting already in the first year of the training program, we have been employing pre- and post-workshop surveys, focus groups, as well as our own observation to constantly evaluate and improve our workshops. As initially reported in our previous paper [18], our training received positive response from the students in the first year. The majority of the survey respondents were satisfied or very satisfied with the workshops, and many would recommend the training to others. Students considered the hands-on activities as the most valuable aspects of the workshop. Students were exposed to technologies, methodologies, software tools, and computational resources far beyond their regular coursework.

While our training yielded many positive outcomes, we saw much room for improvement, as evidenced by the challenges we encountered then. The first notable issue was that the attendance of the workshops faltered towards the end of the semester, when the regular coursework put an increasing demand on students' attention and time. For example, in the Fall 2018, the first workshop was attended by more than 30 students, but the last one was attended by 24, a decrease of $\tilde{2}5\%$. In Spring 2019, the workshops were consistently attended by 11–12 students, considerably lower than half of the preceding semester's attendance.

Each workshop considered its own cybersecurity research topic [18], which means that a sizable fraction of the workshop time had to be devoted to introducing a new cybersecurity topic, thereby reducing the amount of time available for the hands-on activities. Indeed, it is a challenge to design a training program, such as DeapSECURE, which aims to provide a broad yet sufficient introduction to advanced computing topics under the tight time constraints of a workshop format. To overcome the limited length of the time available, we developed a written-lesson website for each training module [17]. These websites are available publicly and can be used by learners to further their learning after the workshops.

Another challenge that we have observed is that learners had difficulty in effectively applying high-level concepts taught from either the CI methods or cybersecurity during the hands-on activities. We have determined that this problem stemmed from the mismatch between a rather low-level command-line interface that is used to access supercomputing resources and students' habit of interfacing with computers via graphical interfaces and plug-and-play environments. Hence, the learners fell behind in the hands-on exercises. To solve this problem in the following years, we have decided to resort to more high-level tools, such as Jupyter notebooks, minimize the set of command-line tools used, and select workshop participants that already have some coding skills.

3 SECOND-YEAR IMPROVEMENTS

In the second year of this program, we implemented a number of changes with the goal to improve the learning experience. Among the most significant changes are (a) grouping modules broadly into the "big-data" (data-intensive) and "HPC" (compute-intensive) categories; (b) providing more continuity across several modules by creating a single cybersecurity storyline spanning them; and (c) introducing an optional post-module "hackshop" to enhance the hands-on experience. The changes are expected to facilitate maintaining students' interest and attendance across the entire year of this non-degree program. To take into account semester course load, we shifted the workshop schedule towards the beginning of the semesters by having a workshop approximately every other week with hackshops conducted in-between. As elaborated later in this paper, we also began training teaching assistants to contribute to the development of the lesson materials.

3.1 Revised Workshop Schedule

We reordered the modules taught in the workshops, recognizing that they fall roughly under two categories:

1. **The compute-intensive category** (the HPC, CRYPT, and PAR modules): The key question for this category is how to deal with the computational complexity of cybersecurity problems that take a long time to compute. A common theme in these three modules is the need to split the computational workload across many worker-processes on a modern HPC cluster to greatly reduce the time to solution. Further consideration for high performance will be part of the PAR module.
2. **The data-intensive category** (the BD, ML, and NN modules): The key issue for this category is how to leverage "big data" to detect and defend against cyber threats. Moden computing technologies have generated and made use of enormous amounts of data. From the perspective of cybersecurity, big data can be a two-edged sword. One the one hand, data are assets that are frequently targeted in cyber attacks such as data breaches, denial of service, and botnets. On the other hand, leveraging the state-of-the-art, data-intensive techniques such as machine learning and deep learning has become an indispensable skill for cybersecurity professionals to stay ahead the increasing level of malice and sophistication used to evade detection and defense measures. The three modules in this category aim to introduce these techniques to cybersecurity students.

Table 1: The revised DeapSECURE modules for the 2019–2020 workshop series.

Module	Research Presentation, Presenter, Affiliation	Workshop Hands-on	Hackshop Hands-on	Toolkits
HPC	High Performance Computing and Cybercrime: “An Ounce of Prevention Is Worth a Pound of Cure” (Roderick Graham, Sociology and Criminal Justice)	Determining country of origin of a large collection of spam emails	Making an IP address scanner using UNIX tools	UNIX shell (bash)
CRYPT	Security and Privacy of AI (Cong Wang, Computer Science)	AES and Pailier encryption and decryption	Brute-force AES encryption cracking	AES-Python [22], Python-paillier [5]
PAR	Introduction to Hardware Security and Physical Unclonable Function (PUF) Devices (Yiming Wen, Electrical and Computer Engineering)	Hands-on introduction of MPI for Python	Parallel homomorphic encryption of a bitmap data	mpi4py [6], Python-paillier
BD	QoS Assurance in Cloud Services (Xianrong Zheng, Information Technology & Decision Sciences)	Analytics on a large dataset of smartphone app activity using Pandas	Visualization and exploratory data analysis	Pandas [15], seaborn [7]
ML	Radio Frequency Signal Classification and Detection of Drones Based on Machine Learning (Michael Nilsen, Electrical and Computer Engineering)	Classification of smartphone apps based on system utilization data using classic ML methods	Exploration of various ML models to compare performance	scikit-learn [16]
NN	Virtual MAC Spoofing Detection through Deep Learning (Chunsheng Xin, Electrical and Computer Engineering)	Building neural networks to classify smartphone apps based on system utilization data	Tuning the networks for the best performance (hackshop was cancelled)	TensorFlow and KERAS [3]

We started the 2019–2020 workshop series with three workshops focusing on diverse aspects of parallel computing in the Fall semester, followed by the workshops on data-intensive computing in the Spring. Table 1 shows the updated sequence of CI and cybersecurity topics, as well as the hands-on activities, which we will elaborate in the upcoming section. Each row of the table shows the module name, the cybersecurity research presentation (along with the presenter and affiliated department at ODU), the hands-on activities chosen for the workshop and the hackshop, as well as the toolkits introduced. The overall flow of the training program is shown in Figure 1.

3.2 Rewriting the “Data-Intensive” Modules

In the present era where cyber attacks are proliferating and becoming increasingly sophisticated, the application of big data and machine learning techniques to derive timely, actionable intelligence from streams of data in real-time is rapidly becoming an indispensable need to increase cybersecurity posture [8, 13]. As we realize that the use of data-intensive techniques has gradually become a critical skill for cybersecurity students, researchers, and professionals to possess, we rewrote the three modules (BD, ML, and NN) in order to streamline the learning experience and maximize the learning outcome. Unlike the compute-intensive modules, techniques covered in the three data-intensive modules are closely related to one another and are frequently employed together in real-world applcations. The BD module covers the skill to handle large amounts of data as well as making sense of them using exploratory data analysis and visualization. The ML and NN modules build upon this foundation to introduce predictive techniques at increasing levels of accuracy. For this reason, we select a single cybersecurity use case to motivate the needs of BD, ML, and NN techniques. As the key points of these techniques gradually expand throughout the three modules, learners will see the entire pipeline by which the raw data are transformed into final insights and predictions, leveraging state-of-the-art techniques.

We choose the topic of malware detection in smartphones in our new data-intensive modules. This topic is a very relevant cybersecurity issue, which is also relatively easy to understand for anyone with little to no formal training in cybersecurity, as most students today have smartphones and use them extensively. In the near future, smart device users can expect a significant increase in malware and advancements in malware-related attacks, particularly on the mobile open-source platform as the user base is growing exponentially [4]. We make use of the publicly available sample of the “SherLock” Android smartphone dataset created by Mirsky et al. [14]. The SherLock dataset contains detailed information collected from smartphones used by volunteers over an extended period time. Using this dataset, Wassermann *et al.* [19] explored

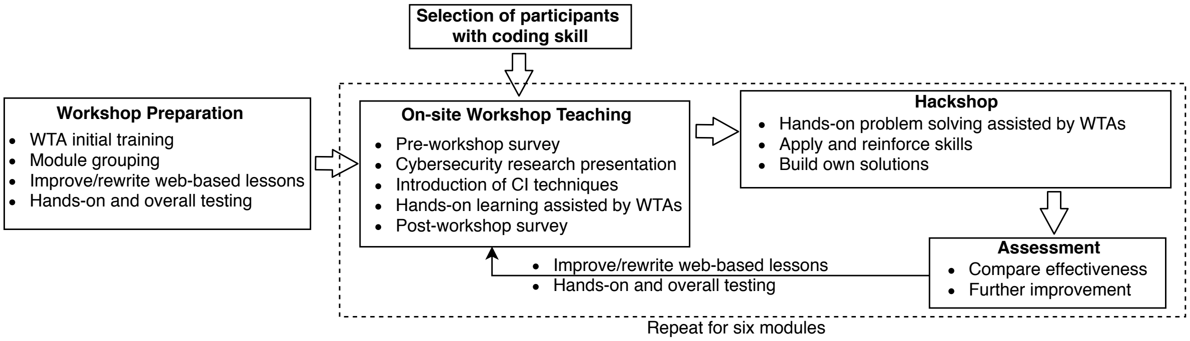

Selection of participants with coding skill
Workshop Preparation
WTA initial training
Module grouping
Improve/rewrite web-based lessons
Hands-on and overall testing
On-site Workshop Teaching
Pre-workshop survey
Cybersecurity research presentation
Introduction of CI techniques
Hands-on learning assisted by WTAs
Post-workshop survey
Hackshop
Hands-on problem solving assisted by WTAs
Apply and reinforce skills
Build own solutions
Assessment
Compare effectiveness
Further improvement
Improve/rewrite web-based lessons
Hands-on and overall testing
Repeat for six modules

Figure 1: Overall process of the workshop series.

an approach to identify running applications and detect malware activity by analyzing this dataset. Based on their idea, we devised a simplified application classification task and split the entire analysis process into three parts in order to fit it into our rewritten lesson modules. Over the course of the workshop series, the approach of using a single cybersecurity topic would help conserve more time to use in teaching and/or hands-on activities.

For the BD module, we switched our choice of toolkit from PySpark [21] to Pandas [15]. Both are widely used tools that have their own use cases. While Spark is a scalable data processing platform capable of handling extremely large amounts of data (on the order of many terabytes and beyond), Pandas has a more gentle learning curve than PySpark for novice learners, and it is also more popular in the data science community. Although Pandas focuses more exclusively on tabular data, and its scalability is limited to a single computer's random access memory, it is nevertheless sufficient for the purposes of our training program. With this switch, Pandas becomes the base toolkit for all the three data-intensive modules.

3.3 "Hackshops"

To enhance the students' learning experience, we added a "hackshop" as a follow-on session to each workshop. A hackshop is a largely unstructured hands-on session, where learners will actively work on a pre-selected problem and come up with a solution in a small group setting, assisted by instructors and/or teaching assistants. The list of the problems we chose for the hackshops are also listed on Table 1. For the hackshops, we gave the learners some basic instructions and guidelines as well as the goal to achieve, then let them try to work it out on their own for the most part. A hackshop provides an additional opportunity for learners to "hack away" and to sharpen the skills they just learned in the workshop. Unlike the workshops, we made this activity optional to all the learners. Hackshop is a feature we experimented in the second year, as we observed in the first year's workshops that learners did not get sufficient time to freely explore the hands-on materials on their own. We set the hackshop to take place on the same three-hour time slot the week following the workshop.

3.4 Participant Recruitment and Selection

We opened a short enrollment window at the beginning of the Fall semester. We advertised the training through the University Announcements channel, as well as through targeted emails to students in cybersecurity, electrical and computer engineering, and modeling and simulation programs. The participants were expected to attend all six workshops (Fall and Spring); we incentivized this by offering a certificate of completion for those participating in at least five workshops. In the enrollment form, we collected their basic demographic information (gender, ethnicity, study area), as well as self-assessment of their computer competencies, such as programming languages (whether they know how to read and write and the level of complexity of the program written). We accepted participants that have basic programming skills (i.e. those who have at least written a short program—fewer than 100 lines in any language). We did so because the computational techniques require some experience of programming to apply them. As a result of this selection, the Fall workshops were attended by significantly fewer participants than our expected number of around 20. We therefore reopened enrollment at the beginning of the Spring semester, where we also promoted the training program to the HPC user community at ODU. This resulted in a large initial spike of attendees in the Spring (around 30), which dropped to 10 in the last workshop.

3.5 Lesson Developers' Training

Once the basic contents of each module were developed after the first year, it became necessary to refine and prepare them for the continuity of development in a plug-and-play fashion. To ensure continuity of the lesson maintenance, development, and improvement, we trained four workshop teaching assistants (WTAs), who are co-authors of this paper, to become content developers. This effort is seeding a community of contributors for this training program, which will be needed when the training project moves toward an open, community-driven development lifecycle in the near future.

Before the Spring workshop series, a PI held weekly meetings for several months to train the WTAs. The training began with an introduction to pedagogy, lesson development methodology, and tools such as Git/Gitlab, Jupyter notebooks [12], and Jekyll [11]. These initial training sessions prepared the WTAs for a smooth collaborative development process with the PIs to update and/or rework the modules.

The three data-intensive modules (BD, ML, NN) were rewritten collaboratively by the WTAs and the PIs immediately following

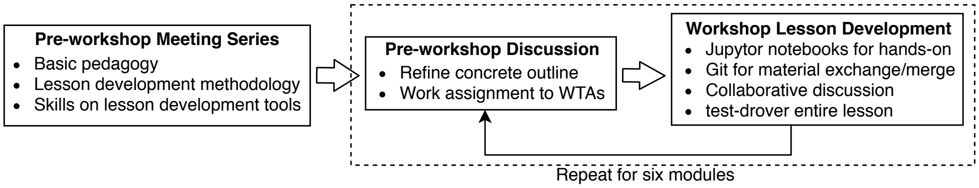

Pre-workshop Meeting Series
• Basic pedagogy
• Lesson development methodology
• Skills on lesson development tools
Pre-workshop Discussion
• Refine concrete outline
• Work assignment to WTAs
Workshop Lesson Development
• Jupytor notebooks for hands-on
• Git for material exchange/merge
• Collaborative discussion
• test-drover entire lesson
Repeat for six modules

Figure 2: Overall process of the lesson developers' training.

the initial training. First, the team applied the reverse instructional design approach [20] to identify the core concepts needed to achieve the objectives of a lesson. These core concepts were weaved into the lesson outline and the hands-on activities. Each WTA was assigned to build specific parts of the written lesson and/or the hands-on activities by utilizing the knowledge learned from the training. Jupyter notebooks were extensively used to draft and refine the hands-on activities, and a private Gitlab repository was used to exchange and merge lesson materials under development. The WTAs also test-drove the entire lessons, ensuring that the involved steps/operations were clearly understood and making the necessary adjustments. These exercises proved especially valuable to prepare the WTAs to lead breakout sessions in the online delivery mode, because each WTA could separately lead the help session that was tuned by them to suit their own teaching style and preferences. The process of WTA training is shown in Figure 2.

4 ASSESSMENTS AND LESSONS LEARNED

Training assessments were conducted both in the first (Y1) and second (Y2) years of the program. They included online demographic data collection, pre- (PRE) and post-workshop (POST) knowledge questions, and post-workshop opinion questions to evaluate the content and format of the workshops. Figure 3 shows the participants' profiles for Y1 and Y2, including the demographics that show the diversity of participants in race and gender, student classification, and major. In both years, we had a similar total number of unique people participating in at least one of our workshops (44 in Y1 and 43 in Y2). In both years, we had a diverse mix of people in terms of their ethnicity, gender, academic classification, and major. In Y2, more cybersecurity students were drawn into this training, which indicates a positive increase in their interest to what we teach in this program.

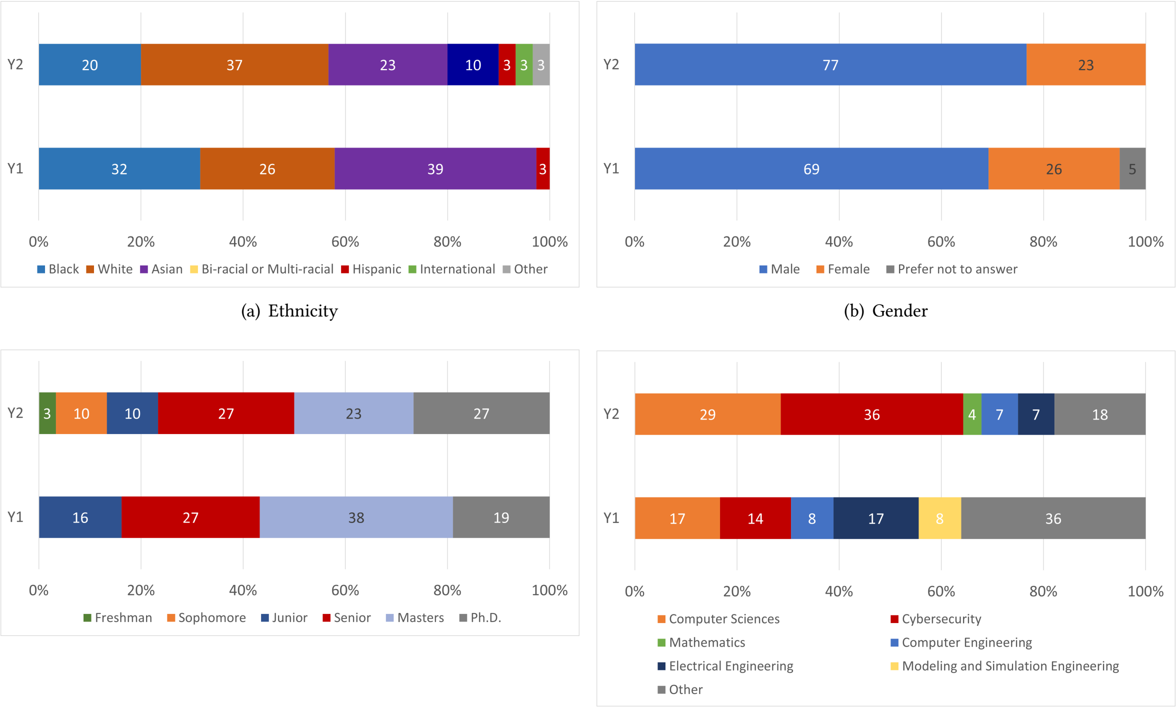
In Y2's enrollment form, we added three new questions to shed light on students' familiarity with UNIX, Python, and C/C++. We asked students to self-evaluate their familiarity with these basic tools: not familiar, novice, intermediate, or expert. In the Fall semester, a large majority indicated that they were not familiar or were novice (about 80%, 90%, and 62% for UNIX, Python, and C/C++). Since UNIX and Python form a critical base for the training, in the first two workshops, we added a brief introduction to these tools. In the Spring semester, we had a better mix of competence, where there were significantly fewer of those who claimed to be unlearned or novice (about 39%, 51%, and 62% for UNIX, Python, and C/C++).

The questionnaires in both years were very similar, which enabled us to compare the effectiveness of our mid-project changes. However, the focus of the evaluation during Y1 was to obtain formative information to improve the workshops as they were being delivered. The post-workshop opinion questionnaire for Y1 was very comprehensive, with 15 questions, including rating of specific components (content, organization, pace, etc.) and open-ended questions to gather qualitative information from participants on what needed improvement and what they found to be most and least valuable from each workshop. For Y2, the rating and opinion questionnaire was shortened to five questions. There were no radical differences in answers to the opinion rating and open-ended questions between the two years. All the workshops in each year were rated as good or extremely good by more than 80% of participants. In Y2, two out of six workshops were rated as "neither good nor bad" by one person; in Y1, the opinions on the very first workshop differed greatly, which we took into account right away and remedied in all the subsequent workshops. (See our description on the necessary adjustments in [18]). Overall, the students received the training program very well in both Y1 and Y2; many of them indicated the hands-on exercises and new knowledge as the most valuable takeaways of the workshops.

Two important metrics that we strive to improve by implementing the changes in the second year are (1) attendance retention and (2) knowledge acquisition. We will consider both quantitative measures (such as number of participants, knowledge assessment results) as well as qualitative and anedoctal feedback to evaluate the impact of our effort in Y2. While the quantitative measures shed light on the areas we need to further improve, we still receive many encouraging feedback from our own observation of, and direct interaction with, the participants.

4.1 Attendance Retention

Figure 4 shows the number of attendees for every workshop we held in Y1 and Y2. Our target is to have 20–25 participants on average per workshop. In Y1, due to the late start of the project in the semester, we held two workshops in the Fall and four workshops in the Spring semester. The first workshop in Fall 2018 was attended by more than 30 students; by the end of Spring 2019, the workshops were consistently attended by 11–13 students, representing 3̃0% of the original number of participants in the first workshop. As we mentioned earlier, in Y2 we started with a lower number of participants, because we required participants to have basic computer programming experience. A second enrollment in the Spring led to another spike in attendance (30), which leveled to 13 at the end. Figure 5 shows a measure of attendees' retention by counting the number of participants who attended any $N = 1, 2, \ldots 6$ number



(a) Ethnicity

(b) Gender

(c) Academic Classification

(d) Major

Figure 3: Demographic distribution of the workshop participants, comparing the first year ("Y1", 2018–2019) and the second year ("Y2", 2019–2020).

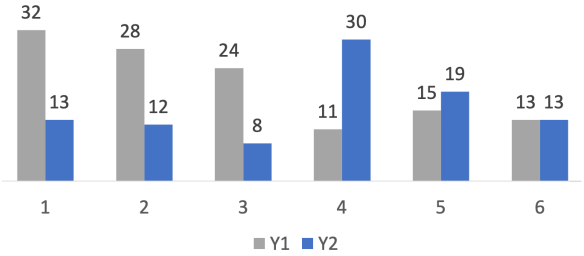


Figure 4: Number of participants attending each workshop.

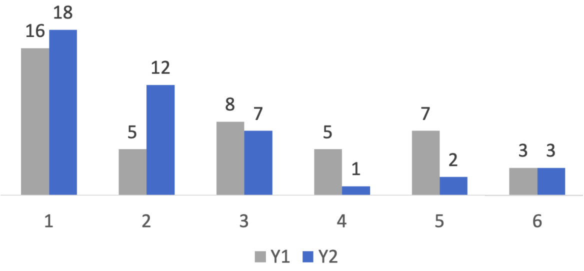


Figure 5: Number of participants that attended any N workshops (N shown on the horizontal axis).

of workshops. The result is also mixed, where Y2 shows better participation for $N = 2$, while Y1 shows better participation in 4 or 5 workshops.

From this we learn that for a non-credit workshop series, attendance tends to spike only on the first workshop; later on participants who remain would do so because they are truly interested in the topic of the workshop. It is worth commenting that while we were not able to achieve our targeted number of 20, from our interactions with the learners, those who remained were very engaged and interested in the materials. The numbers 10–15 may very well be the natural size of the cohort for our local community. Given that our lesson modules are divided into two categories, it seems reasonable that we would open the enrollment twice a year, one for each category, thereby allowing students to pick three modules that better align with their interests.

4.2 Knowledge Assessment

A second metric of interest is whether there is an improvement in the knowledge acquisition as the result of the content changes implemented this year (reorder of the module sequence, rewrite of the data-intensive modules, change in tools). To compare the knowledge acquired by participants, we selected two workshops in each year on the same topics (BD and CRYPT). In each workshop, we asked 5–8 questions on the fundamentals of the CI topics at the beginning of the workshop and at the end to assess the impact of the workshop on the participants' understanding about the topic. These are high-level questions such as, “What is considered the primary goal of looking at big data/large data sets?” and, “Exploratory Data Analytics is ...(mark all that apply)” (for BD module); “What is the homomorphic encryption?” and, “Without the key, you cannot recover the message from a ciphertext. Which statement is true?” (for CRYPT module).

In Y2, the CRYPT module was offered in the Fall, and a short introduction to Python was added due to the fact that the majority of the learners were very new or not familiar with Python. The analysis of the knowledge questions in aggregate for the CRYPT workshops shows that the knowledge acquisition was better in Y1 than Y2. It is likely that participants missed some of the key knowledge due to the inadequate amount of time to cover the less familiar topic of encryption.

In the BD workshop, which was offered in Y2 in the Spring, comparing the PRE- and POST-knowledge responses shows an improvement in four out of five questions, as compared with Y1, where only two out of five questions show an improvement. The BD module was reworked this year, and this improvement may indicate that our improved lesson and delivery resulted in better understanding of the topic.

These numerical results need to be taken with a grain of salt. The sample size, i.e. the number of responses, was very small in these surveys. For the CRYPT workshops, the sample size is 5 and 6, for Y1 and Y2 respectively, whereas for the BD workshops, they are 6 and 11. Hence, an analysis of knowledge acquisition will have to be done at the respondent level to draw deeper conclusions.

In general, the mixed results suggest that we need to adapt further our materials to better fit into the 3-hour workshop duration. For example, in Fall 2019, much time was spent to introduce UNIX shell in the HPC module and basic Python syntax in the CRYPT module. As a result, more pertinent topics (such as job scheduler, parallel processing, and Paillier encryption), were short-handed, and may have lead to weaker results in the POST test after the CRYPT module. In Spring 2020, we adhered better to relative time constraints during workshops, the downside of which was a perception of rushing though the material, as expressed during the focus-group interviews conducted post-workshops. We continue our search to strike a right balance of topic coverage within a workshop. Our current solution is to carefully select topics to cover in depth during a workshop, while leaving the remaining ones for interested learners to pursue on their own using, e.g., our web-based lesson materials and/or Jupyter notebooks.

4.3 Hackshops

In Y2, we provided the new “hackshop” session, which provided a much higher level of interactivity and engagement of the learners with the materials, as well as with TAs and instructors. According to the statistics, over 55% participants came to the hackshop, and we are happy to see five learners from the Y1 workshop series coming back in our new hackshops at least once. They gave us positive feedback on how the workshop synergistically helped them in their coursework. We consider this a promising seed towards building a local community of practice for computational techniques in cybersecurity.

Based on our observation, participants who came to hackshops were able to engage with the hands-on tasks with great interest. In this respect, the hackshops accomplished their purpose. However, the desired goals in these hackshops (e.g., cracking a secret message) were not achieved, partly due to the gap between participants’ programming competence and the required skills to complete these goals. We learned that participants may need more scaffolding, i.e. more guidance and stepping stones, to solve the challenge questions within a three-hour timeframe.

5 PILOT ONLINE WORKSHOP

DeapSECURE workshops were originally designed for in-person workshops, although the sessions were recorded with an intention to build an online version of the training in the future for scalability. The COVID-19 pandemic hit shortly after we finished our last workshop in Spring 2020, which provided us a strong impetus to convert our training to a fully online (remote) format. We decided to try out one pilot online workshop using the BD module in the summer 2020 in lieu of a Summer Institute. This conversion required a thorough redesign of the workshop format to suit the online delivery and learning experience. The planning and redesign process took a substantial amount of time (about three months). A great challenge with the online format was the lack of interpersonal interactions and the inability to directly assist learners on their own computers. Another significant challenge was the limited screen real-estate available for the hands-on format. To help learners overcome these challenges, we developed three Jupyter notebooks which closely mirror the progression of the hands-on activities in the web-based lesson module. The key points as well as incomplete code snippets from the web-based lesson were incorporated concisely in the notebooks, thereby removing the need to open two browser tabs to follow the instructor. Participants accessed the Jupyter environment on ODU’s Wahab HPC cluster via the newly deployed Open OnDemand [9] web-based interface. This proved to alleviate most of the technical difficulties encountered in the past workshop series.

The pilot workshop consisted of three one-hour sessions with 15-minute breaks in between. About ten participants joined the workshop via the Zoom platform. Each session started off with a brief explanation of the basic concepts as well as hands-on demonstration using the Jupyter notebooks, followed by a hands-on exercise held within smaller groups in Zoom breakout rooms led by WTAs. To maintain participants’ level of interest, we conducted a 5-minute interactive yet competitive quiz session using the Kahoot! platform [1] at the end of each session. The results of the quiz provided feedback by measuring the learning success of the session. The Slack [2] platform was used for nonverbal communications (chats) during the workshop, which we leveraged to maintain contact with learners after the workshop. Slack messages are persistent, thus previously answered questions and addressed challenges can be recalled in the future. Overall, based on the informal feedback from participants, the pilot workshop was successful. A detailed assessment of this event is outside the scope of this paper.

6 SUMMARY AND FUTURE DIRECTION

In summary, we performed major improvement of the DeapSECURE lesson modules by grouping them into the "compute-intensive" and "data-intensive" categories, more tightly integrating the modules to streamline the learning experience. The current version of the web-based lesson materials can be accessed from our main website [17]. We added "hackshop" into our training schedule to increase participants' engagement with the hands-on materials. We trained a cohort of workshop teaching assistants to be contributors to further development and refinement of the lesson materials. The assessment results indicate the need for further adjustments to improve learning experience and outcome. The pilot workshop showed great promise to address some challenges we encountered through the second year project. We believe that the improvements we implemented in the second year will put us in a good position to offer the entire portfolio of DeapSECURE modules online and provide learners with the best online learning experience.

The online pilot workshop in Summer 2020 has shown that online training is not only feasible but even more effective in reaching out to trainees who otherwise could not be part of the program. In the next project year, the development of a fully online training format utilizing all the six modules is planned. Efforts will be made to ensure the online training is engaging and effective. The training modules will be streamlined for online delivery. Lectures will be completed in a large group format while labs and games will be completed in small groups facilitated through Zoom breakout room. Effort is underway to ensure that the training materials (lessons and hands-on) can be ported to other institutions and HPC sites. The PIs will also reach cybersecurity as well as CI professional communities throughout the U.S. to promote the adoption of DeapSECURE in other parts of the country. Once the preparation for fully online workshops have been completed, this training can be offered across universities the Commonwealth of Virginia on "ACCORD", a shared cyberinfrastructure currently being built for computation of protected data as well as training and education [10]. The online workshops will be fully assessed along with a reproducibility pilot study to broaden the subject domain from cybersecurity to another area, such as computations with sensitive data.

ACKNOWLEDGMENTS

The development of DeapSECURE training program is supported by NSF CyberTraining grant #1829771. The workshops utilized Turing and Wahab HPC clusters provided by ODU Research Computing Services, part of Information Technology Services. Wahab cluster was acquired in part using NSF Major Research Instrumentation grant #1828593. We thank Issakar Doude, who assisted the lesson development throughout the first year and part of the second year. We also thank the ODU Distance Learning for their support in recording the workshop sessions.

REFERENCES

[1] 2020. Kahoot! Game-based Learning Platform. https://kahoot.com

[2] 2020. Slack. https://slack.com

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[4] Abdullahi Arabo and Bernardi Pranggono. 2013. Mobile malware and smart device security: Trends, challenges and solutions. In *2013 19th international conference on control systems and computer science*. IEEE, 526–531.

[5] CSIRO's Data61. 2013. Python Paillier Library. https://github.com/data61/python-paillier

[6] Lisandro D. Dalcin, Rodrigo R. Paz, Pablo A. Kler, and Alejandro Cosimo. 2011. Parallel distributed computing using Python. *Advances in Water Resources* 34, 9 (2011), 1124 – 1139. https://doi.org/10.1016/j.advwatres.2011.04.013 New Computational Methods and Software Tools.

[7] Michael Waskom et al. 2017. Seaborn: statistical data visualization. https://doi.org/10.5281/zenodo.592845

[8] B Geluvaraj, P.M. Satwik, and T.A. Ashok Kumar. 2019. The Future of Cybersecurity: Major Role of Artificial Intelligence, Machine Learning, and Deep Learning in Cyberspace. In *International Conference on Computer Networks and Communication Technologies (Lecture Notes on Data Engineering and Communications Technology)*, S. Smys, R. Bestak, J.Z. Chen, and I. Kotuliak (Eds.), Vol. 15. Springer, Singapore, 739–747.

[9] David E. Hudak, Douglas Johnson, Jeremy Nicklas, Eric Franz, Brian McMichael, and Basil Gohar. 2016. Open OnDemand: Transforming Computational Science Through Omnidisciplinary Software Cyberinfrastructure. In *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale (XSEDE16)*. ACM, New York, NY, USA, Article 43, 7 pages. https://doi.org/10.1145/2949550.2949644

[10] Ron Hutchins, Scott Midkiff, Masha Sosonkina, Thomas Cheatham, Deborah Crawford, and ACCORD Team. [n. d.]. The Virginia ACCORD Project. https://www.va-accord.org/

[11] The jekyll team. 2018. Jekyll—static site generator. https://jekyllrb.com

[12] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter Development Team. 2016. Jupyter Notebooks—a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, Fernando Loizides and Birgit Schmidt (Eds.). IOS Press, Amsterdam, 87–90. https://doi.org/10.3233/978-1-61499-649-1-87

[13] Tariq Mahmood and Uzma Afzal. 2013. Security Analytics: Big Data Analytics for Cybersecurity: A Review of Trends, Techniques and Tools. In *2013 2nd National Conference on Information Assurance (NCIA)*. IEEE, 129–134. https://doi.org/10.1109/NCIA.2013.6725337

[14] Yisroel Mirsky, Asaf Shabtai, Lior Rokach, Bracha Shapira, and Yuval Elovici. 2016. SherLock vs Moriarty: A Smartphone Dataset for Cybersecurity Research. In *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security (AISec '16)*. ACM, 1–12. https://doi.org/10.1145/2996758.2996764

[15] The pandas development team. 2020. pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.3630805

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (2011), 2825–2830.

[17] Wirawan Purwanto, Issakar Doude, Yuming He, Jewel Ossom, Qiao Zhang, Liwuan Zhu, Masha Sosonkina, and Hongyi Wu. 2020. DeapSECURE Lesson Modules. https://deapsecure.gitlab.io/lessons/

[18] Wirawan Purwanto, Hongyi Wu, Masha Sosonkina, and Karina Arcaute. 2019. DeapSECURE: Empowering Students for Data- and Compute-Intensive Research in Cybersecurity through Training. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning) (PEARC '19)*. ACM, New York, NY, USA, Article 81, 8 pages. https://doi.org/10.1145/3332186.3332247

[19] Sarah Wassermann and Pedro Casas. 2018. BIGMOMAL: Big Data Analytics for Mobile Malware Detection. In *Proceedings of the 2018 Workshop on Traffic Measurements for Cybersecurity (WTMC '18)*. 33–39. https://doi.org/10.1145/3229598.3229600

[20] Grant P. Wiggins and Jay McTighe. 2008. *Understanding by Design* (2nd ed.). Association for Supervision and Curriculum Development, Alexandria, VA.

[21] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica. 2016. Apache Spark: A Unified Engine for Big Data Processing. *Commun. ACM* 59, 11 (Oct. 2016), 56âĂŞ65. https://doi.org/10.1145/2934664

[22] Bo Zhu. 2015. A pure Python implementation of AES. https://github.com/bozhu/AES-Python.git