Open Access License Notice

Teaching Secure Supply Chain Risk: Experiment in an 'Introduction to Cybersecurity' Course

Terry Downing-Harris
Computer and Information Sciences
Towson University
Towson, USA
0009-0006-5097-6097

Siddharth Kaza
Computer and Information Sciences
Towson University
Towson, USA
0000-0002-9561-6128

Blair Taylor
Computer and Information Sciences
Towson University
Towson, USA
0000-0002-4708-5852

Yeong-Tae Song
Computer and Information Sciences
Towson University
Towson, USA
0009-0002-3252-4477

Abstract—The software supply chain and the security of software applications purchased through the Commercial-Off-The-Shelf (COTS) is becoming the focus of government and industry. Higher educational institutions can help by teaching secure supply chain risk management (SCRM), which can help secure COTS software applications. This work presents the results of an experiment that integrated secure SCRM into the software engineering curriculum at Towson University (a diverse, comprehensive institution with a large computer science program). This integration focuses primarily on using the US National Institute of Standards and Technology (NIST) standards to secure COTS software applications effectively. With a focus on undergraduate education, learning modules used in this integration are designed to be injected into almost any course in software engineering curriculum. The overall goal is to provide a model that can be replicated by all universities for integrating secure SCRM into the software engineering curriculum.

Keywords—*supply chain risk management (SCRM), software supply chain, Commercial-Off-The-Shelf (COTS), secure software engineering, software engineering education, cybersecurity*

I. INTRODUCTION

Current software engineering education provides students with important skills that the industry views as being the catalyst to helping organizations achieve their goals [1]. Generally, the curriculum is based on the Software Engineering Body of Knowledge (SWEBOK), which entails teaching students twelve knowledge areas to ensure student's knowledge of software engineering [1], [2]. Most software engineering programs in higher education institutions focus on teaching students' various aspects of the software development lifecycle, including the essential activities of software specification, development, validation, and evolution [3]. The overall goal is to prepare students to create software that is reliable, safe, efficient, maintainable, and meets the user design specifications [3]. With the increase in software security vulnerabilities, traditional network security methods such as firewalls, antivirus, and intrusion protection systems are no longer sufficient to protect against potential cyberattacks [4]. As a result, schools are educating students about secure software engineering, which entails secure software development and design to produce software that continues to function correctly while under malicious attacks [5].

In addition to the current efforts to teach secure software engineering, higher educational institutions need to address the current industry's need to secure the software supply chain by improving the security and purchase of Commercial-Off-The-Shelf (COTS) software applications. Indeed, the biggest cyberattacks in the recent past, including the Solar Winds attack in 2020, are supply chain attacks. Supply chain attacks plague both government and industry, resulting in President Biden's "Executive Order on Improving the Nation's Cybersecurity," issued on May 12, 2021. This order includes key areas needed for security, including that the federal government must take action to improve the security of the software supply chain [6]. Higher educational institutions can help by teaching secure supply chain risk management (SCRM), which can be used to secure COTS software applications purchased by organizations.

Today, many organizations seek software products that are built faster and cheaper than custom-designed software products. As a result, organizations often purchase COTS third-party software products [7], [8], [9]. Most software engineering programs focus on teaching students how to write secure software but generally do not teach students how to use secure SCRM practices [7], [8], [10], [11]. Therefore, this work integrated secure SCRM into a current software engineering curriculum focusing on teaching students how to assess the risk of the software supply chain, evaluate COTS software applications, and the foundations of secure SCRM based on standards from the US National Institute of Standards and Technology (NIST). A controlled experiment was conducted that showed the integration of secure SCRM successfully taught secure SCRM to undergraduate students.

Specifically, the following research questions were explored:

- What secure supply chain risk management topics can be effectively taught in an introduction to cybersecurity class?

- How can we design learning modules to increase learning and comprehension of secure SCRM concepts?
- How can we design and disseminate learning modules that can be easily integrated into the lower-level programming classes?

The long-term goal is to develop a model universities can easily replicate to teach secure SCRM. Section II presents the research background, Section III outlines the study design, Section IV presents the design of the learning module, and Section V discusses the results.

II. RESEARCH BACKGROUND

Traditionally, if an organization needed to acquire trustworthy software products, it did business with companies it trusted [10]. Though this is still true in defense applications, in today's global sourcing environment, even a trusted supplier's software products could be integrated with software subcomponents obtained from an untrustworthy source [10]. The "faster-cheaper-better" mentality of most organizations today has led to acquiring software products rather than developing software internally. Generally, supply chain management involves tangible products that can be monitored and controlled. However, in the case of software, the product to be delivered is invisible, highly complex, and coded rather than manufactured [7], [8], [10], [11].

A. The Software supply chain

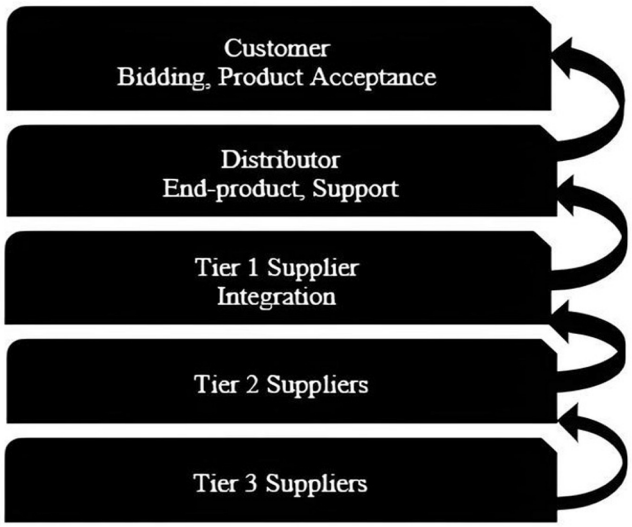

Fig. 1. Software Supply Chain Hierarchy

As the software product moves through the software supply chain, it is difficult to oversee, control and ensure its security. COTS software products are created within a software supply chain that involves an acquisition process and includes three communities of practice: customer, supplier, and integrator. Within the software supply chain, the customer controls the bidding and product acceptance, the supplier is the organization that delivers and ensures the condition of the end product, and the integrator is responsible for aggregating the lower-level components into a single entity which will eventually be passed up the software supply chain [7], [8], [9], [10], [12], [13]. The software supply chain hierarchy operates from the bottom up, as suppliers at the lower-level portion of the hierarchy, such as Figure 1 tier 2 and tier 3, work on software components which are then integrated into a single product and supplied to the customer [7], [10], [14]. See Figure 1.

B. Cybersecurity issues in the supply chain

Vulnerabilities can be introduced within a software supply chain. Another problem is the origin of software used to develop COTS software products is usually unknown to the customer. This anonymity can result in poor quality, namely, unreliable and/or insecure, source code, as well as the potential for malicious objects to be embedded into the code [7], [8], [9], [10], [15], [16]. Purchased licenses and warranties for COTS software products are insufficient for protecting a company's software from these types of supply chain issues; as the number of reported security vulnerabilities in COTS software continues to increase [7], [17]. In addition, supply chains can extend globally, and therefore the security risk of disruption within the supply chain is increased [10]. There have recently been several high-profile supply chain attacks involving corporations such as SolarWinds, Colonial Pipeline and Kaseya, which has further increased the concern over supply chain attacks and third-party risk [18]. Supply chain attacks seek to cause harm to a particular target, such as a company or user, by attacking the less secure components within a supply chain [19]. In July 2021, the Kaseya enterprise Information Technology (IT) firm, announced that the firm had been a victim of a cyberattack against the company's virtual system administrator (VSA) product which is a program that is used by Managed Service Providers (MSPs) to remotely monitor and administer IT services for customers [20]. According to [20], attackers exploited a security flaw in the VSA software and used that security vulnerability to deliver a phony VSA update, which resulted in hackers gaining administrator rights which affected the systems of MSPs and their customers' systems. The Kaseya supply chain attack began with a ransomware attack against one software product which then quickly grew to a ransomware attack across more than one thousand organizations; and hackers demanding $70 billion worth of bitcoin for a universal decryption key to stop the attack. Security Experts say this supply chain attack demonstrates the need for organizations to not only secure their own software, but to also ensure that their supply chain is secure.

There is a need for more work on secure SCRM based on NIST standards being taught in the classroom for an undergraduate software engineering course.

C. Recommendations on teaching secure supply chain risk management

US governmental organizations, as well as organizations involved in developing computer science curricula, are recommending teaching students about supply chain security

and are emphasizing the need to teach secure SCRM [21]. Moreover, government and academic organizations are also stressing the need to teach security in project management.

One such organization is the US National Institute of Standards and Technology (NIST), and the US National Initiative for Cybersecurity Education (NICE), known as the NIST NICE framework. The NIST NICE framework works in partnership with government, academia, and the private sector to provide the basic units needed to describe the tasks, skills, and knowledge needed to perform work in cybersecurity [22]. The NIST NICE Framework consists of key components which are specified as Knowledge, Skills, and Abilities (KSAs) [23]. These KSAs are needed to perform cybersecurity work and tasks for each industry work role. The NIST NICE Framework also provides educators with information for developing curricula, courses, and seminars, along with research that is aligned with the NICE framework's KSAs and tasks [24], [25]. Table I shows knowledge statements from the NIST NICE Framework for the work role of IT Project Management and emphasizes knowledge in secure SCRM.

TABLE I. Knowledge Statements from NICE framework with a focus on SCRM

Knowledge ID	**IT Project Manager (Work Role ID: OV-PMA-002)**
	Knowledge Description
K0169	Knowledge of information technology (IT) supply chain security and supply chain risk management policies, requirements, and procedures.
K0154	Knowledge of supply chain risk management standards, processes, and practices.
K0126	Knowledge of Supply Chain Risk Management Practices (NIST SP 800-161)

The Department of Homeland Security (DHS) and the National Security Agency (NSA) also emphasize the need to teach supply chain security and supply chain management that focuses on managing the risk of external suppliers and vendors. DHS and NSA recognize higher educational institutions that are advancing the study of cybersecurity [26]. To decrease security vulnerabilities through higher education, DHS and NSA jointly sponsor the National Centers of Academic Excellence in Cybersecurity (NCAE-C) program and designate higher educational institutions as a NCAE-C school based on their current degree programs and close alignment to specific cybersecurity-related Knowledge Units (KUs) [26], [27], [28]. NCAE-C is currently recommending the following KUs in both their 2020 and 2019 versions for supply chain security, as shown in Table II.

TABLE II. NSA CAE Knowledge Units focused on Supply Chain Security

NCAE-C Knowledge Units (KUs)		
KUs	***Description***	***Intent of KUs***
2020, 2019	Supply Chain Security (SCS)	The Intent of the Supply Chain Security Knowledge Unit is to provide students with an understanding of the security issues associated with building complex systems out of third party components of unknown (and potentially unknowable) origin.

Other articles [21] reviewing software engineering education; emphasize the need of teaching secure software project management and the assurance of security level of shipped software.

III. DESIGN

This study used a module-based approach to integrate secure SCRM into the computer science curriculum. This was accomplished by incorporating several NIST standards, including NIST.IR.7622, which provides principles that help to mitigate supply chain risk and benefit multiple disciplines, such as software engineers and project managers throughout the software development lifecycle (SDLC). Also, the NIST SP 800-30r1 standard was used to guide on conducting risk assessments, and the NIST SP 800-161r1 standard was used to emphasize how cybersecurity risk can exist throughout a software supply chain.

Each of these standards was incorporated into a course that is part of the software engineering curriculum at Towson University. Learning modules contained NIST standards along with instructional materials to teach students secure SCRM. A control-experiment study was conducted that entailed students taking part in a pretest and post-test with an experiment and control group. Both the pretest and post-test were online but administered in a face-to-face lab setting. The post-test required students to correctly apply secure SCRM practices to help secure COTS software applications. All pretest, post-test, and gain scores were analyzed to determine if the integration of secure SCRM was successful in teaching undergraduate students secure SCRM practices.

IV. SECURE SCRM LEARNING MODULE DESIGN

The use of learning modules can be an effective method for introducing new subject matter in courses. Well-designed modules include a defined structure, teaching materials, and information about the topics taught [29]. The secure SCRM learning modules follow this same general design but are also constructed to fit easily into almost any software engineering course. Three learning modules (i.e., Module#1: Software Supply Chain, Module#2: Challenges in Software Supply Chain, and Module#3: Software Acquisition and Security) were created, each with outcomes defined by Bloom's taxonomy [30]. These learning modules included the following topics:

Managing Risk and Security in Software Supply Chain, Commercial-Off-The-Shelf (COTS) Software Applications, and a brief section on Software Project Management Software Acquisition. Additionally, these learning modules are designed to be taken in sequence (i.e., Module#1, Module#2, Module#3), and the students were required to complete the previous module's lecture, summary, and quiz before the next module became available.

Participants in the experiment group were able to access the learning modules through the Integration of Secure SCRM Experiment website (designed on the canvas learning management system) as often as needed and in a self-paced manner to reinforce secure SCRM practices. These online learning modules were designed for students to complete within 45 minutes to an hour.

A. Module Structure

In general, each module contained real-world examples, learning objectives, and instructional materials along with a list of terms. Quizzes required students to answer multiple-choice and short-answer questions. Students were also able to access published articles pertaining to each module's topics through the Integration of Secure SCRM Experiment website. Learning module#1 and module#2 prepared students for the last module, module #3, and each module entailed the following.

Module #1: Software Supply Chain: Provided foundational information on what a software supply chain is, how a software supply chain can become nested, and the advantages and disadvantages of a software supply chain.

Module #2: Challenges in Software Supply Chain: Emphasized foundational information on how cybersecurity risk can exist throughout a software supply chain, resulting from the potential for harm or compromise from suppliers and their supply chains. In addition, issues caused by globalization of a software supply chain and types of cyberattacks were also covered.

Module #3: Software Acquisition and Security: Covered secure SCRM along with NIST standards. In addition, this module also briefly covered the basics of the request for proposal (RFP) process used in software project management, when acquiring COTS software applications.

Instructions were given on software supply chain risk, and how to use secure SCRM to help to prevent these risks. This was accomplished by teaching the 10 NIST principles, with a focus on principle 1 and principle 2. These NIST standards were broken down into brief descriptions and added into the module to assist undergraduate students' understanding of NIST standards. Principle 1: (Uniquely Identify Supply Chain Elements, Processes, and Actors) and Principle 2: (Limit Access and Exposure within the Supply Chain) were both used in this module to control the communities of practice (i.e., customer, supplier, integrator) within a software supply chain; with a primary focus on suppliers. A sample RFP was given which included several key elements of an RFP (e.g., project description and purpose, project deliverables, acceptance criteria, evaluation criteria).

In preparation for the post-test, this module also included an exercise that presented a potential supply chain risk based on information in a sample RFP, and required students to apply appropriate secure SCRM practices to the RFP that would help prevent supply chain risk from occurring.

V. INTEGRATION OF SECURE SCRM EXPERIMENT

In the spring 2024 semester at Towson University an experiment was conducted in the software engineering curriculum course: CIS377 Introduction to Cybersecurity. Forty-six undergraduate student participants (i.e., twenty-three from experiment group (CIS377.406) and twenty-three from control group (CIS377.101)) completed an online pretest in a face-to-face lab setting. The pretest did not require any prior knowledge or preparation and tested for any previous knowledge or awareness of secure SCRM practices. After completing the pretest, the experiment group accessed the online secure SCRM learning modules for a week and the control group was exposed to the standard lesson plan on discussing software security and other topics. Finally, a total of thirty-eight undergraduate student participants (i.e., eighteen from experiment group (CIS377.406), and twenty from control group (CIS377.101)) followed through with completing the online post-test in a face-to-face lab setting.

A. Results

To determine the experiment's effectiveness, an independent samples t-test was run, and gain scores resulting from the pretest and post-test were analyzed. First, pretest scores were analyzed as an independent samples t-test revealed that the experiment group's pretest mean ($M = 3.70$, $SD = 1.490$) was not significantly different (at the $p<0.10$ level) than the control group's pretest mean ($M = 3.30$, $SD = 1.222$) indicating that the two groups were starting at the same level of knowledge.

On completing the experiment, the experiment group had a higher mean ($M = 9.94$, $SD = 1.474$) in the post-test than the control group's post-test mean ($M = 7.65$, $SD = 2.231$); and the difference was statistically significant at the $p<0.05$ level. See Figure 2.

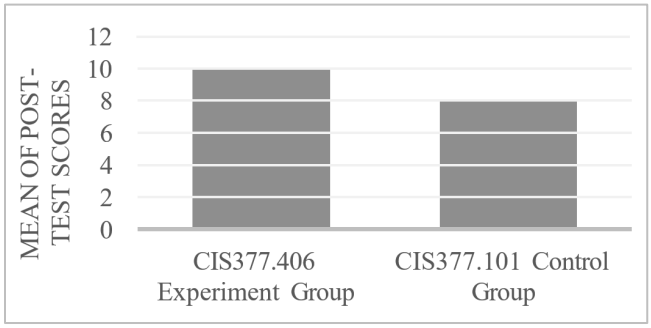

Fig. 2. Introduction to Cybersecurity Courses

The experiment group's post-test mean (M = 9.94, SD = 1.474) was also higher than its pretest mean (M = 3.70, SD = 1.490) at a statistically significant level ($p<0.05$).

B. Discussion

The experiment group was able to successfully assess the risk of a software supply chain and evaluate COTS software applications by applying secure SCRM practices in the post-test. In addition, the experiment group's knowledge and awareness of secure SCRM improved from the pretest to the post-test after students completed the learning modules. This was also indicated in the experiment group's gain scores being higher than the control group's gain scores after both groups took the post-test. See Table III. The experiment showed that this integration of secure SCRM was effective in teaching secure SCRM to undergraduate students.

TABLE III. Experiment Group Higher Gain Score

CIS377 Class Sections	Pretest	Post-test	Gain Score
406 Experiment Group	3.70	9.94	6.24
101 Control Group	3.30	7.65	4.35

VI. CONCLUSION, LIMITATIONS AND FUTURE WORK

Supply chain attacks are on the rise, yet many organizations continue to purchase COTS software applications. Government and academic organizations (e.g., NIST NICE framework, NSA, DHS, ACM, and IEEE) call for academia to teach students about secure SCRM practices. This work integrated secure SCRM practices into the software engineering curriculum at a large, diverse, comprehensive university. Teaching undergraduate students about secure SCRM based on NIST standards helps expose them to the risks and security issues associated with the software supply chain.

Our experiment showed students effectively learned secure SCRM using our modules. However, limitations within the study design involve using just two of the 10 NIST principles standards (i.e., principle 1 and principle 2) to teach students secure SCRM practices. Exploring the use of more of the 10 NIST principles standards would increase security within the supply chain during the development of COTS software applications, and provide additional topics for teaching students about secure SCRM practices.

Therefore, the findings from the successful experiment, in addition to discovering the use of more of the 10 NIST principles standards, will be the catalyst for future work involving further experiments in integrating secure SCRM into the cybersecurity and software engineering curricula. The goal is to disseminate the learning modules created (on the Canvas learning management system) on the CLARK curriculum library (https://www.clark.center/).

This work benefits the software engineering curriculum, by equipping undergraduate students to meet the current industry need for securing the software supply chain, providing a model that all universities can use for integrating secure SCRM into software engineering curricula, and helping to address White House's call to improve the security of the software supply chain.

ACKNOWLEDGEMENT

This project was partially supported by the NSA National Centers of Academic in Excellence in Cybersecurity Program (NCAE-C) under Grant H98230-21-1-0175.

REFERENCES

[1] N. Devadiga, "Software engineering education: Converging with the startup industry," in *Proc. of the 30th IEEE Conference on Software Engineering Education and Training*, 2017. https://doi.org/10.1109/CSEET.2017.38

[2] V.Garousi, G. Giray, and E. Tuzun, "Understanding the knowledge gaps of software engineers: An empirical analysis based on SWEBOK," *ACM Transactions on Computing Education*, vol. 20, no. 1, pp. 1–33, 2019. https://dl.acm.org/doi/abs/10.1145/3360497

[3] I. Sommerville, *Software Engineering Ninth Edition*. 501 Boylston Street, Suite 900, Boston, Massachusetts 02116: Pearson, 2011, ch. 1 and 16.

[4] N. Mohammed, M. Niazi, M. Alshayeb, and S. Mahmood, "Exploring software security approaches in software development lifecycle: A systematic mapping study," *Computer Standards & Interfaces*, vol. 50, no. 1, pp. 107–115, 2017. https://doi.org/10.1016/j.csi.2016.10.001

[5] M. Zarour, M. Alenezi, and K. Alsarayrah, "Software security specifications and design: How software engineers and practitioners are mixing things up," in *Proc. of the 24th International Conference on Evaluation and Assessment in Software Engineering*, 2020. https://doi.org/10.1145/3383219.3383284

[6] J. R. Biden, "Executive order on improving the nation's cybersecurity," *Federal Register The Daily Journal of the United States Government*, vol. 14028, no. 1, pp. 1–47, May 2021.

[7] N. R. Mead, A. Kohnke, and D. Shoemaker, "Secure sourcing of COTS products: A critical missing element in software engineering education," in Proc. of *32nd IEEE Intl. Conference on Software Engineering Education & Training*, May 2020. https://doi.org/10.1109/CSEET49119.2020.9206233

[8] D. Shoemaker, N. R. Mead, and A. Kohnke, "Teaching secure acquisition in higher education," *IEEE Security & Privacy*, vol. 18, no. 4, pp. 60–66, 2020. https://doi.org/10.1109/MSEC.2020.2989644

[9] C.S. Tang and J. Zimmerman, "Information and communication technology for managing supply chain risks," *Communications of the ACM (CACM)*, vol. 56, no. 7, pp. 27–29, Jul. 2013. https://doi.org/10.1145/2483852.2483862

[10] K. Sigler, D. Shoemaker, and A. Kohnke, *Supply Chain Risk Management: Applying Secure Acquisition Principles to Ensure a Trusted Technology Product (internal Audit and IT Series) 1st Edition*. Tylor and Francis Group, CRC Press, 2018.

[11] D. Shoemaker, "Integrating standard SCRM knowledge into conventional academic curricula: an examination of the current state of things.," *Bellwether Publishing*, pp. 1–13, Jan. 2019. https://doi.org/10.1080/07366981.2019.1555931

[12] J. Gido and J. P. Clements, *Successful Project Management Fifth Edition*. 5191 Natorp Boulevard, Mason Ohio, 45040: South-Western Cengage Learning, 2012, ch. 3.

[13] T. G. Hawkins and M. J. Gravier, "Integrating COTS technology in defense systems: A knowledge-based framework for improved performance," *European Journal of Innovation Management*, vol. 22, no. 3, pp. 493–523, 2019. https://doi.org/10.1108/EJIM-08-2018-0177

[14] M. Windelberg, "Objectives for managing cyber supply chain risk," *International Journal of Critical Infrastructure Protection*, vol. 12, pp. 4–11, Mar. 2016. https://doi.org/10.1016/j.ijcip.2015.11.003

[15] D. Blessman, "Protecting Your Software Supply Chain," *Risk Management*, pp. 1–2, Jan. 2019.

[16] F. E. McFadden and R. D. Arnold, "Supply chain risk mitigation for IT electronics," in *Proc. 2010 IEEE International Conference on Technologies for Homeland Security (HST)*, IEEE, Nov. 2010, pp. 49–55. https://doi.org/10.1109/THS.2010.5655094

[17] S. Bulkan, and B. E. Ozkan, "Hidden risk to cyberspace security from obsolete COTS software," in *Proc. of 2019 11th International Conference on Cyber Conflict*, Jun. 2019. https://doi.org/10.23919/CYCON.2019.8756990

[18] S. Weigand. (2021) Supply chain breaches negatively affect 97% of study respondents. [Online]. Available: https://www.scmagazine.com/news/breach/supply-chain-breaches-negatively-affect-97-of-study-respondents

[19] M. Ohm, A. Sykosch, and M. Meier, "Towards detection of software supply chain attacks by forensic artifacts," in *Proc. of the 15th International Conference on Availability, Reliability and Security*, Aug. 2020. https://doi.org/10.1145/3407023.3409183

[20] L. Whitney, "Kaseya supply chain attack impacts more than 1,000 companies," https://www.techrepublic.com/article/kaseya-supply-chain-attack-impacts-more-than-1000-companies/.

[21] L. Yang, B. Jones, H. Yu, and B. Chu, "Secure software engineering education: knowledge area, curriculum and resources," *Journal of Cybersecurity Education, Research and Practice*, vol. 2016, no. 1, pp. 1–25, 2016. https://doi.org/10.62915/2472-2707.1004

[22] Cybersecurity Education And Workforce Development, National Institute of Standards and Technology (NIST), 2021. [Online]. Available: https://www.nist.gov/cybersecurity-awareness-training-education-and-workforce-development

[23] National Initiative for Cybersecurity Careers and Studies (NICCS) (2022) Workforce framework for cybersecurity (NICE Framework) [Online]. Available: https://niccs.cisa.gov/workforce-development/cyber-security-workforce-framework

[24] NICCS. Using the NICE Framework. [Online]. Available: https://niccs.cisa.gov/sites/default/files/documents/pdf/using%20the%20nice%20framework_pdf.pdf?trackDocs=using%20the%20nice%20fram%20ework_pdf.pdf

[25] NICE. (2022) Knowledge IDs: (NICE) cybersecurity framework workforce knowledge. [Online]. Available: https://performatron.risk-redux.io/nice_work_roles/OV-PMA-002

[26] National Centers of Academic Excellence in Cybersecurity (NCAE–C) [Online]. Available: https://niccs.cisa.gov/formal-education/national-centers-academic-excellence-cybersecurity-ncae-c

[27] Department of Defense. (2020) 2020 knowledge units. [Online]. Available: https://dl.dod.cyber.mil/wp-content/uploads/cae/pdf/unclass-cae-cd_ku.pdf

[28] Centers of Academic Excellence in Cyber Defense (CAE-CD). (2019) 2019 knowledge units. [Online]. Available: http://www.itm.iit.edu/faculty/CAE-CD_2019_Knowledge_Units.pdf

[29] S. Raina, "Increasing learning and engagement in cybersecurity through segmented and interactive modules," D.Sc. Dissertation, Department of Computer and Information Sciences, Towson University, Towson Maryland, Maryland, USA, 2016.

[30] L. W. Anderson and D. R. Krathwohl, A taxonomy for learning, teaching, and assessing: *A revision of Bloom's taxonomy of educational objectives*. New York: Allyn & Bacon, 2001.