# Automatic Collection of Scoring Metrics in Competitive Cybersecurity Lab Environments

**Nicholas Hempenius, Te-Shun Chou, and Lee Toderick**
**Department of Technology Systems**
**College of Engineering and Technology**
**East Carolina University**

## Abstract

Cybersecurity game-based learning has shown great promise and is likely to be a powerful means of training security professionals in the future. In most game-based cyber security learning systems there are automated scoring mechanisms in place that score participant performance based on some observable scoring metric. The scoring metric is derived from a predefined learning objective. In the Collegiate Cyber Defense System (CCDC) the services score is calculated by a periodic poling of a service to check that it is still online. Some cybersecurity game-based learning systems use sophisticated agents that collect heuristics on the participants' actions and compare them with the actions of known professionals. However, it is rare that the scores calculated automatically by the games system is the only score that a participant receives, often an observer also submits a score for each participant or group of participants. This paper discussed a system, designed using a five-step process that bridged the gap between a simple service poling score and highly sophisticated heuristics. The five steps were, continuous collection of scoring data, transportation, storage, analysis, and comprehensive display. The scoring system was designed to be used in a Competitive Labs-as-a-Service (CLaaS) learning system. The scoring system provides light-weight, secure, and automated scoring of specific data points as deemed necessary by predefined learning objectives.

## 1. Introduction

In a 2017 Survey done by the Capgemini Digital Transformation Institute, 55% of the 501 employers surveyed said that Cybersecurity is number one in a widening digital talent gap [11]. To compound this talent gap, an annual global survey on the state of Information Technology (IT) by ESG indicated the situation may be getting worse. The survey shows, of the 620 IT and cybersecurity professionals surveyed, 51% believe their organization had a problematic shortage of cybersecurity skill in 2018, almost double that of 2014 (23%) [10]. Addressing the talent gap, and lack of Cyber security professionals, there has been a leap towards research in how to educate the future cybersecurity work force. The National Science Foundation's Secure and Trustworthy Cyberspace (SaTC) initiative received an average of $124 million a year in funding from 2014 to 2016 [13]. In 2017, the National Science Foundation requested $149.75 million in funding towards SaTC, where $70.0 million of that funding would go towards education and human resources. The influx of funding and motivation has sprouted new areas of research into the education of the future cybersecurity workforce [14].

Research into all areas of the design and implementation of cybersecurity education systems is necessary and has been well under way for some time [3]. A hot topic for educating cybersecurity professionals is game-based cybersecurity training, also known as Serious Cybersecurity Games (SCG). SCG could very well be how the next generation of cybersecurity professionals will be trained [12]. A class of SCG is cybersecurity Competative Labs-as--Service (CLaaS). CLasS is an all-in-one package that can be deployed at an extremely low cost as a supplementary cybersecurity education tool. This concept is young in name, but similarly designed platforms are nothing new [12][3].

The CLasS referred to, in this paper, presented a restricted access VM-based environment for students [12]. The students were provided with lab walkthroughs designed specifically for the environment. Each student logged into a VM via web interface, which in turn contained a single attack VM and multiple defend VMs, which provided them with a range, or network, of VMs. The nested VM configuration for each student was identical to exclude minor configurations, such as hostnames and passwords. Each student used the lab walkthroughs to harden their own defender VMs while simultaneously attacking other student's defender VMs. This environment was meant to allow students to act dynamically to the changing environment. Students were "scored" and their scores were displayed on the web-interface, updated in real time. Although there has been much literature created on best practices and designs of cybersecurity wargames, exercises, and game-based educational tools, few present technical specifications of the designs [12]. This paper addressed the design of an Automated Score and Message Board system in technical terms, specific to a new approach using nested VM-based environments, CLaaS.

This paper is structured as follows. Section 2 is a literature review of available works related to the design of automated scoring systems in SCGs. Section 3 is an explanation of the methodology used to design the ASMB used in the Cybersecurity CLaaS platform. Finally, we conclude our work in the last section.

## 2. Related Work

There were three primary goals of the ASMB. First was to gather relevant defense and attack data on targets, second was to translate the data into scores, and third was to update players scores for them to see, all in real time. To discover work related to the topic of automating the scoring process in SCGs, we used the Google Scholar search engine and University library resources. The works discovered fell into three categories, general serious game design guidelines, the design and development of a specific game, and non-game cybersecurity laboratory environments.

A common practice in SCGs is the reward of points to participants when a goal or objective is achieved [18]. In the European Conference on Games Based Learning, Mäses et el, discussed "Obtaining Better Metrics for Complex Serious Games Within Virtualized Simulation Environments" [12]. The article explained methods used in VM-Based SCGs for gathering performance metrics related to measuring player skills. Mäses et el, suggested the following five ways to measure a player's performance metrics; direct through user input, utilizing an automated scoring script, time spent on objectives, logging of a player's actions and comparing

them against a known expert's actions, and by which tools the player used to achieve an objective. Each method may be more useful than another, depending on what skills the SCG or specific objective within the SCG was meant to measure. A skill to be measured should be part of a skill set and can be broken down into specific categories which have at least one point of measurement. Given this model, automated scoring systems can give players specific feedback that can be traced back to a skill set [12]. Patriciu and Furtuna, developed a guide for the design of cybersecurity exercises. The guide suggested seven sequential steps that should be followed to design an effective cybersecurity exercise. The steps were as follows: Objectives, Approach, Topology, Scenario, Rules, Metrics, and Lessons learned. As a high-level guide, Patriciu and Furtuna, explained what each step was meant to achieve; an effective scoring engine is transparent to participants and accurately reflects rules regarding scoring. The scoring metrics are to be directly related to learning objectives [17]. For example, if the learning objective was to secure a service using firewall technology, the corresponding metric should have measured whether that service was secured by the firewall's configuration and no other means. In "Best Practices for Designing and Conducting Cyber-Physical System Wargames", Sullivan et al, described scoring metrics as a game "utility". A game's utility might have been the points it rewarded to a participant. Using points as a utility in a SCG has some benefits; they allow tracking of trends over time and can be displayed to participants to instill motivation [18].

Other works focus on the design of an entire specific SCG. The PicoCTF by Carnegie Mellon University was a competitive cybersecurity game released in 2014 that focuses on the offensive aspects of cyber security. Capture the Flag (CTF) cybersecurity games scoreboards function similarly to the ASMB. However, most CTFs required that a player enter a "flag" that was discovered to earn points. A flag could be a cracked password, or a string found in an encrypted file. In the PicoCTF competition, teams of players completed challenges to earn points and increase their score [4]. Brilingaitė et al designed Cybersecurity exercises on the Cyber Security Coordinated Defense Platform (CSCDP) at Vilnius University in Lithuania. CSCDP was a CTF-based game and was built on the OpenNebula Cloud management solution. Comparable to most cybersecurity competitions [8], CSCDP had a blue team acting as defenders of the network, and a red team acting as the attackers [3]. The environment included 6 VLANs, and each VLAN contained services that the blue team were required to maintain and provide to end users. Although scoring was not directly mentioned, the CSCDP platform utilized the Zabbix monitoring tool to monitor network traffic via proxy. The Zabbix tool is an enterprise class, open-source monitoring tool that can monitor the status of network services, servers and hardware. The information gathered by Zabbix in CSCDP was displayed on a central VIP dashboard server [3].

Closely related to SCGs are the educational cybersecurity laboratory environments. These environments did not generally have game like features such as an ASMB. However, they did contain management systems such as the Report tool in the Smallworld Cloud-based platform [5]. Smallworld was a software defined virtual environment that simulated large distributed systems and could also simulate agents or users, using that system. This environment allowed for extreme flexibility in creating complex virtual networks which can be used to train penetration testers, cybersecurity professionals, students, etc. The extensive logging in the environment also made it valuable for post analysis of user data, business intelligence tools, and malware analysis. The Smallword Report tool collected statistical data within the environment to be queried or

displayed as graphical charts [5]. In 2018 Tunc, Hariri, Montero, Fargo, Satam, presented a proof of concept paper on the design, analysis and evaluation of a "Cybersecurity Lab as a Service", dubbed CLaaS. Their system was similar in many ways to the "Competitive Labs as a Service" platform presented in this paper. The CLaaS by Tunc et al, was accessed through a web portal, and provided a virtual environment where students practiced hands on cybersecurity skills in virtual experiments [19]. Students would select an experiment they wished to do and were presented VM terminals though the web user interface. The environment contained a controller that provided automated management features via a custom algorithm. The algorithm performed tasks such as setting up the experiment VMs, networking, various configurations, and then tore down the experiments. Although this environment contained a management controller, there is no mention of a scoring system. The CLaaS system presented by the proof of concept is directed primarily at providing an experimental environment for students to individually practice cyber-attacks [19]. Although this paper did discuss specific technical aspects of the design, this system was not a SCG and did not contain an automated way of scoring the participants.

There has been extensive literature created around the design and development of SCGs, yet it proved difficult to find written academic work describing the technical components. The Literature we discovered relating to development and design of a SCG, or a class of SCGs, contains concepts or high-level overviews with little to no technical details. Since non-game cybersecurity laboratory environments did not have SCG characteristics like competitiveness and participant scoring, these works provided little to no significant insight into the design or development of an ASMB [5][19]. It should be noted that because there was no standard of nomenclature or taxonomy in literature about the design of SCGs, discovery of material was challenging. For example, some developers referred to scores in SCGs as a performance metric, others referred to scores as the game's utility, and some as Serious Games Analytics (SEGA) [3][17][18]. Works with the most technical details, were patent's, such as the "Scoring Server" under patent number US 9,548,000 B2 [20]. For these reasons and the fact that a wealth of work in this area is done within isolated environments, related literature was hard to come by [12].

## 3. Methodology

As previously stated, the primary goals of the ASMB was to gather relevant defense and attack data, translate that data into scores, and display the data to a scoreboard in real time. The methodology behind achieving these goals was independent of score values and the measurement points of learning objectives. The adjustment of score values and measurement points were considered customizable to accommodate specific lab exercises. As illustrated in Figure 1, the ASMB began by collecting data on nested VMs and ended with the comprehensive display of that data to the player. The subsections below explain the approach and considerations for theoretical technical solutions that address each step.

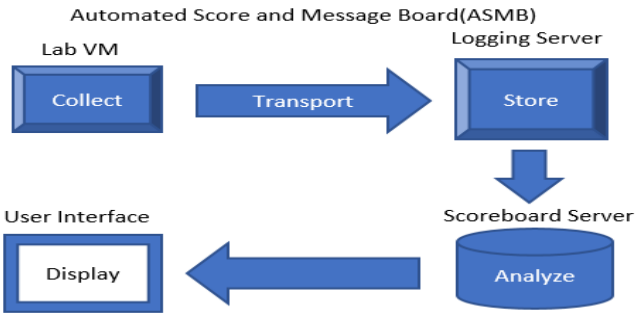


Figure 1. ASMB Model

### 3.1. Continuous Data Collection

It was important to first understand that the data referred to in this section, and sections to follow, are collected from specified data points that were deemed necessary in measuring learning objectives. For example, the data point for learning objective "Securing a login service by disallowing root user login" could have been measured by an actual attempt to login as the root user. The resulting data could have been a zero (0) for a failed root login, or a one (1) for a successful root login.

For the ASMB to have displayed player score data in real time, there must have been a continuous collection of data on a player's VM. However, this posed an obvious issue regarding overhead, on both the player's VMs and the virtual environments network. The collection must have been lightweight and fast on both the VM resources and the networks. It was also important to define "continuous collection". Continuous collection could be a constant stream where data is constantly flowing, or where a variable is under watch for change to trigger collection. It is likely that both options would require considerable resources. Another concern for the collection of data on a student's VM was from where we would collect the data. The very nature of cybersecurity competitions and SCGs promotes pushing the boundaries and bending the rules. The players would also have powerful cybersecurity tools at their disposal. If a player discovered where and how their score data is being collected, the integrity of their score is at risk [12]. This meant a relatively transparent, and difficult to tamper with means of data collection was needed. We considered two ways to collect data in a SCG CLaaS environment. The first was a dedicated VM that collects a player's VM data remotely or second, a collection agent residing on the players VMs. Although a dedicated collection VM may have been a more secure option, it also placed a higher strain on the environment resources and created a single point of failure. A secure lightweight collection agent was likely to be the best option for the CLaaS platform.

### 3.2. Data Transportation

Although the score data was collected within the players VMs, it was not to be stored there. Instead, as described in the section below, the data was to be stored on a VM that acted as the server for receiving data from the agents. This meant the data will also need to be securely transported to the data collection server. However, to ensure that the data flow over the network could match continuous collection the transportation also needed to be fast. A tradeoff between security and convenience was going to be made. We considered two solutions, a security centric solution where data is transported using a secure service, such as FTPS, or data transported using

a specialized high-speed daemon that can be secured, such as Rsyslog with TLS [6]. To prevent congestion and provide added security, a separate network could be easily created that is dedicated to the ASMB system. For example, Figure 2 shows a single student's VM cluster, where each VM has two network connections, one to the lab environment, and the other for the ASMB system.

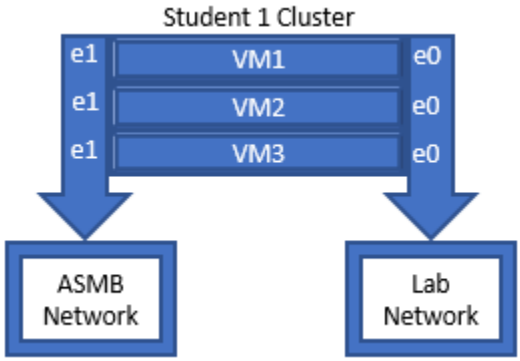


Figure 2. Separate ASMB Network

### 3.3. Data Storage

The data collected will need to be stored using a common format, and in a secure location. A simple solution to secure the data storage is to place a firewall between the VMs that collected the data and the VM that stored the data. A well configured firewall, on a network separate from the lab environment, will provide sufficient security to the storage of the data.

In consideration of creating a common data format, we looked to Rsyslog. Rsyslog allowed for customization of the logging format [1]. The data format needed to meet four criteria. The data needed to be quickly parsed, unique to each player, easily altered, and scalable. A single field delimiter allowed for quick and easy parsing. Using VM hostnames within the log format, and unique identifiers in the log file name, provided a simple means of storing log data unique to each player. Customized logging formats with Rsyslog are easily scaled and changed. Figure 3 depicts an example of a log that meets all criteria. This step, however, is not required. The data could be directly received by the system that is meant to analyze it. By doing this, the data is less likely to be corrupted or tampered with.

Example Scoring Log

```
[root@localhost ScoringLogs]# cat 10.10.101
L1-Defender DATE TIME Field1 Field2 Field3 Field4
L2-Defender DATE TIME Field1 Field2 Field3
L3-Defender DATE TIME Field1 Field3
```

Figure 3. Score Log

### 3.4. Analyze Data

After data has been collected and stored in logs, it needs to be analyzed to create usable output. The goal of the output is to produce a score and message to be directed at the student. To achieve this, the data collected needs to be correlated to a score value along with messages specific to

that score. A simple method would be to use the log format as depicted in Figure 4, and a relational database. Using a relational database, correlating and directing a score and message to a student would be simple, autonomous, and continuous with scripting. Figure 4 is a graphical representation of the concept.

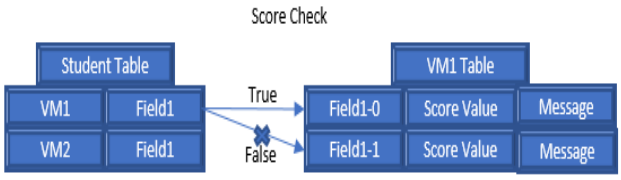


Figure 4. Score Check Relational Database

The data field in the students' VM1 data record contained a value that would then be compared to the VM1 table. The record in the VM1 table that correlated to the value of Field1, contained a score value and a message. Using a database posed potential data integrity and security issues. The database needed to be tamper proof and provide data integrity measures. Initially a decentralized, Byzantine Fault Tolerant (BFT), and Immutable database such as BigchainDB showed promise to address integrity and security issues [2]. However, it became apparent that a distributed blockchain database would not provide a significant increase in security in our environment. The fact that the environment would be self-contained meant we could not utilize key features offered by such a database. Implementing a standard relational, or document-based database such as MariaDB or Mongodb is sufficient for the needs of the ASMB. Leveraging general best practices, the database is sufficiently secure, scalable, and redundant. Simply put, our ASMB design would not utilize the primary features offered by a database such as BigchainDB.

### 3.5. Comprehensive Display of Output

After the data has been collected and analyzed, it will need to be displayed to the student. Depending on what data is collected, and how exactly it is intended to be displayed, this step could be highly customizable. Considering the examples above, a simple score and message board could be displayed on a web interface as a dynamic table. Figure 5 shows a simple example of what the table could look like. This step will be limited to the software used to create the web interface.

Score and Message Board

| Name | Score | Message |
|---|---|---|
| Lisa | +9 | You have successfully cracked the root password of 10.0.2.15. |
| Alexander | +5 | You have successfully configured your DHCP server. |
| David | +1 | You have successfully inject malicious code to home page of 10.0.2.10. |
| Michael | -3 | Your Web server is under attack. |
| Nicholas | -5 | Your FTP server has been compromised. |

Figure 5. Score and Message Board

## 4. Conclusions

This paper focused on a specific and crucial part of a CLaaS SCG, the score and message board system or the ASMB. The purpose of the ASMB was to produce comprehensive output and display it in a user-friendly manner. The output was created by analyzing data retrieve from metric points that align with the SCGs learning objectives. The ASMB operations were broken down into five steps, Collect, Transport, Store, Analyze, and Display. Each steps purpose was discussed and the technical and security aspects of achieving that purpose was addressed.

Step 1 was addressed by developing a heartbeat Terminate and Stay Resident (TSR) script, this made it possible to log relevant data continuously. Using tools, such as WinRAP in Windows, or configuring the "hidepid" option in Linux, we can hide the heartbeat scripts from even privileged users. Step 2 was addressed by utilizing the hidden TSR heartbeat logging script that logged to a rsyslog agent with TLS. For step 3 we created a custom single line rsyslog format that included the host ip address, a single column per objective, and custom values indicating weather objectives were met or not. To analyze the data in step 4 we decided to use a document-based database such as mongodb and parsed data into it using python and bash scripts. Addressing the final step, we created a dynamic table on the CLaaS web GUI using Angularjs, nodejs, and JavaScript. These five steps laid out a suggested theoretical methodology to be used in the development and design of an ASMB. As we develop a SCG CLaaS platform we will continue to refine this methodology. We will first need to place an ASMB design into practice and evaluate it. While evaluating the ASMB we will be able to record lessons learned and use them to bolster the methodology. The produced methodology will provide future developers of SCGs a framework to reference while creating similar scoring mechanisms.

## 5. Acknowledgments

This research is made possible by the National Science Foundation under grant 1723650. The authors are grateful to the support of the Department of Technology Systems in the College of Engineering and Technology at East Carolina University.

## References

[1] Bergfeld, T. 2010. Using a different log Format for all Files. (Feb 23, 2010). Retrieved June 13, 2018 from https://www.rsyslog.com/using-a-different-log-format-for-all-files/

[2] BigchainDB. 2018. BigchainDB Documentation. (2018). Retrieved June 14, 2018 from https://docs.bigchaindb.com/en/latest/

[3] Brilingaitė, A., Bukauskas, Li., Kutka, E. 2017. Development of an Educational Platform for Cyber Defense Training. In *Academic Conferences International Limited*. (Jun 2017), 73-81.

[4] Chapman P, Burket J, Brumley D. PicoCTF: a game-based computer security competition for high school students. Carnegie Mellon University (Aug 2014).

[5] Furfaro, A., Piccolo, A., Parise, A., Argento, L., Saccà, D. 2018. A Cloud-based platform for the emulation of complex cybersecurity scenarios. *Future Generation Computer Systems*, 89, 791-803. DOI: https://doi.org/10.1016/j.future.2018.07.025

[6] Gerhards, R. 2008. Encrypting Syslog Traffic with TLS (SSL). (July, 2008). Retrieved June13, 2018 from https://www.rsyslog.com/doc/v8-stable/tutorials/tls_cert_summary.html

[7] HAL9000. 2017. 8 Tools to Stealth or Hide Running Program from Appearing in Taskbar or Traybar. (2017). Retrieved June 12, 2018 from https://www.raymond.cc/blog/stealth-or-hide-running-program-from-appearing-in-taskbar-or-traybar/view-all/

[8] Heath N, Daniel L, and Erik M. Developing Cyber Competition Infrastructure Using the SCRUM Framework. *8th World Conference on Information Security Education (WISE)*, IFIP Advances in Information and Communication Technology, Auckland, New Zealand. Springer, 20-31. DOI : 10.1007/978-3-642-39377-8_3

[9] Hendrix, M., Al-Sherbaz, A., Victoria, B. 2016. Game based cyber security training: are serious games suitable for cyber security training? In *International Journal of Serious Games* 3, 1 (Jan, 2016), 53-61. DOI: 10.17083/ijsg.v3i1.107

[10] Jon Oltsik. 2018. ESG Research Suggests Cybersecurity Skills Shortage Is Getting Worse. (Jan, 2018). Retrieved June 5, 2018 from http://www.esg-global.com/blog/esg-research-suggests-cybersecurity-skills-shortage-is-getting-worse

[11] Joreome Buvat, Ramya Puttur, Mike Turner, and Marisa Slatter. 2017. Cybersecurity Talent the Big Gap in Cyber Protection Eight Recommendations for How Organizations Can Bridge the Cybersecurity Talent Gab. (July 2017). Retrieved June 7, 2018 from https://www.capgemini.com/wp-content/uploads/2018/02/the-cybersecurity-talent-gap-v8_web.pdf

[12] Masas, S., Hallaq, B., and Maennel, O. 2017. Obtaining Better Metrics for Complex Serious Games Within Virtualised Simulation Environments. In *European Conference on Games Based Learning.* Academic Conferences International Limited, United Kingdom, Reading, 428-434.

[13] National Cyber Foundation. 2016. Secure and Trustworth Cyberspace (SaTC) FY 2016 NSF Budget Request to Congress. Retrieved June 5, 2018 from https://www.nsf.gov/about/budget/fy2016/pdf/41_fy2016.pdf

[14] National Cyber Foundation. 2017. Secure and Trustworth Cyberspace (SaTC) FY 2017 NSF Budget Request to Congress. Retrieved June 11, 2018 from https://www.nsf.gov/about/budget/fy2017/pdf/41_fy2017.pdf

[15] nixCraft. 2014. Linux: Hide Processes From Other Users. (Aug 21, 2014). Retrieved June 12, 2018 from https://www.cyberciti.biz/faq/linux-hide-processes-from-other-users/

[16] Patriciu, V.V., Furtuna, A.C. 2009. Guide for designing cyber security exercises. In Proceedings of the 8th WSEAS International Conference on E-Activities and Information Security and Privacy. (2009), 172–177.

[17] Patriciu, V.V., and Furtuna, A.C. 2009. Guide for designing cyber security exercises. In Proceedings of the 8th WSEAS International Conference on E-Activities and Information Security and Privacy. (2009), 172–177.

[18] Sullivan, D., Colbert, E., Kott, A., and Osterritter, Luke. 2018. Best Practices for Designing and Conducting Cyber-Physical System Wargames. In *International Conference on Cyber Warfare and Security.* Academic Conferences International Limited, United Kingdom, Reading, 651-659,XII,XVI.

[19] Tunc, C., Hariri, S., Montero D. L. P., Fargo, F., Satam, P. 2015. "CLaaS: Cybersecurity Lab as a Service -- Design, Analysis, and Evaluation," 2015 International Conference on Cloud and Autonomic Computing, Boston, MA, 2015, pp. 224-227. DOI: 10.1109/ICCAC.2015.34

[20] William, J., Watters B., Hendricks A., Reider, B., Odom, R., Ledesma, R. 2017. Scoring Server. (Jan. 2017). Patent No. US 9,548,000 B2, Filed March 21st., 2014, Issued Jan. 17th., 2017.

## Biography

NICHOLAS HEMPENIUS is a graduate student in the Network Technology Information Security Concentration Master's degree program at East Carolina University (ECU). He received his bachelor's degree in Information and Computer Technology with a Concentration in Information Security at East Carolina University. He is a member of the ECU Colligate Cyber Defense Competition (CCDC) Team and works as a Graduate Research Assistant.

TE-SHUN CHOU is an Associate Professor in the Department of Technology Systems at East Carolina University. He received his Bachelor's degree in Electronics Engineering at Feng Chia University and both Master's degree and Doctoral degree in Electrical Engineering at Florida International University. He serves as the program coordinator of the Master program in Network Technology for the Department of Technology Systems and the lead faculty of Digital Communication Systems concentration for the Consortium Universities of the Ph.D. in Technology Management. He is also the point of contact of ECU National Centers of Academic Excellence in Cyber Defense Education (CAE-CDE). Dr. Chou teaches IT related courses, which include network security, network intrusion detection and prevention, wireless communications, and network management. His research interests include machine learning, wireless communications, technology education, and information security, especially in the field of intrusion detection and incident response.

LEE TODERICK is a full-time Teaching Instructor within the Dept of Technology Systems, College of Engineering and Technology, ECU, Since 2001. Lee Toderick earned a BS in Computer Science from ECU (1988), and a MS in Computer Information Systems from Boston University (1994). His teaching course load includes Data Storage Management, Cloud Services, Linux Advanced System Administration, and Information Assurance Technologies. He has authored numerous lab experiments in support of his courses, created several secure lab environments, and is copyright author for BroadReach Extended (BRE), a secure, student-centric automated grading system.