Automatic Cyberattack Emulation for Interactive Security Defense Training

Tang Thanh Dat

School of Information Science
Japan Advanced Institute of Science and Technology
September, 2017

Master's Thesis

Automatic Cyberattack Emulation for Interactive Security Defense Training

1510210 Tang Thanh Dat

Supervisor : Associate Professor Razvan Beuran

Main Examiner : Professor Yoichi Shinoda

Examiners : Associate Professor Ken-ichi Chinen

School of Information Science
Japan Advanced Institute of Science and Technology

August, 2017

Abstract

The demand for improving the cybersecurity skills of system engineers is increasing as the number of cyberattacks is raising. There are many training programs and systems for creating training environments. However, in most cases, they only focus on setting up the machines, without a pedagogical point of view.

This thesis has three main contributions. Firstly, it introduces a system to automate learning content preparation and performing cyberattacks through the use of a training database of vulnerabilities. Secondly, it proposes the development of a training user interface inspired by pedagogy theories of distance learning. Last but not least, the implementation is validated in many aspects, from feature coverage to system performance. With the support of the theory, we have demonstrated that the user interface is easy to use and effective. The combination of a tool to setup cyber range from a previous work, an automatic cyberattack mechanism and an interactive web-based training interface creates a complete framework, so trainees can learn cybersecurity by themselves, independently from setting up training sessions manually, which always requires organizers and experts.

In this thesis, first of all, we define requirements for a modern e-learning system for cybersecurity training. We then describe how a defense training system is designed and implemented in this research. The output system is evaluated from two aspects: features and performance. To evaluate the system features, we compare it with other systems and the theory of authentic learning activity. System performance is also validated by user experience and running time - duration needed to create the training environment. The results demonstrate that the new system performs better than the original one in some aspects. Moreover, the pedagogic theories on interaction and authentic activity are proved to help improve the quality of cybersecurity training, so applying educational theories is a promising method to develop technical training programs.

Keywords: Cybersecurity, Cybersecurity training and education, Cyber range, Interaction, Web-based learning.

Declaration: I hereby declare that this whole dissertation is my own work, and that it has not been previously included in any other thesis, dissertation or report.
Student: Tang Thanh Dat

Acknowledgments

I would like to express my sincere appreciation to my supervisor, Associate Professor Razvan Beuran, Japan Advanced Institute of Science and Technology (JAIST), for his support during my master program. Without his comment, encouragement and guidance, even in my worst moments, I cannot finish my master program.

I owe a debt of gratitude to Professor Mikifumi Shikida, Kochi University of Technology for giving me a chance to study in Japan. I will remember his wise words forever.

My sincere thanks are due to Professor Yoichi Shinoda and Associate Professor Ken-ichi Chinen, JAIST. As the committee members, they have given me insightful comments to clarify and improve my research.

There is no word I can use to express my greatly thanks to my parents and my family. Thank you for your love, , for all of the sacrifices that youve made, for raising me up, for teaching me the right things. I would not make it here without you.

I also want to send my thanks to my friends, Bui Ha Duong, Do Khac Phong, Pham Duy Cuong and Nguyen Thi Hao to be with me during my time in JAIST, for sharing with me all joy and sorrow.

Last but not least, many thanks come to Japan, a beautiful country with kindness people.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Our global society is moving into the digital era of Internet of Things (IoT), Big Data, Cloud Computing and Machine Learning. This leads to two consequences: the number of computers increases dramatically, and digital devices integrate deeply into human life. Therefore, it is easy to see in mass media that the number and severity of cyberattacks climb up very quickly. Cyberattacks now do not come from individuals doing it for fun or to get fame. They are usually funded by organizations or states and have serious consequences on financial aspects and system availability. The most recent and famous attack is WannaCry that infected more than 400.000 machines [1] and requires victims to pay in Bitcoins to restore their data. Various other attacks can be named, and they happen every day, aim at everyone, such as the IoT-based Mirai DDoS with 1 Tbps [2] or the Heartbleed vulnerability of OpenSSH - a fundamental software package on many devices.

People soon realized that to prepare for tackling cyberattacks in the future, the most effective way is to train via hands-on activities, where cyberattacks are simulated or emulated in an environment similar to a real-life information system, so that trainees – who are usually cybersecurity engineers or members of Computer Security Incident Response Teams (CSIRT), can see what happens when an incident occurs, then find the best solution to prevent, detect or mitigate it. To do this, cyber ranges – virtual environments for training purpose, are used, to save preparation time and resources.

With the final goal of bringing cybersecurity training to everyone by automating the training content generation and environment setup tasks, the Cyber Range Organization and Design (CROND) NEC-endowed chair at Japan Advanced Institute of Science and Technology (JAIST), is developing CyTrONE (Cybersecurity Training and Operation Network Environment). CyTrONE supports creating a cyber range automatically from a description in text format provided by a training organizer. Utilizing this framework, the cyber range creation task is much simpler and does not require technical works done manually.

Motivated to improve the interaction between trainees and the training environment, in this research, we focus on finding a theory of interaction in distance learning and then trying to apply it to our system. First of all, we define distance learning and the more specific area related to cybersecurity training, such as web-based learning, so the

difficulties are realized. Then some papers about the interaction in web-based learning are surveyed. They show that there are 5 types of interaction in this case. Among them, the most particular issue in web-based learning is *learner - interface* interaction. It is about how easy and effective to use the system, along with how it supports participants during a training session. Besides this, the authenticity is also concerned, since cybersecurity training requires practical situations, in which in-class solutions can be applied to real incidents.

The approach of this research is to contribute to improving and extending the CyTrONE framework. Therefore, by studying CyTrONE, we identified several factors that can be added to help it create a better security training. The framework now is still using traditional training format with questions and answers, where the training content is prepared manually on Learning Management System (LMS) interface. Moreover, the most different aspect between cybersecurity training and other topics, the attacks, require manual activities from an organizer or an expert. Therefore, it still cannot be done fully automatically. Hence, in this research, we propose three new requirements for a modern cybersecurity training, which are:

- (R1) Automation in preparing training session
- (R2) Automation in performing cyber attacks
- (R3) Interaction between the system and trainee

These requirements are implemented as extensions of the original CyTrONE framework. Considering the original training format of CyTrONE, we propose two new training models. The first one is a traditional learning session with a quiz, where both the cyber range environment and training content on LMS are set up automatically, so that an organizer does not need to do any work manually to prepare a training session. The second one is a specific model for cybersecurity: training with vulnerabilities and attacks. In this training, emulated attacks are performed automatically, so that there is no need for IT specialists performing attacks to reproduce incidents. A training database is prepared in advanced, including a Vulnerability Database, an Exploit Database and an Instantiation Database. They have used to setup the cyber range and the attack for training purpose. Learners work with a cyber range trying to patch vulnerabilities. The result is evaluated by the success of the attack. An interactive interface is provided for a better interaction between learners and training environment, such as a clear guidance and a facility to open a terminal to connect directly to the cyber range via the web interface.

This thesis has three main contributions:

- Developed a tool set to automate learning content preparation and performing cyber attacks for studying from a training database of vulnerabilities, so these activities are unmanned.
- Following pedagogy theories, an interactive web-based interface is proposed to make a better communication between learners and training systems.

- Evaluated many aspects of the implementation: feature comparison, validation versus pedagogy theories of authentic learning activity and interaction in distance education, user experience, and the overhead of new system compared with the original one.

The following remainder of this thesis includes 5 chapters. Chapter 2 - Research Background, is an overview of background knowledge related to this research. Pedagogy theories, paper surveys and an introduction to technical standard and software applications are presented in this chapter. From the reviewed knowledge, three requirements for an advanced defense training system are defined in Chapter 3 - Requirement Specifications. Chapter 4 and 5 describe how an Automatic attack controller and an interactive interface is designed and implemented to satisfy the requirements. In Chapter 6, the results of evaluations are illustrated and discussed. There are four evaluations: feature comparison with the original CyTrONE and another cyber range creation tool; running time comparing with the original system; validation with the pedagogy theory; and a survey of user experience. The thesis ends with the conclusion and the plan for future works.

Chapter 2

Research Background

2.1 Pedagogy Background

This section describes some pedagogical theories which are related to learning using new technologies in information science or e-Learning. It comes from the definition of distance learning to using web-based technology in education. After that is the introduction to the classification of interactions in an e-Learning environment.

2.1.1 Distance Learning

Distance learning is not a product of high technologies in the digital era. It appeared a long time before the first digital computer. Distance learning, likes its semantic, is the ability to learn/study from a far distance. That means teachers and learners do not meet each other face-to-face. In this learning environment, learning materials are distributed in forms of books or hard-copy prints. As computers became involved in the delivery of instructional materials, not only print but also electronic media is used, such as electronic files, web-based contents, etc. [4]. The antonyms of *Distance learning* is *on-site learning* or *on-campus study*.

In this definition, there are two terminologies which are usually confused. They are distance learning and distance education. Distance learning is referenced more as the ability to absorb knowledge, whereas distance education is an activity to implement that ability [3].

From a theoretical point of view, Holmberg [7] define distance education as: "Distance education is a concept that covers the non-contiguous learning-teaching activities in the cognitive and/or psychomotor and affective domains of an individual learner and a supporting organization. It can be carried out anywhere and at any time, which makes it attractive to adults with professional and social commitments."

Nowadays, with the development of modern technology and the popularity of digital devices in human life, in most cases, distance learning is the involvement of e-Learning.

2.1.2 e-Learning, Online Learning, and Web-based Learning

e-Learning is an educational activity with the involvement of digital devices and technologies. e-Learning can be distance learning or on-site learning. Some examples of e-Learning are:

- Study over video records of lectures.
- Simulation of physic, chemistry or geography phenomenons by computer application (Adobe Flash, etc.)
- Study over educational computer games.
- Study with mobile applications on smartphones.
- e-Learning websites (Coursera [15],Khan Academy [16], etc.)
- Code challenges (codewars [17])

It is quite true to say that **online learning** is the intersection of *distance learning* and *e-Learning*. Many researchers identify online learning as a more recent version of distance learning which improves access to educational opportunities for learners [3]. In this learning model, a new factor joins learning activities - the Internet. Taking advantage of Internet, learners can study from a very long distance, with no limitation in time and place. Moreover, it simplifies or does not require some preliminary activities such as contacting the educational institution, submitting an application or preparing learning materials.

The last terminology is **Web-based learning**. It is a part of online learning, where a specific technology is used - the Web or WWW (World Wide Web). This is the trend of e-Learning recently. It makes use of advantages of web technology such as popular, easy to use, no need to install, interactive and customizable interface. A web-based application is the combination of a front-end interface (written in HTML, CSS, and Javascript) and a database management system (DBMS), for instance, MySQL. Some back-end programs may work behind to manage and control the operation of the website. Representing for this model is e-Learning websites as shown above. They provide a set of learning materials and tools, including lecture notes, lecture slides, lecture videos, even forums for discussing between learners and exams for assessing study result.

Another implementation of Web-based learning is Learning Management Systems or LMSs. They can be commercial products (e.g. WebCT, Blackboard), or free open source products (e.g. Moodle, Claroline) [5]. LMSs provide frameworks for learning activities to lecturers and education and training institutions. Based on these frameworks, teachers can add contents (e.g. Create courses, Create exams, Add videos, Add homework, Distribute study materials). In this research, Moodle - an LMS is used as an implementation tool.

Figure 2.1 describes the relation between learning environments. Based on the physical distance between instructors and learners, there are *on-site learning* and *distance learning*. The new learning environment - *e-Learning*, can be on-site or distance learning. If an

environment is *online learning*, it must be both *e-Learning* and *distance learning*. A special case of *online learning* is *Web-based learning*.

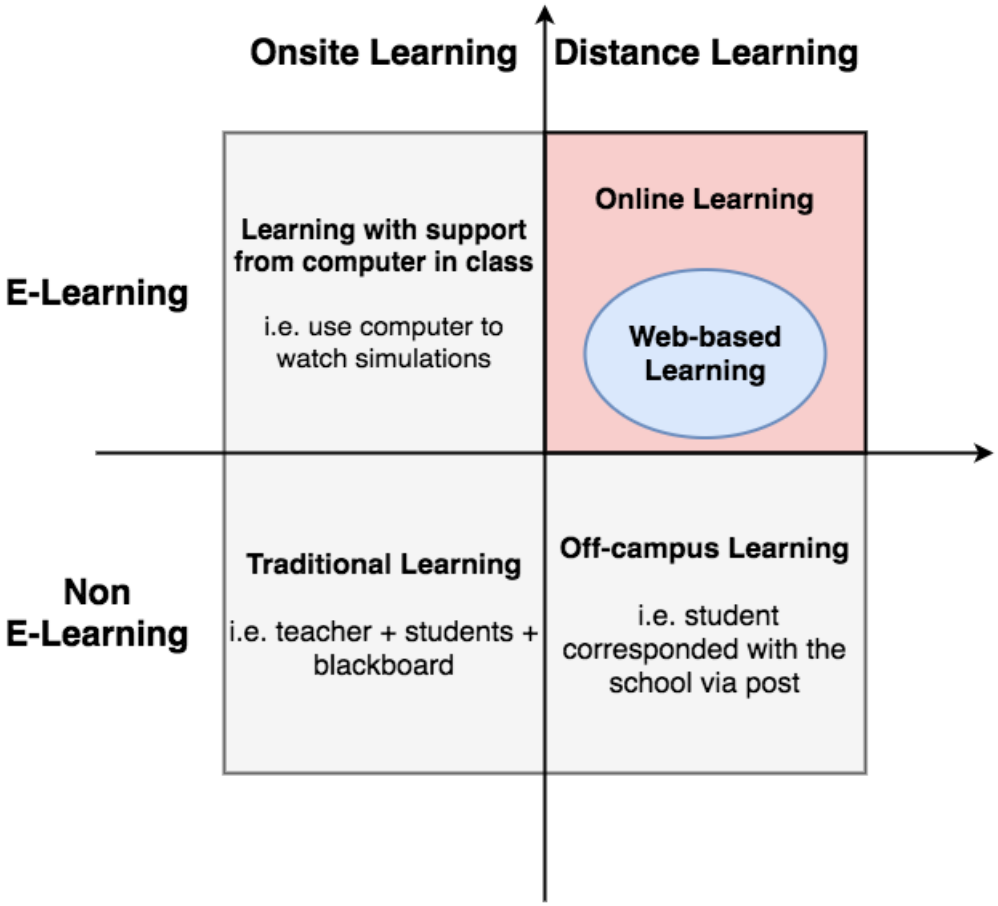


Figure 2.1: Learning environments: Based on technology, there are E-Learning and Non-E-Learning. Based on regard to distance, there are Onsite and Distance learning. Online learning is the intersection of Distance learning and E-Learning while Web-based Learning is one of its special cases

2.1.3 Interaction in Web-based Learning

Distance learning or distance education has a quite long history. Therefore, many theoretical researches have been done in this field. In order to understand this field scientifically, 4 concepts are made: transactional distance, interaction, learner control, and social presence [8]. This research focuses on the interaction concept, so this section only explains about it.

The research of M.G.Moore is the most popular and cited on the topic of interaction in distance learning. Based upon a communication-based framework, defining the sender and receiver [6], M.G.Moore [9] defines three types of interaction, which are learner-

content, learner-instructor and learner-learner interaction. After that, emphasizing the effect of technology on the interaction, Hillman, Willis, and Gunawardena [10] added the fourth type, called learner-interface interaction. They pointed out that "Although each of these three types of interaction addresses the use of technologies as bridges for interaction, they fail to take into account the interaction that occurs when a learner must use these intervening technologies to communicate with the content, negotiate meaning, and validate knowledge with the instructor and other learners." In 2001, Sutton [11] proposed the fifth types of interaction, vicarious interaction. However, there are still discussions about this fifth type. In summary, there are 5 types of interaction in distance learning, with the meanings:

- **Learner-content:** This is the most fundamental interaction in an educational activity. Without it, there cannot be education [9], since this is the process of interacting between learner and learning materials to transfer knowledge. The oldest form of content is didactic text in books and documents. This type of content has increased dramatically between the 18th and the 20th century with the development of printing and paper making technology. From the late 20th century until now, the types of content has expanded to be radio and television programs, video records, e-Learning and website content. The most modern medium to transfer educational content is interactive graphical interfaces of digital software. With the evolution of the content, the interaction between learner and content also changes. It is no longer a one-way communication, from content to learner. The learner-content interaction now can be bilateral, or even multilateral communication.

- **Learner-instructor:** Along with the learner-content interaction, this is also a general one. This type of interaction is usually thought about when mentioning traditional education. In distance learning, it has different characteristics. The main reason for these differences is the physical distance between learner-instructor. It limits the ability to communicate between two sides. Therefore, to maintain the interaction between learner-instructor, some requirements are needed. First of all, instructors must maintain the student's interest in what to be taught [9]. Secondly, instructors need to check the learning progress along with the quality of the output of the course to students. However, in distance learning, due to its characteristics, learners usually rarely give any feedback. Therefore it is hard for instructors to make any further interaction. Besides of it, differing from traditional education, in distance learning, the communication between instructors and learners is one-on-one. It requires much more time and effort to manage and support all student. With the support of modern technology, learner-instructor interaction becomes more effective. It is easier to discuss over a telephone, instance messaging applications, class forum or video-telephony.

- **Learner-learner:** The third type of interaction is made by students is a class or group. Historically, this interaction is proofed to have positive influences on the result of educational activities. Same as learner-instructor activities, distance learning makes it more challenging, but it even worse that without any effort, no

interaction is made between learner-learner, since the learners usually do not know each other or meet each other in the learning process. Even in the modern learning technique of online learning, learners do not realize the existence of others and learning programs do not require the communication between students. However, in recent educational programs, learner-learner interaction drew attention, and the developers/designers try to improve it over some tools, thanks to the development of high technologies such as class forums or even virtual reality (VR) technology.

- **Learner-interface:** authors of this interaction define it shortly: "Learner-interface interaction is a process of manipulating tools to accomplish a task." Moreover, they also suggest that a successful learner-interface interaction must bring to learners not only the guidance to work with the interface but also why they need to do it [10]. All data, or in the case of educational activities, all knowledge is transmitted through an interface. In traditional education, these interfaces are papers with didactic texts and images, blackboards, voice between teachers and students or even body language. In this case, the quality of interface is paper quality or communication skill of people, but it is hard to improve the quality or the improvement is hard to be realized.

  In case of distance learning, the quality of interfaces affects the overall quality much more significantly. They are the quality of audio and video records, Internet speed, graphical interface of an e-Learning application and features which the application support. For traditional education, an instruction to use learning interfaces is rarely used, whereas in distance learning, learners usually require technical support or some kinds of how-to-use document.

  Therefore, in distance learning, especially online-learning and web-based learning, designing an interface which satisfies the interaction between learners and training environment, plus being easy to use and supporting additional features is an important requirement. This is also a goal of this research.

- **Vicarious interaction:** This interaction can be called passive observation or interaction. It appears when a student actively observes and processes direct interactions among other leaners and the instructor [11]. In this interaction, there are four different type of interactors emerged [12]:

  - Direct interactors - who directly interact with other in an activity.
  - Vicarious interactors - who passively observe and process information transferred between direct interactors.
  - Actors - who provided unilateral input regardless of the reactions or comments of others.
  - Non-actors - who did not participate in the communication process.

  An example of this interaction is when a student answers a question from a teacher, and from that, another student understands the content of the lecture or even joins the communication to clarify or add more information into the interaction.

From the definition and analyzing each type of interaction, maintaining interaction in web-based learning environments is much more challenging than in on-site learning, where everyone see others face-to-face. Therefore, when designing and developing a web-based learning program, it is important to concern about how learners interact with the system and trying to improve it. In [6], the authors summarized from various theories in computer interaction, message design and motivation model, to suggest some characteristics for screen design for improving interaction quality:

- Key information (e.g., questions, notice) in prominent locations, for instance, the middle of the screen
- Use highlighting
- Provide orientation clues
- Use universal and familiar icons

2.1.4 Authentic Activity in Web-based Course

In education, there is one thing which is often discussed, namely authenticity of studying activities. Usually, in-school problem solving is formalized and standardized, having clear and enough information, having only one solving method and one answer and sometimes nonsense in practical. Authentic activities require situations in class are same as real-life environments. They should have a meaningful context, related components, being ill defined, and require both investigation and problem solving [13]. An example of comparing traditional learning activity with authentic activity is shown in Table 2.1.

Reeves et al [14] identified ten main characteristics of authentic activities:

1. Have real-world relevance.
2. Be ill-defined, requiring students to define the tasks and sub-tasks needed to complete the activity.
3. Comprise complex tasks to be investigated by students over a sustained period of time.
4. Provide the opportunity for students to examine the task from different perspectives, using a variety of resources.
5. Provide the opportunity to collaborate.
6. Provide the opportunity to reflect.
7. Can be integrated and applied across different subject areas and lead beyond domain-specific outcomes.
8. Be seamlessly integrated with assessment.

9. Create polished products valuable in their own right rather than as preparation for something else.

10. Allow competing solutions and diversity of outcomes.

Following this theory, later in this research, the quality of the implemented system is evaluated apart by matching the theory with features of the system.

In-school learning activity	Authentic activity
John bought 3 apples, each weighs 0.2 kg. After that, he buys 5 oranges, each weighs 0.15 kg. What is the total weight of fruits.	The hand baggage weight restrictions for this flight is 7 kg. John wants to buy oranges and apples and bring them onboard. How should he buy when an apple weighs around 0.2 kg and an orange is 0.15 kg.

Table 2.1: Example of in-school versus authentic activity

2.2 Automation in Cybersecurity Training

Deploying and maintaining information systems is a tedious, time-consuming and error-prone process. Therefore, automation is a work which researchers and engineers always want to develop and improve. In industry, for automating deployment and management of information system, operating system configuration management tools are used, such as Ansible, Chef, Puppet, SaltStack, Vagrant, etc. [18]. However, in the specific case of cybersecurity training, normal tools cannot be used, because 1) They do not support security features, since they are not created for that purpose; and 2) Many researchers choose to simulate cybersecurity incidents, so ordinary tools are useless.

One of the main goals of this research is to automate attack activities in a cybersecurity training. Therefore, surveying for related works focuses on attemps to create cyberattacks for training purpose.

Since the requirement to have a cyber range for testing, training or competing has been raised for a long time, there are many researches and products related to it. Many of them try to simulate or emulate attacks. They can come from military, government, academic research or commercial products. However, due to the fact that they can become attack tools for malicious people, most technical details are omitted from publications [23].

From surveying researches and projects which also reproduce cyberattacks, there are various ones found. From the military area, SIMTEX and SAST can be named. However, SAST only uses a machine with installed tools as an attacker for whole system [19], while SIMTEX only specifies for the computer network of US Air Force [23]. More importantly, as mentioned above, they are military projects, so there is no way to clearly understand what they can do or contribute to improving them. Other researches from Michael E. Kuhl et al [21] and Michael Liljenstam et al [22] only simulate attacks, which means

they are not real attacks, but only demonstrations of systems under attack situations. They are more suitable for understanding basic concepts rather than hands-on activities. Ariel et al [20] use agents on each machine to simulate the effect of each exploit, such as crashing a machine, running a program or seizing root permission. However, it is still a "fake" attack, as it only emulates consequences of cybersecurity incidents, therefore these malicious activities are untraceable, undetectable and unpreventable.

2.2.1 CyTrONE

Realizing the drawback of other systems and frameworks, CROND (Cyber Range Organization and Design) research group at JAIST has developed CyTrONE (Cybersecurity Training and Operation Network Environment) - a security training framework [24]. Figure 2.2 shows the architecture of CyTrONE, which starts with supporting a user interface (UI) for training organizer to manage the training program. A Training Description Generation framework with the support of a Training Database will analyze the requirement of the user to prepare to build up a training environment set. This set includes two main components: an interface for trainees and an environment for hands-on activities.

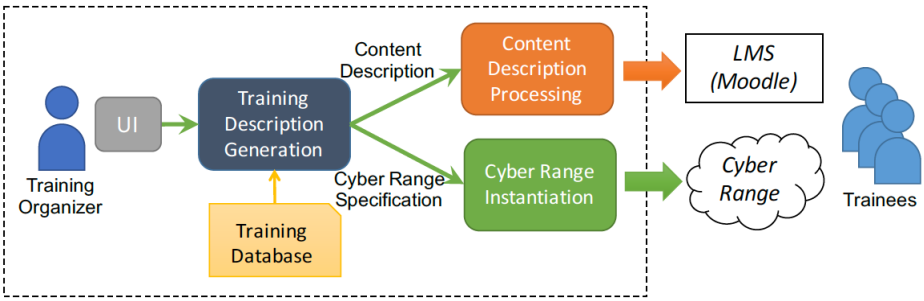


Figure 2.2: System architecture of the cybersecurity training framework CyTRONE [24]

In a hands-on activity, the experimental environment is the most important component. In cybersecurity training, the best training environment is a lab with a "real" information system, so the trainee is able to perform actions same as in practical situations. CyRIS [25] (Cyber Range Instantiation System) is proposed to prepare a virtualized environment specified for training. It receives requirements from the organizer as an input and create a cyber range, aims to serve training sessions with multiple users, thus it is expandable. An overview of CyRIS is presented in section 2.2.2.

To guarantee the quality of the interface, a learning management system (LMS) is used, since it contains a set of functions (e.g. deliver material to the students, administrate tests and other assignments, track student progress, and manage record-keeping [27]) which satisfies learning process in a pedagogic point of view. In this research, an LMS called Moodle [26] is used. Detailed information about it will be described in secion 2.3.

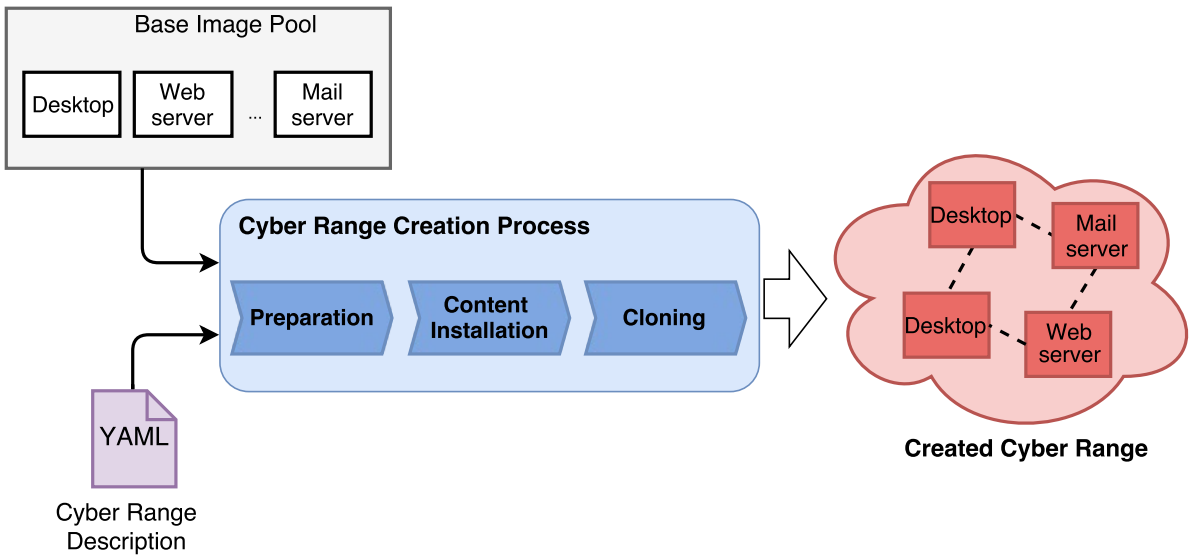


Figure 2.3: CyRIS working flow [25]

2.2.2 CyRIS

CyRIS uses Kernel-based Virtual Machine (KVM) technology for virtualization. This technology is used widely both in research and enterprise environment, since it has many benefits like being built into Linux kernel, open source, having large community, supporting backup and restore, etc. Figure 2.3 shows the workflow of CyRIS. In this system, developers or organizers prepare in advanced a set of base images for different machine purposes. After that, a YAML file called Cyber Range Description is written by the user or produced automatically by CyTrONE framework. This file contains detailed information about requirements for training system, which CyRIS uses in cyber range creation process, which contains 3 steps: preparation (e.g. distribute base image, set up physical machines), content installation and cloning. The output of the system is a cyber range, which contains multi-instance, each instance is used by a trainee, thus the system can serve many users.

2.2.3 STIX

Structured Threat Information Expression (STIX) is a language and serialization format used to exchange cyber threat intelligence (CTI) [28]. Its goal is to simplify and automate the process of sharing incident information for collaborating incident responding, cybersecurity analyzing and cybersecurity situational awareness. This project belongs to OASIS Cyber Threat Intelligence (CTI) TC with the participation of many big corporations and organizations: MITRE, Cisco Systems, AlienVault, Dell, Intel, National Institute of Standards and Technology (NIST), US Department of Defense (DoD), Fujitsu Limited, Fujitsu Limited, NEC Corporation, etc.

STIX uses XML (in the pre-release version STIX 2.0, it changes to use JSON) to de-

scribe events and detailed information of a cybersecurity incident. This formalized and standardized information helps the receiver understand the incident easily and systematically. Since the shared information now has a frame, it is possible to reproduce an incident with its environment (e.g. involved machines, network system, setup of machines) from s STIX package. Therefore, STIX opens a chance to not only cooperation incident response but also recreate cybersecurity incidents for training purpose.

2.3 Learning Management System: Moodle and SCORM

Moodle is one of the three most popular learning management system worldwide [27]. It supports a wide range of functions, such as modern and easy to use interface, personalized dashboard, collaborative tools and activities, file manager, track progress, etc. [26]. The benefit of using Moodle in research is that it is an open-source software program. Anyone can modify it to make their own version of Moodle to meet requirements, edit website interface, or add plugins. Moodle's community has developed many plugins shared for everyone.

Moodle support many methods for users to create learning content, from adding text, videos, images to creating a quiz. Among them, Moodle supports a content package standard, called SCORM - Sharable Content Object Reference Model. It is a promising format since it is widely used and can be used across platforms, thus it saves time for content integration. SCORM is a set of technical standards to guide software developers to create training contents which are able to integrate with learning management systems. Figure 2.4 illustrates the conceptual model of a SCORM package working with an LMS over an API.

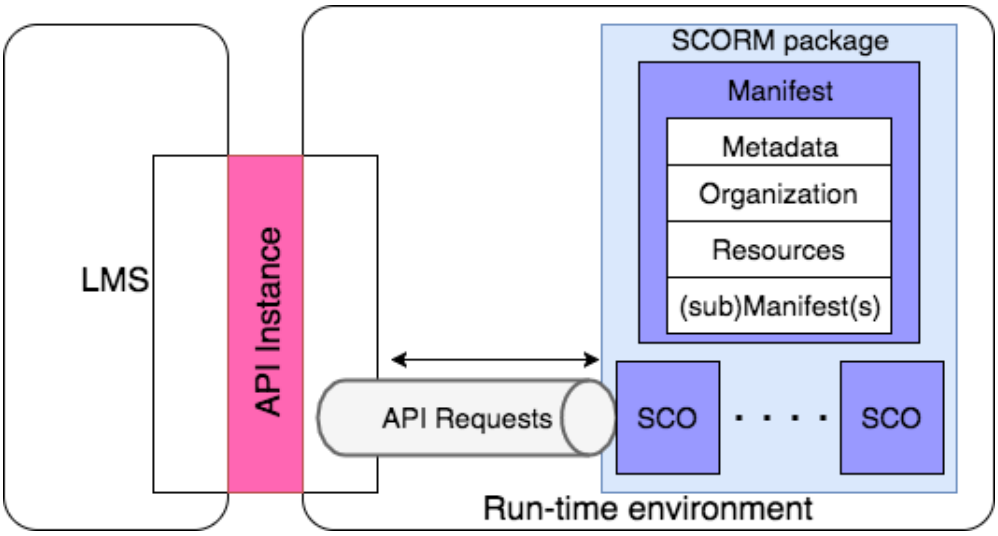


Figure 2.4: Integration between an LMS and a SCORM package

Chapter 3

Requirement Specifications

3.1 Original CyTrONE

A modern cybersecurity training preparation system needs to improve the following points:

- Reduce preparation time
- Provide an unmanned automatic system
- Interact with learners to provide better teaching and training quality

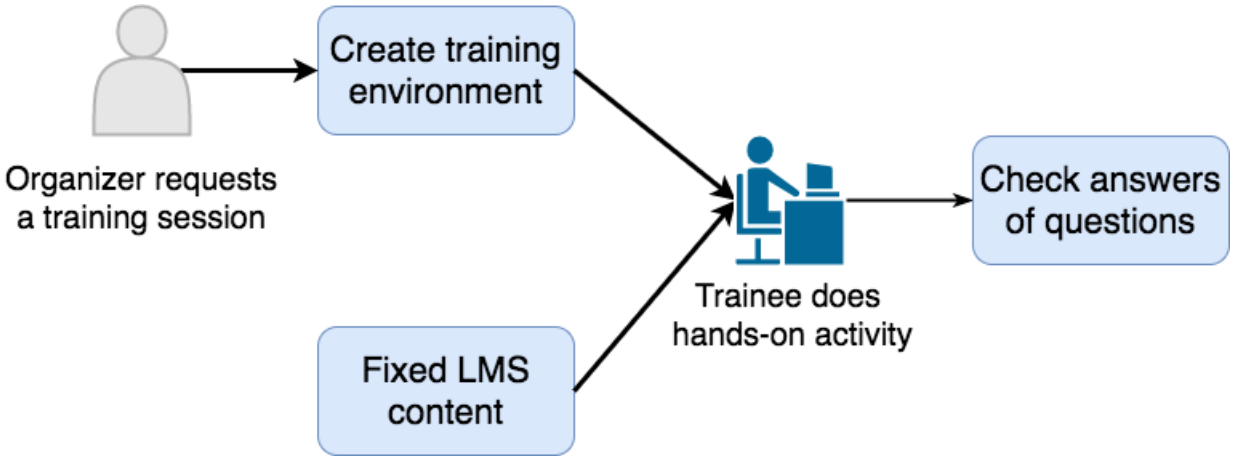


Figure 3.1: Current flow in a training session in CyTrONE

In JAIST, CyTrONE is developed with the purpose to address the drawback of current cybersecurity training. Currently, the system is as shown in Figure 3.1, which it only has one main process of creating training environment, which is now done by CyRIS. In this model, the content is prepared in advanced manually on Moodle web-based interface. The content is a list of questions for trainees to answer based on doing hands-on activities

with a cyber range. The result is evaluated by Moodle by giving scores to the answers. However, this system lacks three factors:

- Prepare training content automatically
- Perform attack automatically
- Bring a better interactive interface to learners

The main difference between an ordinary information technology training program and a cybersecurity training program are the attacks on a hypothetical target. In other words, in normal IT-related courses, there are only "static" objects, whereas cybersecurity training introduces the involvement of "dynamic" objects, which are the attacks at arbitrary moments throughout the training time. It leads to the requirement of automating attacking process, which also affects how the system should interact with trainees. Beside of hands-on environment issue, how to setup LMS to work with many trainees and training sessions also needs to be taken into account.

Therefore, in this research, we propose some improvements and extensions of the original CyTrONE system. They fit into two training models. The first one is illustrated in Figure 3.2. It is an extension of the original CyTrONE training model. In the current model, an organizer only interacts with the training system to instantiate a cyber range. The training content - which is a test including questions, still requires manual creation on Moodle interface by creating a new SCORM package and uploading it to Moodle. Thus in the new proposed model, I want that somehow questions can be received and put into Moodle automatically. That is the first requirement.

The second model is a totally different one. In the original CyTrONE, there is only one kind of studying over asking questions in a quiz. Therefore, I propose a new training model with cyber security incidents as Figure 3.3 shown. In this case, no learning content is required. An organizer is also not needed. A trainee can freely choose a vulnerability, start a training, has an interactive interface and a cyber attack is performed automatically. With this model, any trainee can learn independently, without any organization or event needed. To satisfy this new model, there are two requirements for performing a cyberattack and preparing a learning interface. Details about these additional requirements of the system are presented in following subsections.

3.2 (R1) Automation in Preparing Training Session

When mentioning training in IT, most people only think about machines with software packages for hands-on activities. They forget one important part, it is training content. It can be text, images or video records for introducing new knowledge, system utilization tutorial, or questions to evaluate learners. In the original CyTrONE framework, virtual machines are prepared by CyRIS. Therefore, there should be a program to add or upload training content automatically. In the original training program, our developing system uses two levels of questions - easy and medium for security awareness and training

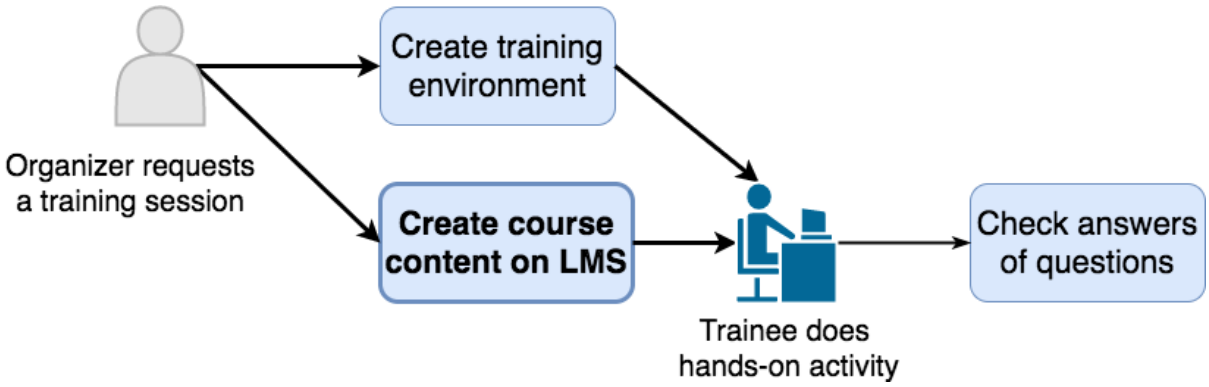


Figure 3.2: Training model 1: An extension of the original model. Content on LMS is prepared automatically after a request from an organizer

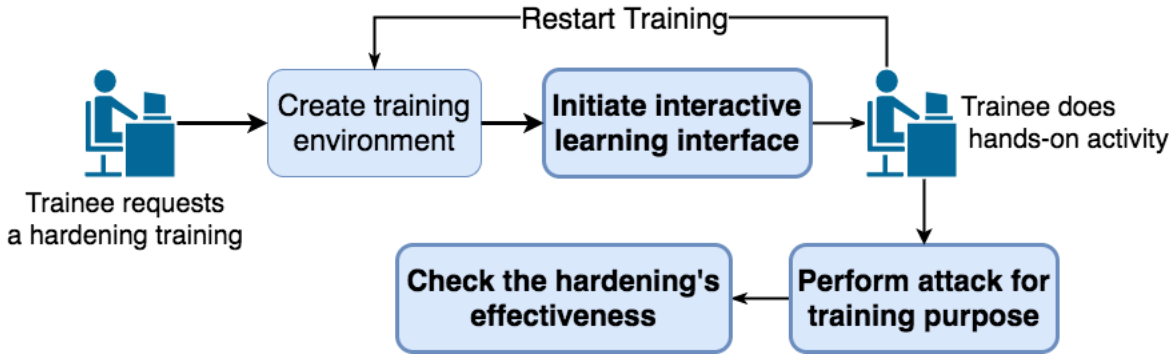


Figure 3.3: Training model 2: Training with cyber vulnerability: an attack - hardening model. A trainee try to harden a system against a vulnerability. The effectiveness is evaluated based on the result of the attack

program, following the standard and requirement of National Institute of Standards and Technology (NIST), US [29]. The combination of Moodle and SCORM is used in this research because of their characteristics:

- Moodle is an open-source software package, so it is possible to analyze it and add more functions on demand, such as accessing the database to analyze statistically the difficulty of questions or interacting between SCORM and Moodle to create isolated training content for each trainer.
- SCORM is a universal standard, so it is possible to work in another LMS than Moodle.

The training content interface needs to be prepared at the starting point of the session or in advanced. To reach the final goal of the project, this process is automatic. In this research, the requirement is preparing a SCORM package working on top of Moodle for every trainee.

3.3 (R2) Automation in Performing Cyber Attacks

As described in section 3.1, CyTrONE system has CyRIS to prepare cyber range. However, it can only serve a "static" session, that means everything is prepared in advance, and the task of the student is to get information from the cyber range, without any interaction or modification to change the state of the virtualized environment.

A modern cybersecurity training program requires events to happen in real time, or in other words, the state of the virtualized environment is changed during the session, with some attacking and defending activities. Thus, the requirement here is performing cyber attacks in an attack-defense or hardening training automatically.

3.4 (R3) Interaction Between the System and Trainees

The interaction between the system and trainees is the weakest point in current cybersecurity training programs. As systems are developed by information technology engineers, they only focus on IT-related features, sometimes they are over-engineering. Therefore, trainees usually find it complex to approach the system, while developers do not know if the system is simple enough to use or the educational output is good or not. As far as the survey about other cybersecurity training system, hands-on environments and learning environments (e.g. LMS, web-based quiz, question sheet) are independence. It is somehow inconvenient to switch between two environments.

This issue requires a pedagogical approach to identify a suitable method for improving the interaction between the system and trainees, so trainees can do a hands-on activity and use study material at the same time.

As mentioned in section 2.1.3 about interaction in distance learning, there are five types of interaction. Among them, we concern the forth one, learner - interface is the one which is largely affected by distance matter. However, current training systems usually do not realize it, and still use the same method and program design as traditional education. Therefore, this research aims to improve learner - interface interaction, thus it creates a user-friendly training interface, where learners take less time to understand how to work with the system and feel comfortable throughout using time.

Chapter 4

System Design

Addressing the weak points of the original CyTrONE system, this research proposes and implements the following system design based on the initial design of CyTrONE, which is shown in Figure 2.2. The new system design is illustrated in Figure 4.1. The main improvements are presented in the following sections.

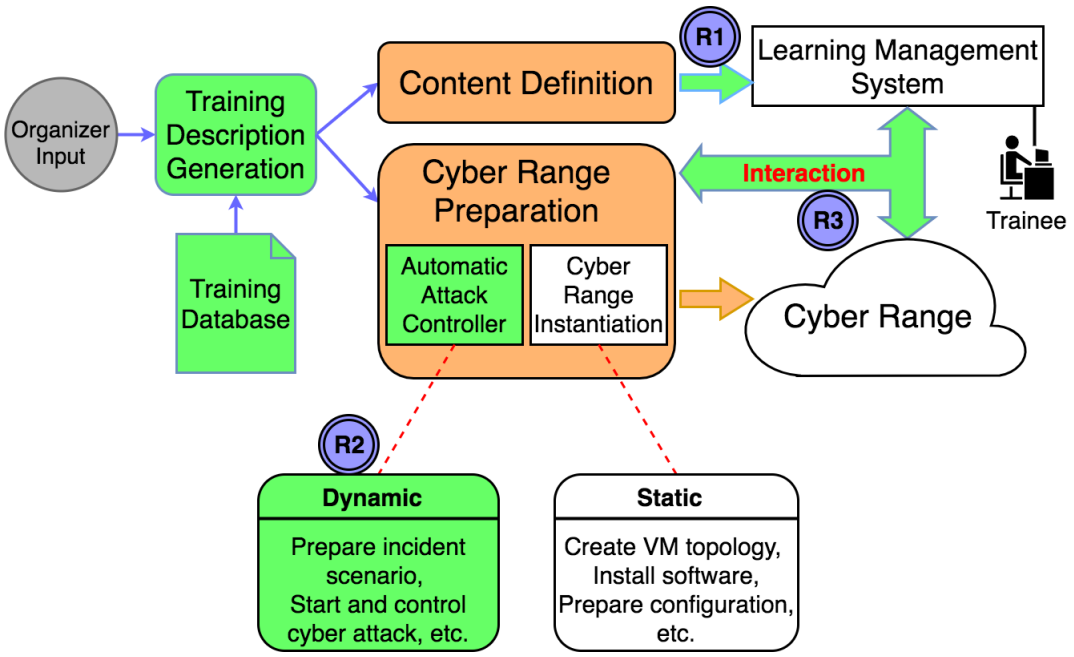


Figure 4.1: New system design for interactive and automatic training

4.1 Automatic Training Session Preparation

The idea to design this part of the system is very simple. Organizers of training programs always have a set of questions to examine participants for each training lesson. They are usually in human readable text format. Therefore, the role of this block is converting

that text format into an e-learning format. The workflow of the program is shown in Figure 4.2. Parsing by a converter, the training content is put into a SCORM package - which can be displayed by a web-based LMS. However, this whole process needs to be automatic, so the program should not only put content into a SCORM package but also integrate/upload the package to the LMS program, which in this research is Moodle. Detailed information of how this process is implemented is described in the section 5.1 about system function.

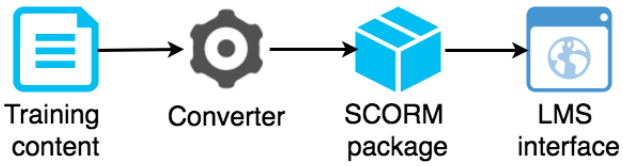


Figure 4.2: System design for automatic training session preparation

4.2 Automatic Cyber Range Preparation and Attack

Initially, there was only this CyRIS system for Cyber Range Instantiation in a static manner. This research proposes a new feature called "Automatic Attack Controller", which takes care of dynamic parts during a training session such as preparing incident scenario or starting and controlling cyber attack. In the new design, this block work based on the information in the training database, to send a corresponding request to CyRIS and prepare attack activities on demand. Figure 4.3 illustrates the workflow of Automatic Attack Controller inside the system.

First of all, the input to our system is an incident ID. It can be a STIX file or an ID in the training database. The input can come from a subsystem of CyTrONE or a third-party API. However, in the current state, the input of the system is a CVE id [30]. Based on the incident ID, the system should gather information from training databases.

In the initial plan, there are four main training databases: *Incident Report Database*, *Vulnerability Database*, *Exploit Database*, and *Instantiation Database*. The first one, *Incident Report Database*, we planned to hold the reference from an ID to related value in other databases. In the other words, this is the central database, which points to other databases. In the plan, we intended to match a vulnerability - CVE with an attack scenario - STIX file for training purpose, then other databases use CVE id as the search key. However, there is one objective reason to explain why STIX is not used now. It is because STIX is relatively new and being developed and updated, so it is hard to find STIX packages in real life situations or a guide to build a standard package reporting a cyber incident. We found it difficult to build a complete scenario for a cyber range attack from a STIX file. Therefore, we decide to skip this first database at this time, and currently, there are only 3 databases:

- *Vulnerability Database:* This is a list of every affected software versions by each CVE. Using this database, the system can know automatically which software package and

its version which need to be installed in order to demonstrate an exploitation. Vice verse, given a software version, it is possible to know if that version is vulnerable to any attack technique and a training session can perform an attack against it for educational purpose.

- *Exploit Database:* While *Vulnerability Database* stores a list of targets, *Exploit Database* has a list of attackers - which are attack techniques, hacking programs or proof of concept (POC) scripts. In overview, these two databases give the system a general scenario of any cyber attack: an attacker does something to a victim.

- *Instantiation Database:* This is a complementary database for *Vulnerability Databases*. In the initial design, there is no such database. However, during the development of the system, it is realized that preparing a victim environment is not as easy as installing a software program by one simple command. Some limitation, namely lacking of old software versions on public repositories, dependencies of programs and system-specific configuration. Because of these reasons, *Instantiation Database* is designed to fulfill this limitation. Following its goal, this database should support and guide the *Cyber Range Instantiation* block to install the software packages required by the cybersecurity incident, along with performing environment setup. At this moment, the information from this database work with CyRIS. The detailed information is provided in the section about functions.

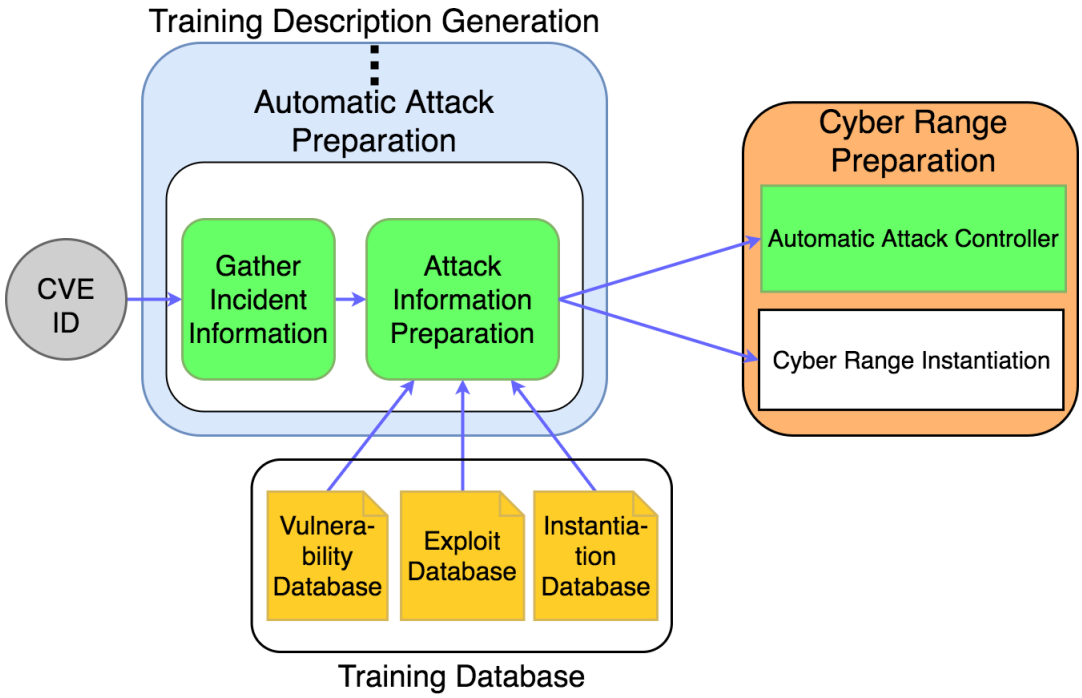


Figure 4.3: System design for automatic cyber-attack

With the support of these databases, Attack Information Preparation can create inputs

for Cyber Range Preparation block. They are description file for CyRIS for setup cyber range environment and attack description for the attack controller.

4.3 Interactive Training over Web-based LMS

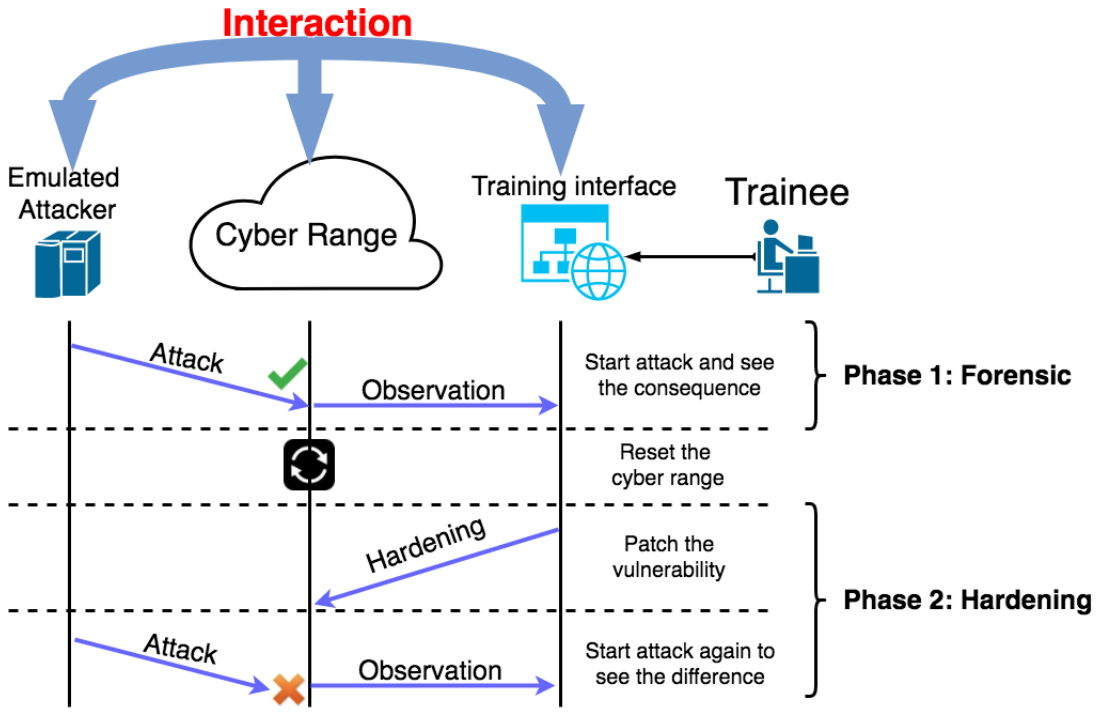


Figure 4.4: System design idea for an interactive training

Inside our system, there should be some components to match *R3) Interaction between the system and trainee* . This research proposes an idea of how a learner should work with the system, which is illustrated in Figure 4.4. There are two general phases in a training session: Forensics and Hardening. First, the emulated attacker perform an attack against a cyber range environment having one or many vulnerabilities. The attacker here must be *"emulated"*, thus it satisfies the ultimate goal of an automatic training system. It does not require security experts or white-hat hackers, which means taking less money and time while improving flexibility and proactivity. The trainee, over a training interface - a web-based LMS, receives the result of the attack and is able to access the cyber range to investigate more. This is the end of Phase 1. After that, before going to Phase 2, the cyber range should be reset if it is needed, since some cyber attacks damage or change the state of the system permanently. The difference between Phase 1 and Phase 2 is that in the later one, the learner is required to patch the vulnerability on the cyber range. This activity is performed first. The idea here is the student known what is the problem at this point, so he/she should do some actions toward the cyber range to fix the problem.

Then the attack is performed again to check the result of the trainer's work. The system should be able to detect if the attack is successful or not.

Following this idea, a training model is designed and illustrated in Figure 4.5. It includes various steps:

- Step 1: Choose a CVE: From a list of CVE ID, a trainee can choose 1 to study with it. This list should be got from the Incidents Report Database, where stores a list of CVE incident which the system is ready to reproduce. After choosing an id, a short description of the CVE is showed to the learner to understand background information around the vulnerability.

- Step 2: Cyber range creation: A cyber range is created by CyRIS from the information in Training Database as described in section 4.2.

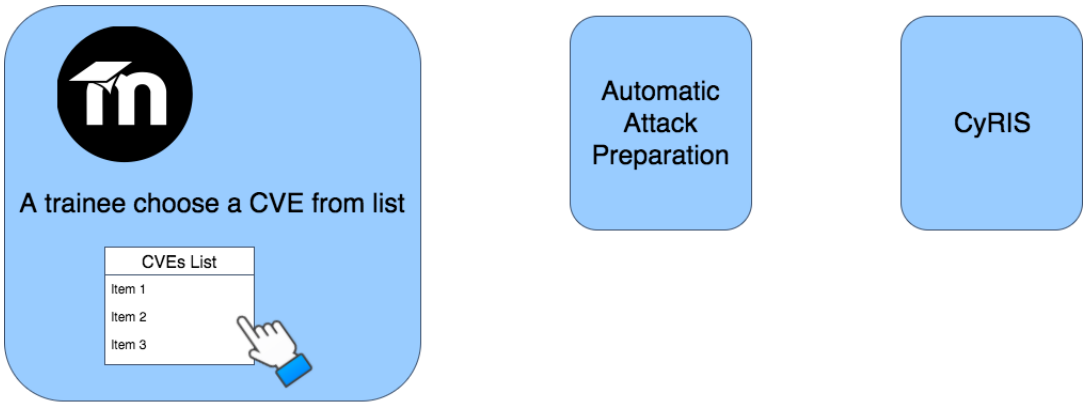
- Step 3: Perform cyber attack: The trainee triggers an attack by clicking a button on the website interface. The Automatic Attack Controller controls the attack and informs the trainee if it is successful or not.

- Step 4: Incident forensics: In this step, the learner look at the system to see what happened. In the future development, there should be a guidance from the system interface to guide the learner what to do.

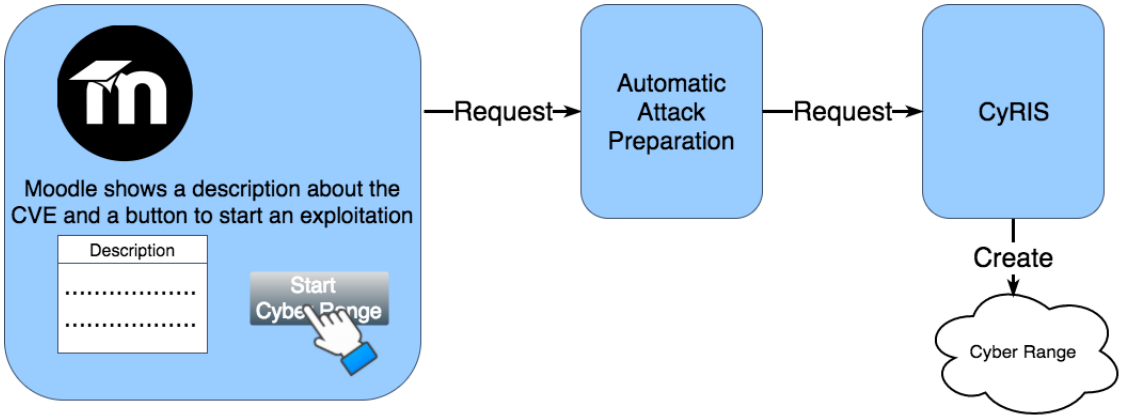
- Step 5: Restart cyber range (Optional): In some cases, a cyber attack changes the state of the cyber range permanently (e.g. change configuration files, replace binary files, delete operation system files, etc.). Therefore, it is better to reset the training environment before doing next steps.

- Step 6: Hardening cyber range environment: This is the starting point of Phase 2 in a training session. The learner should patch the vulnerability successfully by connecting to the cyber range and perform some actions with virtual machines inside.

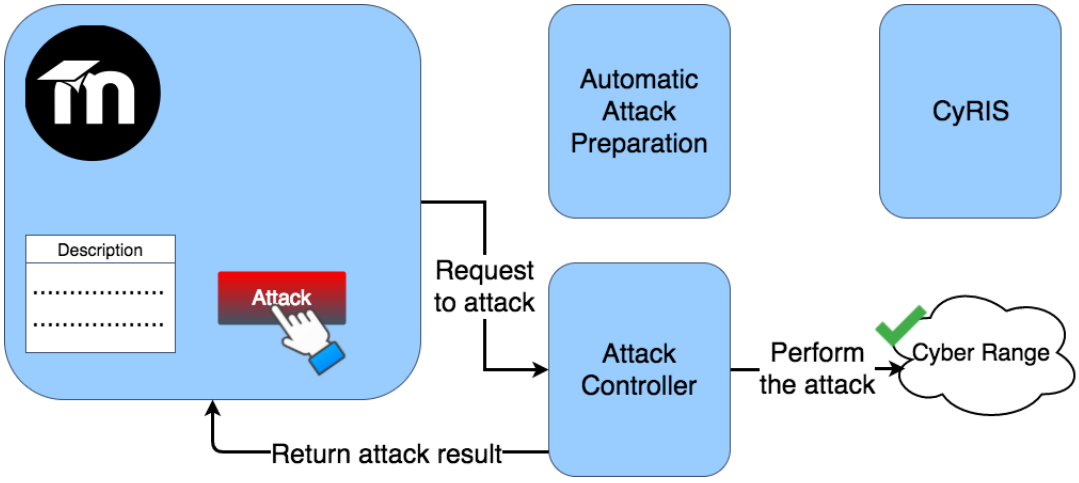
- Step 7: Failed attack after successful hardening: After the trainee finishes the hardening, the attack runs again to test if what the trainee did in Step 6 is correct or not. Same as step 3, the system is able to answer if the attack is done normally or not. This time, the answer should be a failed attack.



(a) Step 1: Choose a CVE

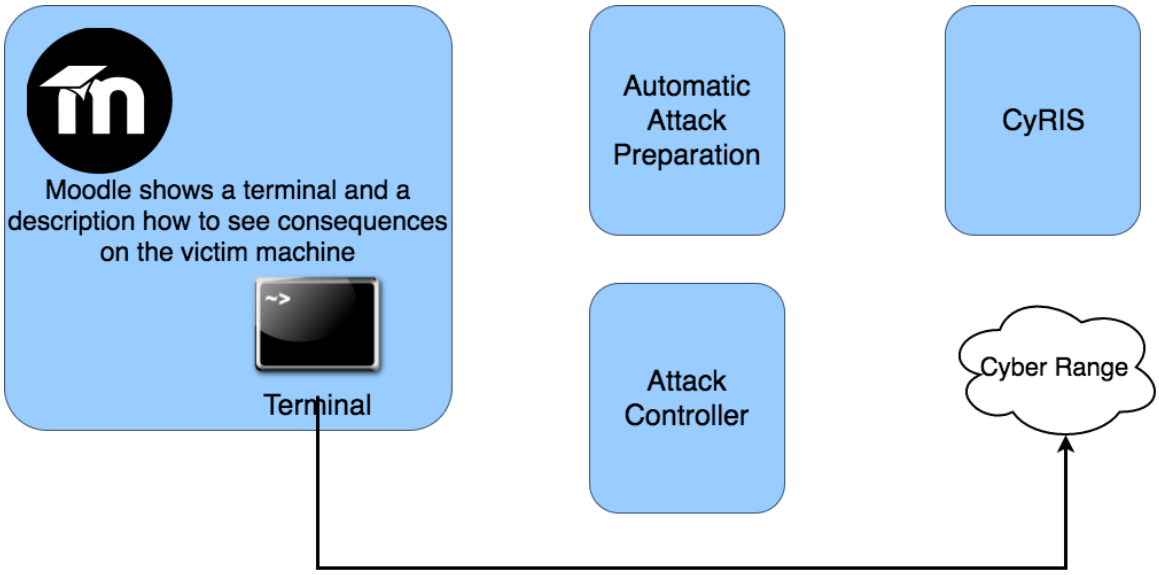


(b) Step 2: Cyber range creation

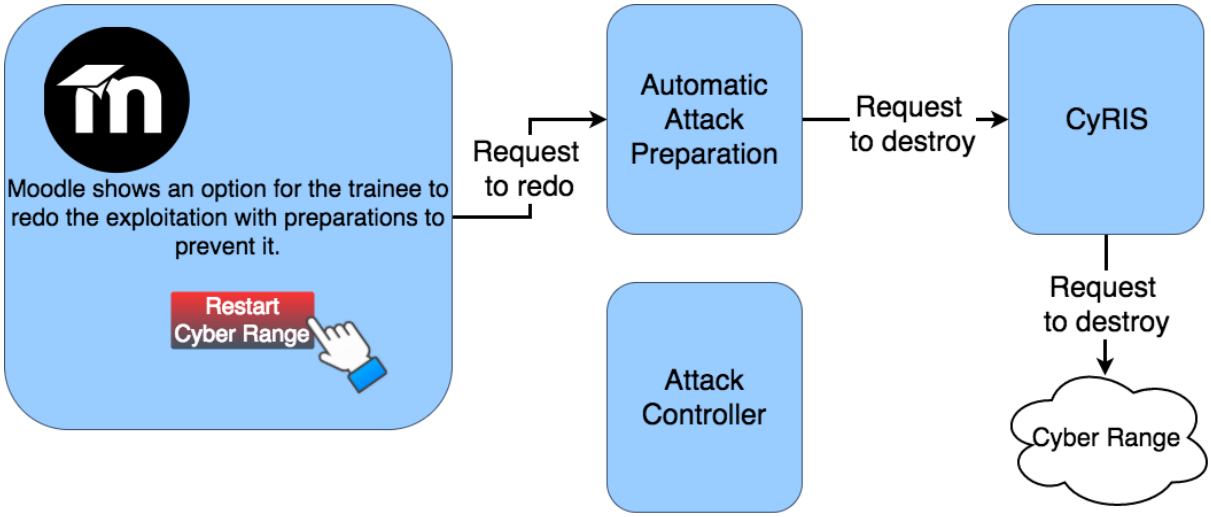


(c) Step 3: Perform cyber attack

Figure 4.5: Interactive training model

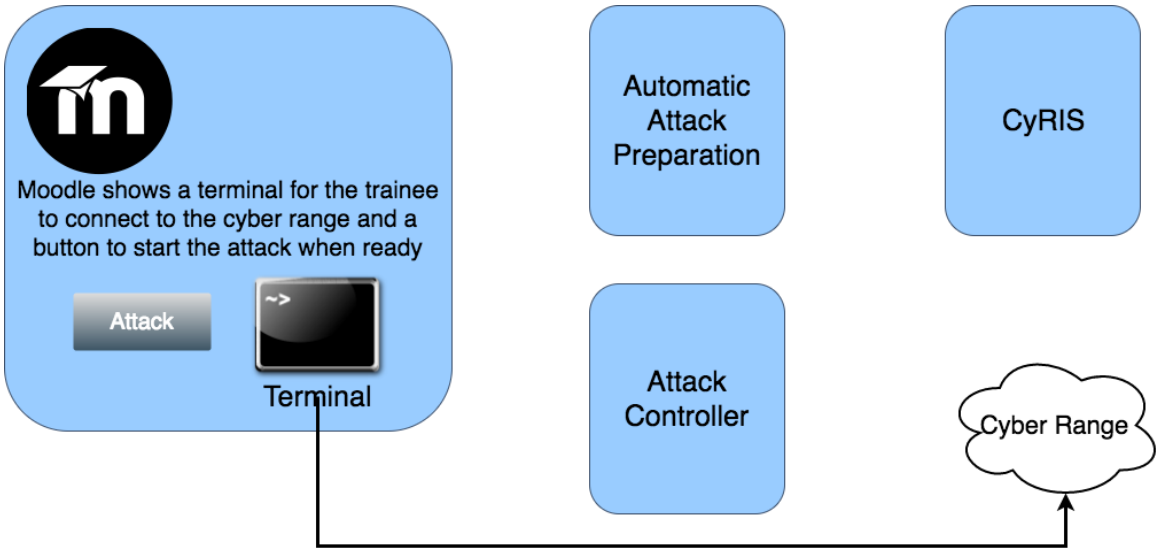


(d) Step 4: Incident forensics

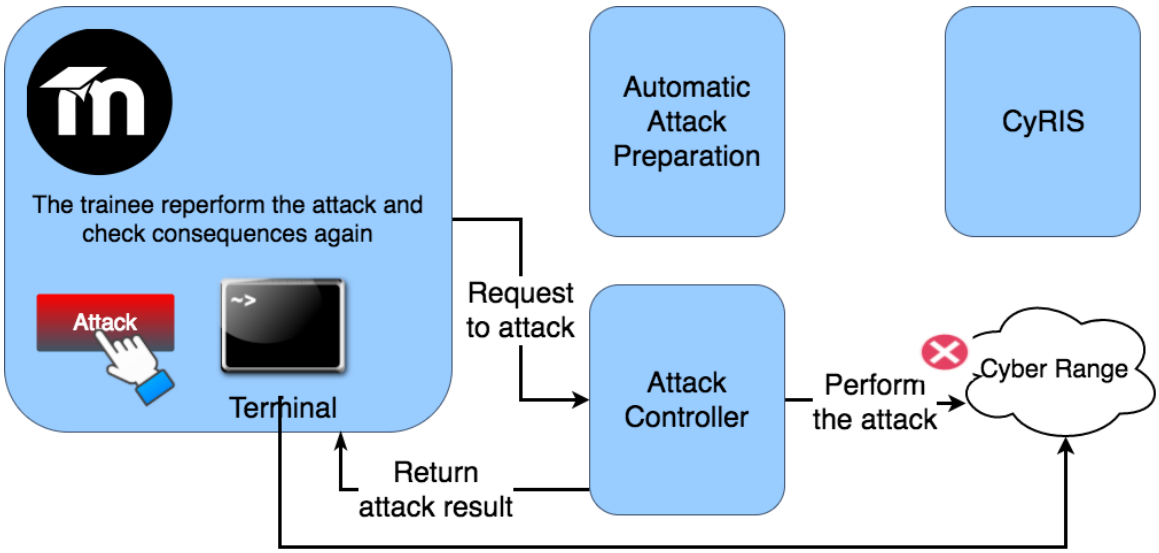


(e) Step 5: Restart cyber range(Optional)

Figure 4.5: Interactive training model



(f) Step 6: Hardening cyber range environment



(g) Step 7: Failed attack after successful hardening

Figure 4.5: Interactive training model

Chapter 5

System Functions

This section presents the implementations of the designs introduced so far in Chapter 4.

5.1 Automatic Training Session Preparation

For preparing training sessions from contents, a tool called `cnt2lms` was developed. It is made available on GitHub at `https://github.com/crond-jaist/cnt2lms`. An overview of `cnt2lms` is provided in the figure 5.1 below. Generally, this program reads a training content file, in YAML format - which is user-friendly and then creates a SCORM package - which can be used on Moodle. A user guide is also released at `https://github.com/crond-jaist/cnt2lms/releases/download/0.2.3/cnt2lms-0.2.3-guide.pdf`. For tool's utilization, this guide is very informative and provides hands-on tutorial.

In the next subsections, the training content file format and workflow of the program are discussed.

Figure 5.1: Overview of cnt2lms [32]

5.1.1 Training Content Format

Currently, a training content file consists of basic information about the training session, and a list of questions. It is written in YAML format since this format is both human-

readable and machine-readable.

- *Training session information*: First of all, it has a training id - which is unique in a system. Besides of it, a description and a header are provided to show learners a brief information about the training. The level field is added in order to show the difficulty of the question set in a multi level training session.

- *Questions*: This is the main part of a training content file. There are 2 possible options for question type. They are *choice* and *fill*, which learners can answer by fill in the blank or choose from multiple choices. Hints are also provided to help learners if they need.

5.1.2 Implementation of `cnt2lms`

`cnt2lms` is coded in Python. It runs in command-line Linux environments. Figure 5.2 shows how this program runs, which it contains of three components.

- *yamlParser.py*: Consists of main functions of the program. It reads a configuration file (which has predefined settings such as input and output file of the system, a location of Moodle, etc.), then parses the training content in YAML format. From the content, it adds information to a SCORM package. The SCORM package is a template which is prepared in advance. It has a frame of quiz format and can connect with Moodle over its API. Therefore, when learners do a test based on this SCORM template, their results are recorded and stored by Moodle. Since the template has the frame of a standard SCORM format, `yamlParser.py` just add information in suitable place, all training contents are displayed properly. In the last step, everything is packed into a SCORM package, which is in `zip` format.

- *copyToMoodle.py*: Moodle maintains a repository for storing its resources, including SCORM packages. Therefore after moving a SCORM package here and setup Moodle to use it, it can be displayed on Moodle.

- *cnt2lms.py*: This is the controller of the whole program. It calls these two other ones. Other components of the system can call this script to use `cnt2lms`.

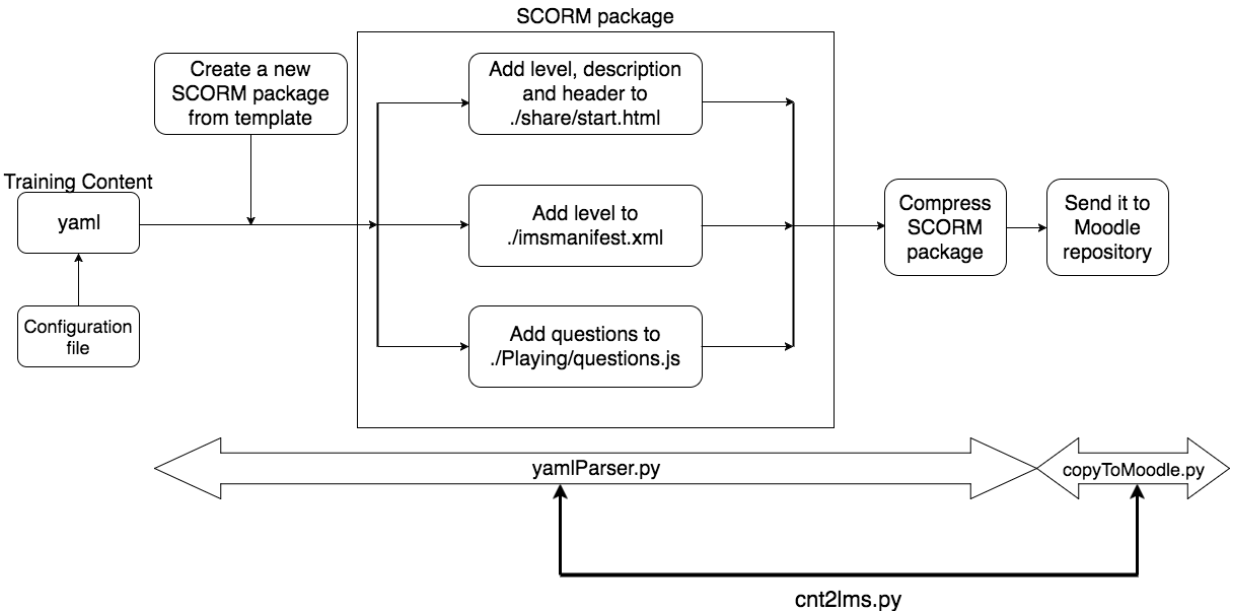


Figure 5.2: Workflow of `cnt2lms`

Figure 5.3 illustrates the input - a training content file, and the output - a SCORM package displayed on Moodle. It is clear that the input is simple and easy to write by both human and computer program, whereas the output is more familiar with trainees in a training. The output SCORM package has HTML interface, allows trainees interact in graphical environment. It also provides necessary functions such as connecting to cyber range over a terminal, show hints and submit answers. `cnt2lms` allows organizers of training sessions prepare the learning material on Moodle automatically and programmatically, thus saving time and effort.

5.2 Automatic Cyber Range Preparation and Attack

5.2.1 Databases Implementation

As mentioned in section 4.2, this research created three databases for training purpose: *Vulnerability Database*, *Exploit Database* and *Instantiation Database*. This section discusses how these databases are implemented in practice.

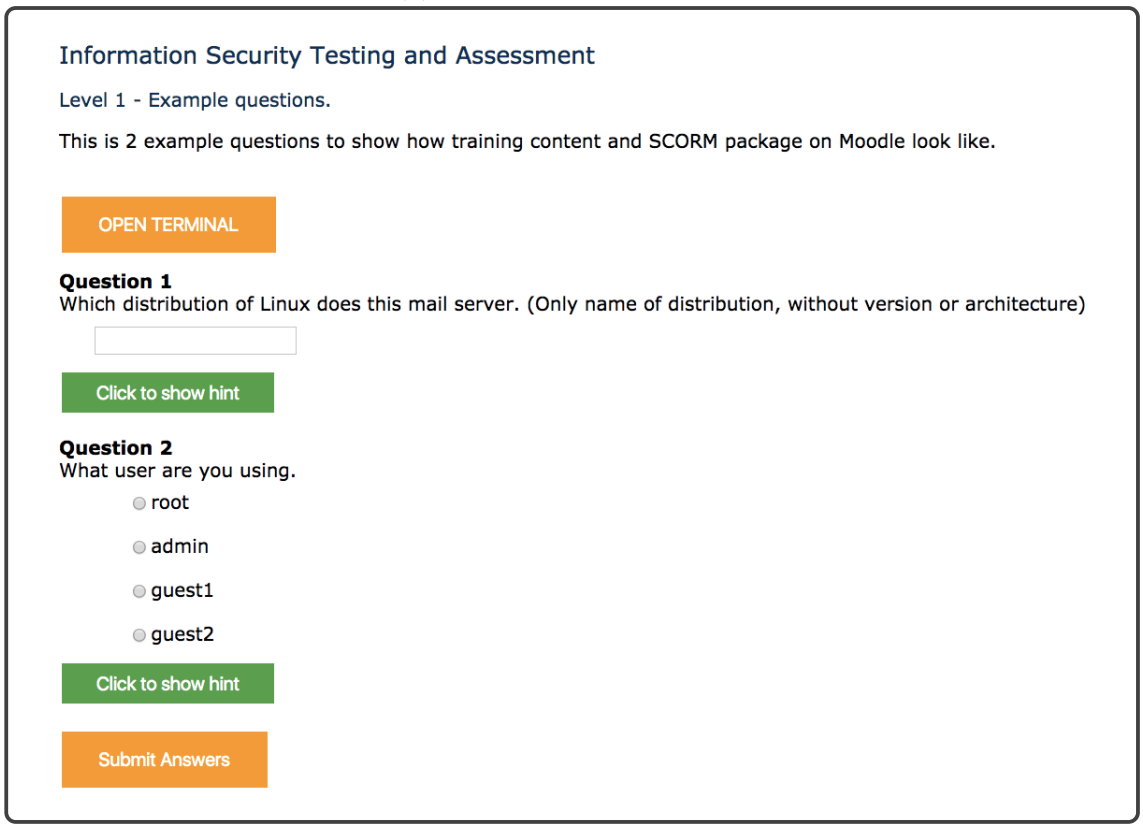
- *Vulnerability Database*: For building this database, a storage of detailed and structured information related with CVE should be used. In this research, CVE Details [33] is used. In this data source, for every CVE id, there are details about taxonomy, severity, products affected, references and exploit modules. They are enough to provide a complete view to any vulnerability. Moreover, the most important reason for choosing it is that all information is ordered in the same data structure. Every web page - each for any CVE, has the same HTML structure. Therefore in this research, we choose to crawl every page by a popular tool called Scrapy [34]. From that, the

```
---
- training:
  - id: L1-E1
    description: Example questions.
    header: |
      <p>This is 2 example questions to show how training content and SCORM
package on Moodle look like.</p>
    level: 1

    questions:
    - id: E1-1
      type: fill
      content: Which distribution of Linux does this mail server. (Only name
 of distribution, without version or architecture)
      answer: Ubuntu
      hints:
        - hint: You can use the command <code>uname</code> to find out OS de
tails.
        - hint: <code>$ uname -r</code>
        - hint: An alternative solution is to get the required information f
rom the <code>/proc/version</code> file.

    - id: E1-2
      type: choice
      content: What user are you using.
      choice: "\"root\", \"admin\", \"guest1\", \"guest2\""
      answer: guest1
      hints:
        - hint: "Have you ever as yourself: Who am I."
        - hint: <code>$ whoami</code>
```

(a) Training content format



(b) SCORM package in Moodle

Figure 5.3: An example of a test in training content format and on Moodle interface [32]

system owns a table of every software packages which have vulnerabilities, classified by CVE id. Figure 5.4 shows some queries from this database.

- *Exploit Database*: While the *Vulnerability Database* stores targets of cyber attacks, *Exploit Database* has tools for exploiting and performing cyber attacks. Since CVE Details also has a list of Metasploit [35] modules (if they are available for that CVE id) and POC (Proof of Concept) codes. Gathering information from that website brings a huge amount of exploit tools to this database. At this state of the system, only Metasploit module names are used to call corresponding modules. The POC codes are not used because 1) They are not downloaded to local and 2) Many POC codes are tutorials for exploiting a specific vulnerability, which cannot be used by machines. However, since the database has links to get these scripts, it has the potential to be used in the future.

- *Instantiation Database*: In this database, each CVE id links to a script to instantiate a training environment. When CyRIS is called to create the cyber range, using a function of CyRIS called *Custom Install* to copy scripts into virtual machines and run them when machines start after cloning. When implementing the function which setup training environment for attack-defense cyber security training, we realized some specific issues related to this:

  - Vulnerable software versions are usually removed from official download sites or public repositories since they are old and dangerous for systems installing them.
  - Application packages have dependencies. In case of outdated packages, it is a real problem because one old package requires many other old ones, which are replaced by new versions by default.
  - Installing using package managers such as `yum, apt` requires Internet connection, while cyber ranges for cyber security training should be isolated from outside to prevent unintended consequences in training. Beside of it, downloading from Internet prolongs instantiate time, which this research project is trying to minimize.
  - Compiling packages from source code takes a significant amount of time, which can be eliminated by creating binary files in advanced.
  - Many files and actions for preparing the environment makes description file of CyRIS long and complicated.

  Because of these issues, in this research, a single script to prepare everything is prepared in advanced. All files and scripts for setting up a virtual machine is packed into only one executable script by a tool called `makeself` [36]. For using this tool, there are two components. First is a directory which consists of all needed files. Notice that all files are ready to use, every package is compiled already, and just copying them in suitable directories is enough for running programs. Execution time is therefore reduced dramatically. In some cases in this research, it reduced

cve_id	product_type	vendor	product	version	update	edition
CVE-2013-1070	Application	Ubuntu	Metal As A Service	1.2		
CVE-2013-1070	Application	Ubuntu	Metal As A Service	1.4		
CVE-2013-1069	Application	Ubuntu	Metal As A Service	1.2		
CVE-2013-1069	Application	Ubuntu	Metal As A Service	1.4		
CVE-2013-2186	OS	Ubuntu	Ubuntu	10.04		LTS
CVE-2015-1322	Application	Ubuntu	Network-manager	0.9.8.7		
CVE-2015-2150	OS	Ubuntu	Ubuntu	12.04		LTS
CVE-2015-2285	Application	Ubuntu	Upstart	1.13.2-0ubuntu7		
CVE-2015-2285	Application	Ubuntu	Vivid	15.04		
CVE-2015-5479	OS	Ubuntu	Ubuntu	12.04		LTS

Figure 5.4: Vulnerability Database: detailed information about what software versions are affected by each CVE

from over 10 minutes to less than 1 minute, so that the total time also decreases. The second component is a script to move files into their intended position and do system configuration if it is needed, such as start services, add firewall rules or change startup services. In other words, with a set of software binary files and a automation setup script, a vulnerable virtual machine for training purpose can be setup, but in this case, all of them are packed into one script, make it easier for distributing and running.

5.2.2 Starting and Controlling the Attack

To satisfy *(R2) Automation in performing cyber attacks*, the system should be able to perform cyber attacks without human interaction from cybersecurity experts. In the previous section, how the training database is setup is presented. This section describes how the system uses these databases to setup and perform cyber attacks. Moreover, because this function is integrated into the interactive interface, here only details of each component are showed. The explanation about how everything is orchestrated is mention in the next section, *Interactive training over Web-based LMS*.

- **Starting CyRIS**: CyRIS is a tool which has an advantage that the input and output of the program is easy to prepare and use since they are text files. The input of CyRIS is a description file about how the cyber range should be, whereas there are three main outputs. The first one is a cyber range. The second one is a cyber range description file including detailed information of the cyber range, such as name of each virtual machine, network address. The third output is an email content which is intended to send to the training session's organizer. This email has login credentials of each instance inside the cyber range. Taking advantage of CyRIS, in order to setup a hands-on environment for incident training, in this research, the system get information from the training database, combine with a

template of CyRIS description file, to create a description file to start CyRIS with is input. Inside the description file, the most important part is the location of the instantiation script, which is queried from Instantiation Database. CyRIS runs this scripts on cloned machines, called victim machines to install vulnerable software packages, then trainee can try to harden these machines. Currently, there is only one victim machine for each instance. The output file of CyRIS is stored and used later by Atomatic Attack Controller. This file has address of the victim, so attack tools can use it to perform attack against victim. Another file which has login credentials is used by trainees to access cyber range.

- **Restarting CyRIS**: This is a function to support training process. It is the combination of two actions, which are destroying the current cyber range and starting it over again. One better solution is using a feature of KVM called snapshot, which it is able to save the state of the virtual machine before giving it to the trainee, then when it is needed to restart the training session, the saved state is restored. However, this feature is not implemented yet.

- **Using Metasploit to attack**: Metasploit is the most popular framework for exploiting vulnerabilities. It is usually used for penestration testing, then it is suitable for training purpose. From the point of view of this research, Metasploit has three main handy features, which we employ:

  – It integrates many modules for cyber attack, which can be used for reconnaissance, exploiting vulnerabilities or privilege escalation. The attack modules are updated frequently. Therefore, using Metasploit is a good solution to start with automatic cyber attack.

  – Since this is a framework, attack modules has almost same format, which means they share same options and command. Thus it is possible to generalize attack activities.

  – It supports logging and scripting running, so the program can be run programmatically and the result of the program can be used for further actions.

  First of all, a script is created as an input of Metasploit. This script sets up the attack options for Metasploit, which are the name of vulnerability, the IP address of the victim and the log file. Figure 5.5 show an example of an input script for Metasploit. After CyRIS instantiates a cyber range, the cyber range description is used to set the target of the attack. The vulnerability is searched in Exploit Database to match between CVE id and Metasploit module. The log file holds the output of Metasploit after running, and can indicate if the attack is successful or not. Therefore, the output is parsed and analyzed to give the attack result to the trainee, to say if it is successful or failed.

```
# Specify log file
spool /home/dev/metasploit_automation/metasploit_output.txt
# Choose attack module
use auxiliary/scanner/ssl/openssl_heartbleed
# Setup options for the attack
set verbose true
set RHOSTS 112.1.1.2
# Run Metasploit
exploit -z
```

Figure 5.5: An auto generated script for Metasploit

5.3 Interactive Training via LMS

5.3.1 Interactive Web-based Interface

In order to bring a better experience to learner, this research develops a web-based interface for security training. This interface run on top of Moodle and inside a SCORM package. Moodle is an e-learning platform, so it owns some basic interactive features such as user login, result and learning progress management. On top of it, we build a graphical interface specifically for cyber security training. Following the interactivity requirement, a layout was designed as shown in Figure 5.6. The main components of the interface include:

- A list of vulnerabilities to choose from.
- Buttons for start and restart training session and a button for performing cyber attack against the cyber range.
- A terminal to connect to cyber range.
- A text field for a short description about the selected vulnerability.
- A text filed for training tutorial.

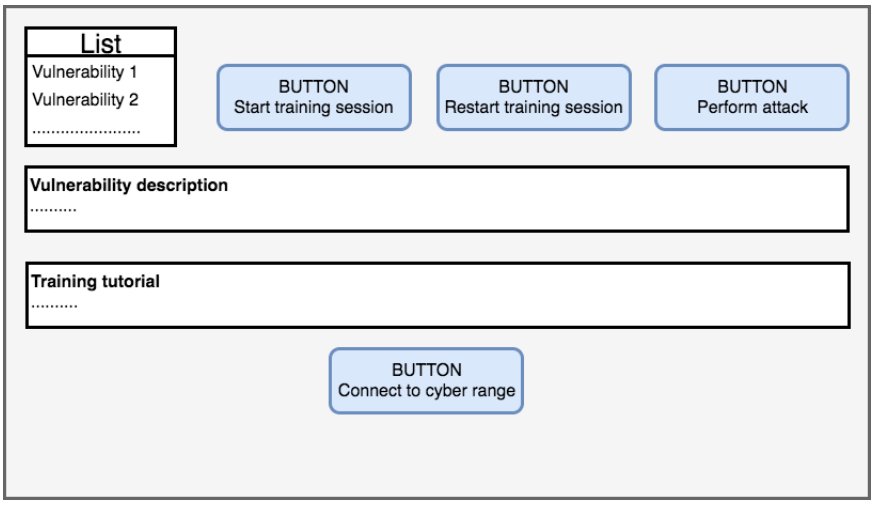


Figure 5.6: Layout of the interactive interface

From this layout, an actual web-page is built using HTML and Javascript, with PHP as a back-end language. Figure 5.7 is a screen shot of the actual web-page. In general, it follow the layout architecture, with the description is generated when a vulnerablity is chosen. The instruction is fixed at the current state because now every training session follow a same flow, which has two phases as mentioned. A progress bar is added to let the learner know the progress of CyRIS in instantiating the cyber range, since the instantiation usually takes few minutes and the system is only ready when this work finished. The terminal is opened in a new browser tab when the button is clicked, so the learner has more space to look at the tutorial. When the start button is press, Metasploit is called in background, and then a pop-up appears to inform if the attack is successful or fail.

This layout and its implementation follow the suggestion in [6] for better interactive interface such as important information on the top and in the center; buttons with bright color to get attention; and orientation clues for every training session.

5.3.2 Back-end Functions

Running in the background of the web-based interface are many functions running on three machines: LMS machine, Attack Controller and CyRIS master machine. Figure 5.8 is a diagram of how programs on these three machines work together to serve the interactive interface which is mentioned above.

- **Moodle machine**: There is a set of scripts written in Bash shell script and PHP to receive and process input from trainee over web-based interface. They serve all requirements which mentioned above, such as opening a terminal on web browser, creating and restarting cyber range environment, performing cyber attack for training purpose and checking the progress of the training activities like cyber range creation and attack performance. Programs on this machine support the graphical interface by calling other programs on the *Attack Controller* and *CyRIS*.

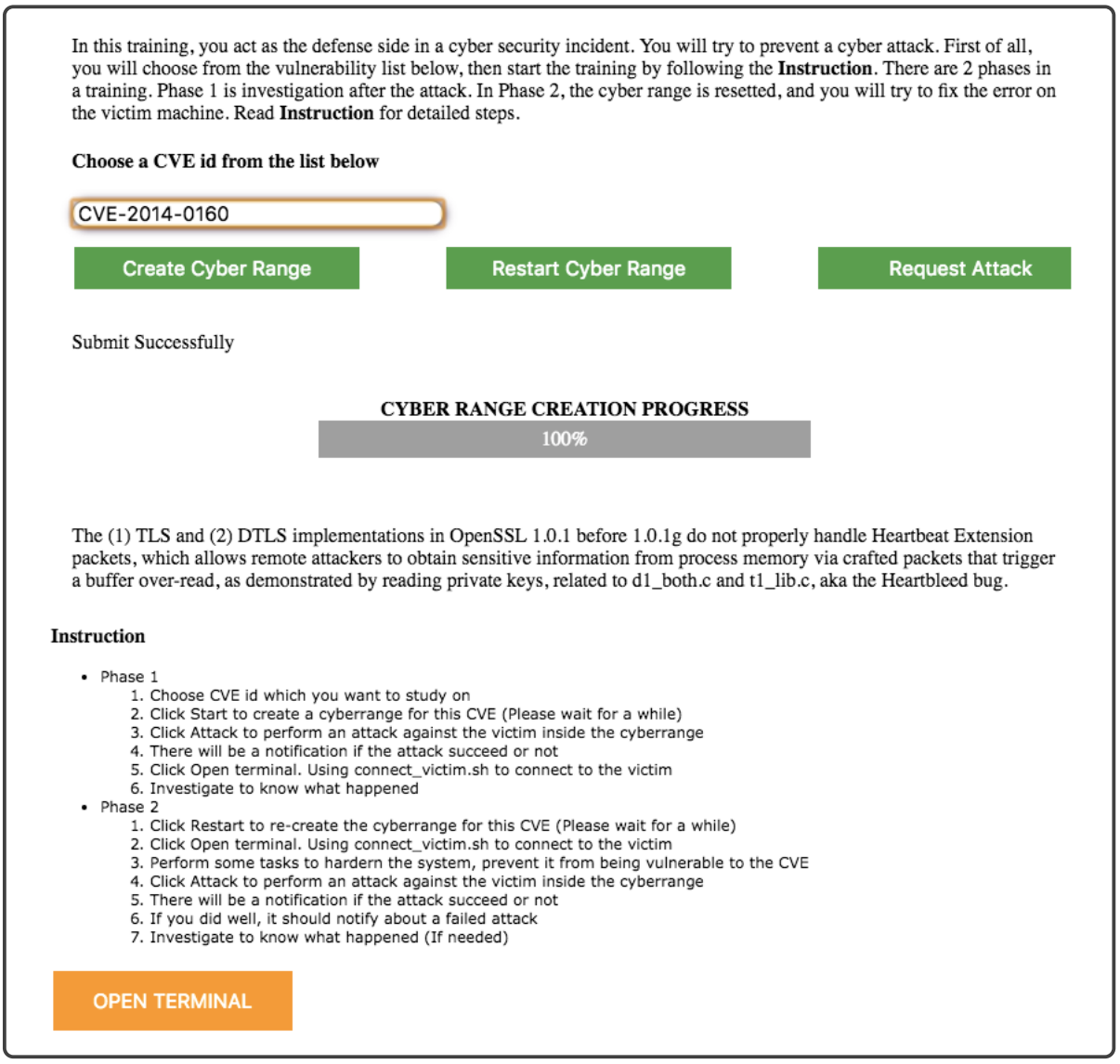


Figure 5.7: Actual web interface: Include all components in the layout and add a progress bar for instantiating cyber range and the terminal is opened in new new window

- **Attack Controller**: This machine includes two components. The first one is the Training Database. It feeds a script in Moodle machine vulnerability description, and the second one - Attack Controller vulnerable environment information and attack technique. The Attack Controller does two main tasks. It controls CyRIS in creating cyber range for training purpose. The progress of creation process is sent back to Moodle machine and displayed on a percentage bar. Besides that, it performs a cyber attack against the prepared cyber range and then gets the result. This result is showed to the trainee over Moodle interface, so he or she knows the result of the attack.

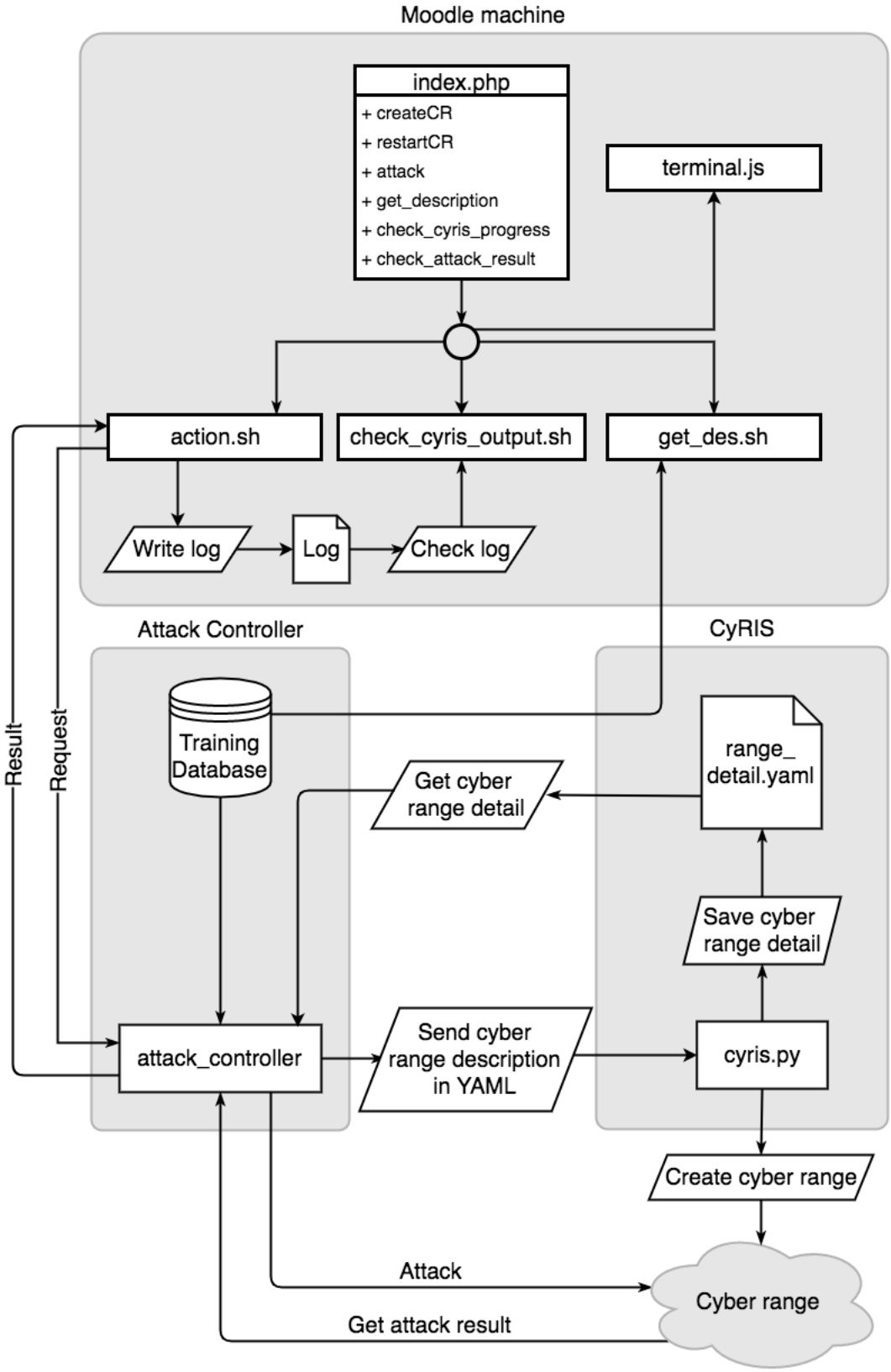


Figure 5.8: Module diagram for the interactive web-based interface

Chapter 6

Evaluation

6.1 Feature Comparison

Table 6.1 is the comparison between the improved system in this research with the original CyTrONE framework and Alfons [38]. First of all, let have at look at Alfons – a tool for constructing mimetic network environment, so Alfons owns some common features to satisfy its mission: cyber range creation, the ability to setup network connections between virtual machines and allowing access from outside to the cyber range. These features also appear in CyTrONE. It is clear that since our system is developed based on CyTrONE, it inherits all feature, such as setting up the cyber range and building virtual network topology.

There are two other functions which also appear on both systems: setting up vulnerability and accessing to the cyber range. However, in the new system, they have some improvements. The vulnerability is set up differently (symbol (*) in the table 6.1). CyTrONE based on CyRIS, which uses package management tools such as `apt` and `yum` to install packages. The Attack Controller uses another method, *Instantiation Database*, which installs vulnerabilities from prepared binary files. This method is better in many cases since it takes less time and can install many vulnerable packages which the former method cannot. The second difference is the way to access the cyber range (symbol (**) in the table 6.1). Alfons and the original CyTrONE did not develop any special tool for supporting trainees connecting to the cyber range. They lay on the capability of the hypervisors to access virtual machines. Therefore, in order to improve this limitation, along with the terminal windows on the interactive web interface, a small plugin is added into the Attack Controller, so whenever a cyber range is created, a script to connect to the cyber range is also generated. Thus, as explained in the previous chapter, it is easier to connect to the cyber range than in the original CyTrONE.

Compared to Alfons and the original CyTrONE, the improved system adds a unique ability: performing cyberattacks. In Alfons, since it is developed for the general purpose, a specific feature like this one is not included. On the other side, at the initial state, CyTrONE only supports "static" features, while everything is set up in advance before starting the training session. Hence, CyTrONE only has hands-on activities of investigat-

Table 6.1: Feature comparison between the improved system and others

	Machine		Network		Perform Cyberattack	Training Content Setup
	Cyber range creation	Vulnerability setup	Virtual network connection	Cyber range access		
Alfons	○		○	○		
Original CyTrONE	○	○	○	○		
Improved CyTrONE	○	○(*)	○	○(**)	○	○

(*) and (**) are new improvements on the improved CyTrONE

ing the post-attack phase of a cyberattack. A real cyberattack happening in real time is not available. Now, in the improved system, real cyberattacks perform on demand from trainees during a training session.

6.2 Training Environment Creation

Since the vulnerable environment creation running on top of CyRIS, the total creation time is an important number when evaluating. As mentioned in Implementation section, using Instantiation Database helps to minimize the overhead of the new training model creation time comparing with the original system. Therefore, after implementing, the system creation time is compared with the original system and with installing software packages by the traditional method of compiling source codes, as shown in Table 6.2. Note that since the required packages are usually removed from public repositories because they are old and contain vulnerabilities, using a package manager to download them from the Internet is not a working way.

On a Fujitsu PRIMERGY (S26361-K1272-VXX) server with 2 x Intel(R) Xeon(R) CPU E5504 @ 2.00GHz, 48 GB RAM, the new system only adds 23 seconds overhead, which is reasonable for a training system while it is usually prepared in advance. Note that preparing a vulnerable environment by traditional method takes almost 2.7 times more time, which makes it unusable in practice. Therefore, this system can be used with the current system for defense training. The automatic attacker also only takes 20 seconds more, so learners can perform it many times during a training session without interrupting their learning activity.

Table 6.2: System creation time: The overhead of new system compares with original system

Tasks	Average creation time	
	(s)	(%)
CyRIS with base VM	475.3	100 %
CyRIS + Prepare vulnerable machine (from source code)	1273.2	268 %
CyRIS + Prepare vulnerable machine (by Instantiation Database)	498.6	105 %
CyRIS + Prepare vulnerable machine (by Instantiation Database) + Perform attack	518.3	109 %

6.3 Authentic Activity Validation

In section 2.1.4, a list of characteristics of authentic activity from Reeves et al [14] is showed. Authentic activity is believed to bring better outcome to students. Therefore, the training system is developed for satisfying these characteristics. Table 6.3 explains details of the comparison between theory and implementation.

Table 6.3: Satisfaction of the system with authentic activity characteristics

Characteristics of authentic activities	**Satisfied**	**Explanation**
Have real-world relevance	◯	The input of the training is a real-world incident or vulnerability.
Ill-defined, requiring students to define the tasks and sub-tasks needed to complete the activity	◯	The training scenario only has information about the type of vulnerability, without any guidance or limitation of methods to patch the error.
Comprise complex tasks to be investigated by students over a sustained period of time	◯	The training aims at people who have some knowledge about IT and cyber security. It requires patching a vulnerability without any hint, except the name of the vulnerability. Therefore, any trainee taking this course should spend some time on study about it, thus he/she cannot solve it immediately
Provide the opportunity for students to examine the task from different perspectives, using a variety of resources	◯	Since the training does not has any limitation of methods to solve the issue, students can try any techniques and solution they know, as long as it prevents the attack.
Provide the opportunity to collaborate		Each cyber range is isolated for one learner, so the system does not support any special way of collaboration between students, except in-class communication and forum feature of Moodle.
Provide the opportunity to reflect	◯	With the restart feature, a student can do hardening many times and compare the result before and after the hardening.
Can be integrated and applied across different subject areas and lead beyond domain-specific outcomes	◯	Through learning about facing and fixing a vulnerability, a trainee can learn not only how to fix that vulnerability but also knowledge about computer network, operation system, etc.

Seamlessly integrated with assessment	◯	The training session run on top of Moodle, which Moodle handles assessment activities.
Create polished products valuable in their own right rather than as preparation for something else	◯	A significant outcome of a training about a vulnerability is how to fix that vulnerability in real life, without any conversion or modification to fit with a practical situation.
Allow competing solutions and diversity of outcomes	◯	As trainees are free to do anything to patch the error, many solutions are accepted, as long as they prevent the attack.

Table 6.3 shows that the system satisfies almost all requirements of an authentic learning activity. It can be explained by features of the system. The most important reason is that in this training, the inside machines are "real". It is "real" in sense of the operating system, the software packages, and network data. Building on top of KVM – a para-virtualization hypervisor, system activities are almost same as bare-metal-machines. The interactive interface supports the back-end program by providing a tool for interacting. Last but not least, the input of the system is CVE ids, so they are real incidents which trainees should face them in daily work. Therefore, studying with this system, trainees can hands-on with authentic activities, which reflect real-life situations.

There is only one authentic activity characteristic which the interactive interface cannot satisfy, it is the capability for collaborating between trainees. It comes from a feature of our cyber range environment: each cyber range is isolated from others to prevent leakages of cyberattacks, so until now there is no way to perform any activity between the cyber ranges. Thus, in the system now, there is no special method for collaboration. However, trainees still can use traditional ways such as class forum and in-class discussion.

6.4 User Experience Survey

As introduced in section 2.1.3, this research approaches cybersecurity training from the perspective of pedagogy theories. Among them, the interaction in distance-learning theory defines types of interaction and explains how to implement or improve it. In this research, we focus on the *Learner-Interface* interaction, since it is a real challenge and difference between web-based learning and traditional learning. The interactive interface is developed based on this theory, so after finishing, it needs to be evaluated by learners.

We conducted a user experience survey to check if the improvement on developing an interactive interface and automating cyber attack for training purpose. We asked 15 people: 12 of them are/were students in Information Science field, with a person is a cybersecurity engineer; and 3 of them are working in non-IT related jobs. There are 7

questions, shown in Table 6.4, with Q1 - Q4 about the quality of interaction, while Q5 - Q7 about the result and effectiveness of the training. Participants give scores from 1 to 10, with 1 is worst and 10 is best. There is only question Q3 about the duration of the training, where 1 means too fast and 10 means too long. Participants are classified based on their academic background, experience with Linux and cybersecurity. They use the interface independently, without any guidance. The average scores for every question are shown in Figure 6.1. From this result, some comments can be pointed out:

- All participants agreed that the interactive interface brings an easy way to access cyber ranges. Besides of it, the interface also provides enough and clear information and guidance, so trainees know what they need to do.
- There is a difference between IT and non-IT related people in term of realizing the training motivation. Since IT people are more or less familiar with IT terminologies and command line interface, they understand why the training happens in this way, and they believe that they can acquire knowledge after doing the training. In the other side, non-IT people just follow the instruction without clear understanding, and they are not sure that they can learn anything from it.
- Participants thought that time spent for this training will be a little bit long, since there is no hint to lead to the answer, and they have to study about the vulnerability by themselves.

Following the survey result, the new vulnerability-based defense training is more suitable for hands-on activity for IT-related people to study more about not only a specific bug but also cybersecurity knowledge. The interactive interface provides a comfortable environment for trainees to work with cyber ranges.

Table 6.4: Questions for user experience survey

Topic	**ID**	**Criteria**
Interaction	Q1	Clarity on what to do
	Q2	Clarity on why to do
	Q3	Duration of the training is reasonable
	Q4	It is easy to connect to the cyber range
Result and Effectiveness	Q5	The training help/support you to solve practical problems
	Q6	The training help/support you in learning about cybersecurity
	Q7	The training improves your awareness of cybersecurity

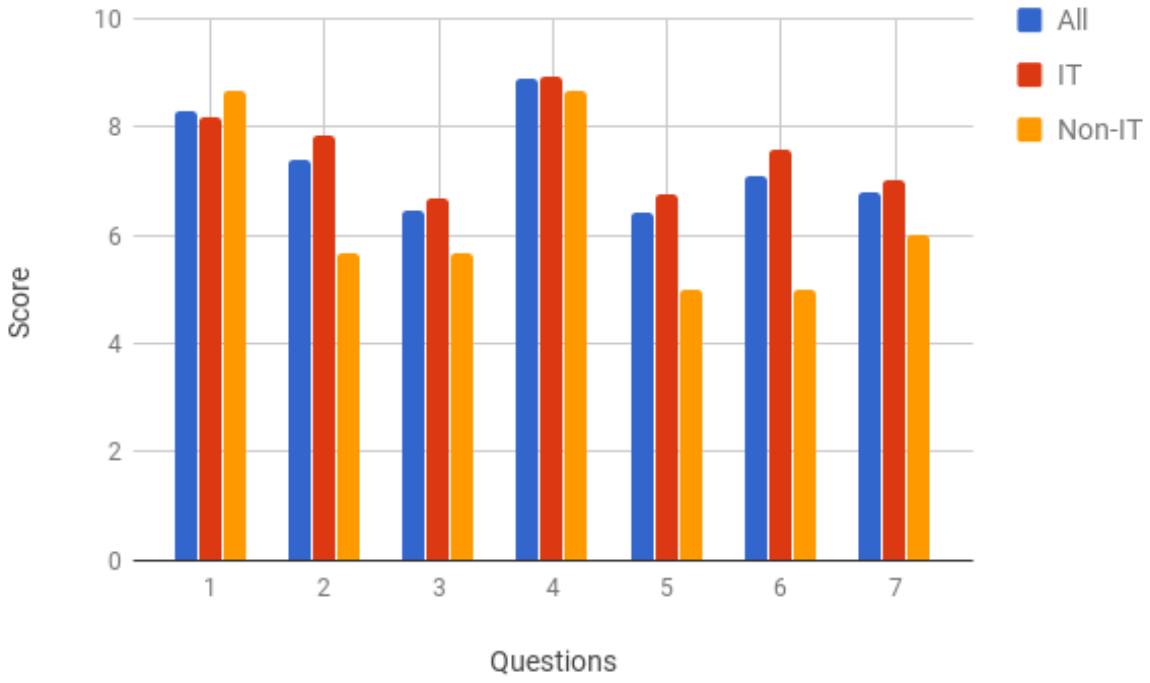


Figure 6.1: Average score for each question in survey

Chapter 7

Conclusion

This research started with the motivation to contribute to the CyTrONE project, with the desire to create a better cybersecurity training methodology. The idea came from looking at the CyTrONE architecture, as we realized that it still lacks automatic cyberattack functionality. Then we considered how trainees manipulate the system, so the interactive interface appeared. After that, we wondered how can we know that our interface is good at interaction with participants. Therefore, pedagogy theories about interaction about distance learning are taken into account. At the initial state of this research, many researches and programs we reviewed. They are about IT training programs, cybersecurity programs, efforts to reproduce incidents in information system. From these articles, we realized that there was not any program to perform real cyberattacks automatically specifying for training. Many researches accepted that the attack activities should be done by IT experts at the time of a training event. Others have workaround solutions, such as investigating the consequences of the attack or simulating them.

The Automatic Attack Controller is created to perform cyberattack without humans that actually do it. The challenge of this task is that each incident or attack technique has its own method and procedure to approach a system. In order to do it by a machine, all attacks need to be generalized. Therefore, we created the training database with information about affected software versions and exploit tools corresponding to vulnerabilities. However, installing vulnerable applications was not an easy task, and that is the reason why instantiation database appeared. The scripts in this database are able to setup a vulnerable machine quickly, but the limitation is that they are prepared manually in advance. If there is a community contributing to this database, it should be a huge advantage and the practical value of this program increase dramatically, since the scripts are "prepare once, run everywhere".

Another plan which we could not finalize in this thesis is using STIX. As introduced in section 2.2.3, STIX is a promising format to reproduce cyber incidents, since a package packs all information related to an incident into it. However, during the time doing the research, we cannot find a real-life package, and STIX is still being developed, so the format and standards are not stable yet. In the future, if STIX is used, the training system will be easy to use and the authenticity of the training session is improved. The incidents are reported by cybersecurity experts in a STIX package with complete and

structured information, and people can replay it in an isolated experimental environment.

About the interactive interface, this is a contribution of another point of view in building a cybersecurity training system. Usually, engineers and researchers know much about cybersecurity and IT related techniques. However, they can fail if a program has many features but it is hard for users to understand and use. Hence, this research introduces the pedagogy theories to support developing a training program. In this research, two pedagogic characteristics we considered: interaction and authentic activity, since easy-to-use and applicability in practical situations are the special features of cybersecurity training programs. We tried to follow the theories in making up the layout, such as how should buttons are placed. Moreover, a terminal connecting directly to the cyber range is added to the web-based interface, because from our experience, we know that accessing the cyber range is always a tedious work and usually cause misunderstanding between learners and organizers. With the terminal, trainees do not need to care about how to connect to the cyber range; and the terminal is on website interface, so it does not require installing an SSH-client software package.

The effectiveness of the pedagogy-theory perspective was proved by the user experience. Even though this survey was done with only 15 people, the result demonstrated that trainees had good reviews about the interactive interface. Therefore, in the future, if developers and organizers pay attention to the pedagogic aspect of both content and appearance of cybersecurity study programs, trainees can learn more from hands-on activities.

Our future plan includes several tasks: 1) Continue to optimize the environment creation time; 2) Integrate a network topology visualization into the interactive web-based interface; 3) Support more vulnerabilities; 4) Support other kinds of cyberattack such as DDoS, malware, etc; and 5) Further improve the interface, so it looks better and familiar with learners.

To conclude this thesis, we want to express our hope in the CyTrONE framework, leading to the *democratization* of cybersecurity training. Currently, in CyTrONE we have a system to create cyber ranges, a system to perform cyberattacks and the framework integrates a part with LMS. Other modules are planned to be integrated into the framework such as a visualization of the cyber range, an IoT module or a wireless module. Then CyTrONE will become a universal framework for every cybersecurity training program, and everyone will be able to use it to study about information security, without the limitation of time or location.

For more information, our programs are open-sourced on GitHub [37], so everybody can try to use our program, and start their training from now.

References

[1] WannaCry Ransomware Statistics: The Numbers Behind the Outbreak. Retrieved on Jul 12th, 2017 from https://blog.barkly.com/wannacry-ransomware-statistics-2017.

[2] World's largest 1 Tbps DDoS Attack launched from 152,000 hacked Smart Devices Retrieved on Jul 25th, 2017 from http://thehackernews.com/2016/09/ddos-attack-iot.html.

[3] Joi L. Moore, Camille Dickson-Deane, Krista Galyen. e-Learning, online learning, and distance learning environments: Are they the same? Internet and Higher Education 14 (2011) 129-135

[4] Moore, M. G. Background and overview of contemporary American distance education. Contemporary issues in American distance education (pp. xiixxvi). New York: Pergamon Press (1990)

[5] Ellen Kalinga. Development of an Interactive e-Learning Management System (e-LMS) for Tanzanian Secondary Schools. Doctoral Dissertation Series No. 2010:10, School of Planning and Media Design, Blekinge Institute of Technology

[6] Younghee Woo, Thomas C. Reeves. Meaningful interaction in web-based learning: A social constructivist interpretation. The Internet and Higher Education, Volume 10, Issue 1, 2007, Pages 15 - 25

[7] Brje Holmberg. The Evolution, Principles and Practices of Distance Education. Studien und Berichte der Arbeitsstelle Fernstudienforschung der Carl von Ossietzky Universitt Oldenburg, band 11.

[8] J. Michael Spector, M. David Merrill, Jan Elen, and M.J. Bishop. Handbook of Research For Educational Communications and Technology, chapter 13. Association for educational communications and technology (AECT).

[9] M.G.Moore. Three Types of Interaction. American Journal of Distance Education, Jan 1989.

[10] D.C.A. Hillman, D.J. Willis, C.N. Gunawardena. Learner-interface interaction in distance education : An extension of contemporary models and strategies for practitioners

[11] Sutton, L. A. The principle of vicarious interaction in computer-mediated communications. International Journal of Educational Telecommunication, 7(3), 223-242.

[12] Vicarious Interaction in an Online Environment Retrieved on Jun 10th, 2017 from http://proctorfree.com/blog/vicarious-learning-online-environment

[13] Jan Herrington, Thomas C.Reeves, Ron Oliver, Younghee Woo, Designing authentic activities in Web-based courses, Journal of Computing in Higher Education, Fall 2004, Vol. 16(1), 3-29.

[14] Reeves, T. C., Herrington, J., & Oliver, R., Authentic activities and online learning, in A. Goody, J. Herrington, & M. Northcote (Eds.), Quality conversations: Research and development in higher education, vol. 25 (pp. 562-567). Jamison, 2004.

[15] Coursera. Retrieved on Jun 8th, 2017 from https://www.coursera.org/

[16] Khan Academy. Retrieved on Jun 8th, 2017 from https://www.khanacademy.org/

[17] Codewars. Retrieved on Jun 8th, 2017 from https://www.codewars.com/

[18] Configuration management Retrieved on Jun 10th, 2017 from https://en.wikipedia.org/wiki/Configuration_management

[19] Michael G. Wabiszewski et al, Enhancing Realistic Hands-on Network Training in a Virtual Environment. Proceeding SpringSim '09 Proceedings of the 2009 Spring Simulation Multiconference, Article No. 69 .

[20] Ariel Futoransky et al, Simulating Cyber-Attacks for Fun and Profit, 2nd International Conference on Simulation Tools and Techniques 2009.

[21] Michael E.Kuhl et al, Cyber attack modeling and simulation for network security analysism. Simulation Conference, 2007 Winter.

[22] Michael Liljenstam et al, RINSE: the Real-time Immersive Network Simulation Environment for Network Security Exercises. Principles of Advanced and Distributed Simulation, 2005. PADS 2005.

[23] Jon Davis and Shane Magrath, A Survey of Cyber Ranges and Testbeds, DSTO, 2013.

[24] R. Beuran, C. Pham, D. Tang, K. Chinen, Y. Tan, Y. Shinoda, CyTrONE: An Integrated Cybersecurity Training Framework, International Conference on Information Systems Security and Privacy (ICISSP 2017), Porto, Portugal, February 19-21, 2017.

[25] Cuong Pham, Dat Tang, Ken-ichi Chinen, Razvan Beuran, CyRIS: A Cyber Range Instantiation System for Facilitating Security Training, International Symposium on Information and Communication Technology (SoICT 2016).

[26] Moodle - Open-source learning platform | Moodle.org. Retrieved on Jun 12th, 2017 from https://moodle.org/

[27] Learning management system Retrieved on Jun 12th, 2017 from https://en.wikipedia.org/wiki/Learning_management_system

[28] Cyber Threat Intelligence Technical Committee Retrieved on Jun 13th, 2017 from https://oasis-open.github.io/cti-documentation/

[29] Mark Wilson and Joan Hash Building an Information Technology Security Awareness and Training Program, National Institute of Standards and Technology, Technology Administration, US Department of Commerce, 2003.

[30] Common Vulnerabilities and Exposures, The Standard for Information Security Vulnerability Names. Retrieved on Jun 17th, 2017 from https://cve.mitre.org/

[31] cnt2lms User Guide, Dat Tang, Cyber Range Organization and Design, Japan Advanced Institute of Science and Technology, April 2017.

[32] cnt2lms v0.2.3, Retrieved on Jun 26th, 2017 from https://github.com/crond-jaist/cnt2lms

[33] CVE Details, The ultimate security vulnerability datasource. Retrieved on Jun 27th, 2017 from https://www.cvedetails.com/

[34] Scrapy - A Fast and Powerful Scraping and Web Crawling Framework. Retrieved on Jun 27th, 2017 from https://scrapy.org/

[35] Metasploit: Penetration Testing Software. Retrieved on Jun 27th, 2017 from https://www.metasploit.com/

[36] makeself - Make self-extractable archives on Unix. Retrieved on Jun 28th, 2017 from https://github.com/megastep/makeself

[37] Cyber Range Organization and Design Chair. Retrieved on August 4th, 2017 from https://github.com/crond-jaist

[38] Shingo Yasuda, Ryosuke Miura, Satoshi Ohta, Yuuki Takano, Toshiyuki Miyachi. Alfons: A Mimetic Network Environment Construction System. 11th EAI International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities, TridentCom 2016.